Noname manuscript No.
(will be inserted by the editor)

A High-Throughput Energy-Efficient Passive Optical Datacenter Network

Yang An · **Changcheng Huang,** ***Senior Member, IEEE***

Received: date / Accepted: date

Abstract Datacenter applications impose heavy demands on bandwidth and also generate a variety of communication patterns (unicast, multicast, incast, and broadcast). Supporting such traffic demands leads to networks built with exorbitant facility costs and formidable power consumption if conventional design is followed. In this paper, we propose a novel high-throughput datacenter network that leverages passive optical technologies to efficiently support communications with mixed traffic patterns. Our network enables a dynamic traffic allocation that caters to diverse communication patterns at low power consumption. Specifically, our proposed network consists of two optical planes, each optimized for specific traffic patterns. We compare the proposed network with its optical and electronic counterparts and highlight its potential benefits in terms of facility costs and power consumption reductions. To avoid frame collisions, a high-efficiency distributed protocol is designed to dynamically distribute traffic between the two optical planes. Moreover, we formulate the scheduling process as a mixed integer programming problem and design three greedy heuristic algorithms. Finally, simulation results show that our proposed scheme outperforms the previous POXN architecture in terms of throughput and mean packet delay.

Keywords delay analysis · datacenter networks · multiple access networks · optical networks · passive optical device

1 Introduction

The emerging trend in cloud computing is to group computing devices into large-scale datacenters that provide existing and growing cloud services and diverse Internet applications to many independent end users. To leverage the rich computing resources that are available, advanced computing technologies such as MapReduce [1] are being widely adopted. Application tasks are partitioned into multiple smaller pieces that are assigned to various servers. This leads to extensive data exchange among servers to complete a single job. The result is massive traffic volumes flowing within a datacenter. Additionally, recent studies [2–6] indicate that various types of traffic, such as unicast, incast, multicast, and all-to-all cast coexist in a datacenter network and exhibit highly dynamic and unpredictable patterns. These patterns change constantly at a granularity of 15 ms [3, 6]. To adjust for this variability, datacenter networks are typically engineered with excessive bandwidth [6–8]. However, supporting the continued exponential growth of bandwidth requires more-expensive electronic switches with higher transmission rates and higher power consumption.

Optical technologies, which feature high bandwidth and low power consumption, offer viable solutions to meeting the high bandwidth and low power consumption requirements. Many optical devices enable transparent data transmission, which makes it easy to upgrade to a higher bit-rate transmission. Moreover, power consumption is dramatically reduced with optical devices compared with their electronic counterparts, where signals are processed at the packet level. Therefore, it is important to explore the feasibility of utilizing optical technologies in constructing datacenter networks [9–13].

Some studies advocate offloading high-volume traffic to optical circuit-switched networks for stand-alone point-to-point bulk transfers. Basic optical modules that are utilized to implement optical interconnections are typically composed of MEMS-based optical space switches. Examples of this architecture include C-through [9], Helios [10], and Proteus [11]. Currently, optical circuit switches are being proposed

Yang An, Changcheng Huang
Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, K1S 5B6 Canada
E-mail: yangan@sce.carleton.ca, huang@sce.carleton.ca

to replace a fraction of the core electronic switches [9], to construct an all-optical architecture [11], and/or to interconnect edge ToR switches as shortcut paths [10]. However, commercially available optical circuit switches are constrained by their high costs and slow configuration speeds. Both limit the performance of optical circuit switches. Therefore, optical circuit switched networks are more suitable for slowly varying traffic with aggregate bandwidth. Moreover, utilizing point-to-point optical lightpaths for multicast/incast traffic transmission requires multiple optical lightpaths to be set up to send redundant traffic, which greatly reduces the effectiveness of the optical networks. Other studies encourage the use of optical packet switches [12, 13]. Because of the difficulties associated with optical buffering, optical packet switched networks require extremely complex systems and electronic control mechanisms for contention resolution, which negates the benefits of optical technologies.

In this paper, we propose a high-throughput passive optical cross-connection network that enables dynamical traffic allocation to satisfy communication demands with mixed traffic patterns. In particular, we use $N \times N$ optical coupler fabrics instead of active optical devices to construct an optical cross-connection network, which dramatically reduces power consumption. Our network consists of two optical planes, where each plane is optimized for specific traffic patterns. Both planes maintain the power and capacity advantages deriving from the deployed optics. Moreover, one of the two planes is designed with a new mechanism that enables high-throughput communications for unicast traffic. We name this novel network Passive Optical Cross-connection Network with Multiple Planes (POXN/MP). We also propose a link layer protocol to coordinate traffic transmissions among connected ports. The proposed protocol enables dynamical traffic distribution between the two planes. We evaluate the protocol's performance using simulations.

The following sections of this paper are organized as follows: Section 2 outlines the physical-layer system and the benefits of POXN/MP; Section 3 describes the proposed link layer protocol; Section 4 presents the designed heuristic algorithms; Section 5 presents numerical results; Section 6 highlights our contributions; and Section 7 concludes our paper.

2 Passive Optical Cross-Connection Network with two Optical Planes

Ni *et al.* proposed the passive optical cross-connection network (POXN) for datacenters [14]. The key optical device used in the POXN is the $N \times N$ coupler fabric. An efficient design to build large-scale coupler fabric was proposed for POXN by interconnecting 3×3 couplers using a Banyan or baseline topology. In this case, all the input power of the 3×3 couplers is delivered to outputs with equal power splitting [14]. Fig. 1 depicts the Banyan topology for building a 27×27 coupler fabric based on 3×3 couplers.

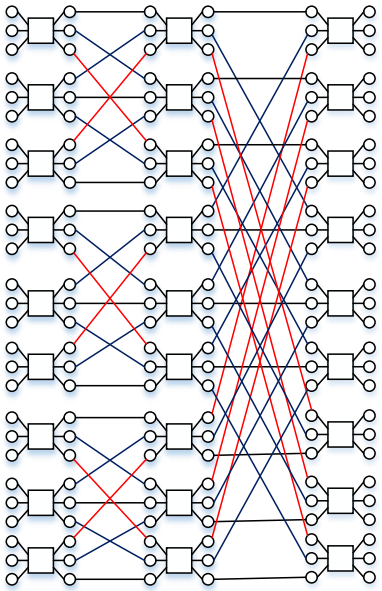

Fig. 1 Interconnection topology within a 27×27 coupler fabric using Banyan topology [14]

In POXN, ports connected to the same optical coupler fabric share a common wavelength and transmit data sequentially within their assigned time intervals. Due to the broadcast property of coupler fabrics, all the connected ports can receive the same data without requiring duplication by intermediate fabrics. Therefore, POXN is efficient in carrying multicast/incast traffic, but it suffers from low throughput for unicast traffic because ports sharing a common wavelength must transmit their traffic in sequence. To overcome the performance limitations of POXN under unicast traffic, we propose POXN/MP, which introduces an extra plane to address unicast traffic. Additionally, POXN/MP efficiently enables high-throughput communication with adaptive response to varying communication patterns in datacenters. This section explores the physical-layer system of POXN/MP and studies its benefits in terms of power consumption and capital expenditure (Capex).

2.1 Physical Interconnections

Similar to POXN, the core of our hardware construct is a large-scale optical coupler fabric that forms a passive optical cross-connection. However, unlike POXN, each input port of the proposed scheme is connected to two transmitters in a transmitting port of an electronic switch or server through a 2×1 wavelength flattened fiber optic coupler (WFFOC),

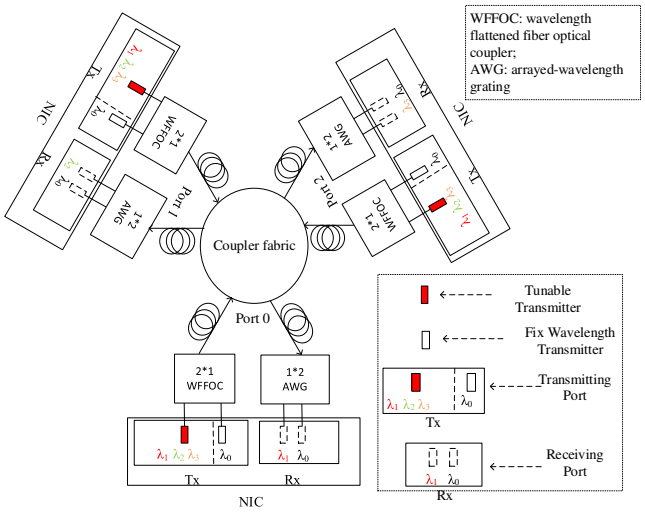

Fig. 2 Physical interconnection of a sample POXN/MP

as shown in Fig. 2. One of the two transmitters uses a fixed wavelength that is common to all transmitting ports, while the other is tunable. In the reverse direction, each output ports of the scheme is connected to the corresponding receiving port of the electronic switch or server through a 1×2 arrayed-wavelength grating (AWG) optical splitter [12]. Each receiving port is equipped with two fixed receivers of different wavelengths. One receiving port uses the same wavelength as the fixed wavelength transmitter, while the other works at a wavelength that varies from receiving port to receiving port.

All the transmitters and receivers with a common wavelength that are associated with different ports form an optical plane. This plane works on a shared transmission channel that is similar to the POXN. The plane, referred to as a multicast plane, can carry multicast traffic efficiently [14]. All the tunable transmitters and fixed receivers with different wavelengths that are associated with different ports form the second optical plane, which we call the unicast plane. The unicast plane is a new mechanism designed for unicast traffic. The different wavelengths of the unicast plane are called different channels. There are $N+1$ channels for an N-port POXN/MP, which includes one channel with a shared wavelength for the multicast plane, and N channels with N different wavelengths for the unicast plane. A tunable transmitter can send unicast traffic to any receiving port by tuning to its channel as long as the receiving port is available. With today's technology, a tunable transmitter can tune from one wavelength to another in 5 ns [15], which is significantly faster than the switching time of optical MEMS switches (typically, approximately 10 ms [14]). However, to reduce facility costs, we use tunable transponders with tuning speeds on the 5 μs time scale [16].

In our POXN/MP, the switching fabric provides a passive broadcast network interconnecting all ports. Both unicast and multicast planes use this switching fabric to carry traffic from any transmitting port to any receiving port. Switching for unicast plane is achieved through wavelength tuning. Specifically, in the unicast plane, each receiving port is set to receiving a specific single wavelength that varies from port to port. A transmitting port can send traffic to any receiving port by tuning its wavelength to the wavelength of the receiving port. The switching speed is therefore depending on the wavelength tuning time of the transmitter, which is much smaller than the switching time of a MEMS based active optical switch. Meanwhile, multiple transmitting ports can communicate with multiple receiving ports in parallel through the unicast plane as long as traffic is delivered through different unicast channels. Moreover, a transmitting port can simultaneously send traffic to a receiving port through both the multicast and unicast planes. Thus, in theory, an N-port POXN/MP achieves $N+1$ times bandwidth compared with POXN.

2.2 Advantages of the Proposed Architecture

This subsection discusses the power budget for POXN/MP and analyzes its benefits in terms of Capex and power consumption. These benefits are demonstrated by comparing POXN/MP with electronic packet-switched networks (EPSNs).

2.2.1 Power Budget

Similar to POXN, there are no extra active optical devices involved in the optical domain of POXN/MP. The calculation of the power budget for POXN [14] shows that the deployed coupler fabric in POXN/MP causes extra power split loss. The power split loss at each output port is $5.47 \cdot [\log_3 N] - 0.2dB$ [14], where N stands for the port number of a coupler fabric. In addition, it is essential to consider the 2×1 WFFOC (2.5 dB), the 1×2 AWG optical splitter insertion (3.5 dB), and the fiber transmission loss (4 dB) [14, 17, 18]. Together, these devices cause a power loss of 10 dB. Given a power budget of 35 dB, offered by a long-range (LR) (1550 nm) transponder with the currently available technology, the port count number of a coupler fabric can scale up to $N = 81$.

2.2.2 POXN/MP vs EPSN

To demonstrate the Capex benefits of POXN/MP, it is essential to compare it with the Capex of EPSN competitors, which can be undertaken by developing a general formula based on a price comparison model. In this model, we assume that the use of the POXN/MP is placed closer to the end servers in the Fat Tree topology. In theory, the unicast

plane of the POXN/MP can sustain the same performance level in terms of network bandwidth capacity when compared to EPSNs.

The typical maximum range for a short-range (SR) transponder is up to 300-400 m, which is insufficient for current warehouse-scale datacenter network environments. We assume the fiber length in our model to be 1 km, which is longer than the maximum range for an SR transponder [14]. Therefore, the EPSNs deploy two LR (1310 nm) transponders per link, one at the server end and one at the switch end. Its price formula can be written as:

$$C^E = S + 2T^{13} \tag{1}$$

where C^E represents the Capex per link of the EPSN, S stands for the cost per port of an electronic switch, which includes the cost of line card and switch fabric, and T^{13} represents the cost of an LR (1310 nm) transponder.

Each port of a server in the POXN/MP equips itself with one fixed wavelength transponder and one tunable transponder. The total number of transponders deployed in a POXN/MP is the same as in an EPSN. The price formula for POXN/MP can be computed as:

$$C^0 = C + T^{15} + T^X \tag{2}$$

where C^O represents the Capex per link of POXN/MP, C stands for the cost per port of the deployed coupler fabric, the 2×1 WFFOC, and the 1×2 AWG optical splitter, T^{15} represents the cost of an LR (1550 nm) transponder,and T^x represents the cost of an LR tunable transponder.

To illustrate the cost comparison in a real network, we use the datacenter network suggested by Cisco as an example. This datacenter network is typically constructed in the Fat Tree structure, where the popular deployed ToR switches are CiscoNexus 3548switches [18]. These switches have 48 $\times$ 10 GBits SFP+ports. We use the 48-port coupler fabric to replace the Cisco Nexus 3548 switch as an example.

The actual prices for the aforementioned elements vary among customers, depending on the quantity of equipment sold. In this model, the price of an LR (1310 nm) transponder is based on the vendor's online sale price. However, the prices of an electronic packet switch and a WDM tunable transponder are based on published Capex studies. We list the prices that are relevant to our study in Table 1. The prices are given in relative values. Actual prices may vary from vendor to vendor.

To explore the energy-related economic benefits, we take the power consumption of the deployed devices into consideration. The coupler fabric we deployed is a passive optical device that consumes zero power, while the Cisco Nexus 3548 switch consumes 265 W [18]. Moreover, Cisco's public product data sheets [21, 22] reveal that the tunable LR and fixed wavelength LR (1550 nm) transponders deployed

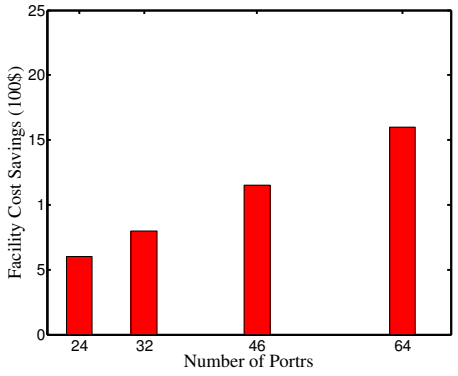

Fig. 3 Facility cost savings of replacing the EPSN with the POXN/MP

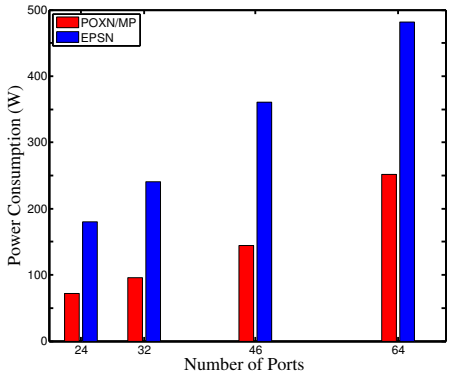

Fig. 4 Power consumption (W) of the POXN/MP and the EPSN

in POXN/MP both consume 1.5 W, while the LR (1310 nm) transponder deployed in EPSNs only consumes 1 W.

Based on the prices given in Table 1, we can compute the Capex per port for the EPSN and the POXN/MP through equations (1) and (2), respectively. The facility cost savings for replacing a N-port electronic switch in EPSN with a N-port coupler fabric in the POXN/MP are depicted at Fig. 3. We can see the savings of utilizing POXN/MP increae as the scale of a datacenter grows. We choose port numbers 24, 32, 46, and 64 as examples, since these port numbers are common to commercial electronic switches. Fig. 4 depicts the power consumption of the EPSN and the POXN/MP on a per port basis.

POXN/MP does not use electrical equipment, which leads to high-energy efficiency and easier migration to 40-GigE and beyond. However, to avoid collisions among different transmitting ports at receiving ports and to optimize the unicast plane's bandwidth utilization, a high-throughput Multiple Channels Distributed Access Protocol (MCDAP) is proposed next to schedule traffic transmission among connected transmitting ports. The MCDAP also supports dynamic traffic allocation between the two optical planes. Our system results in efficient communications for both unicast and multicast/incast traffic with a fully distributed operation.

Table 1 Deployed Devices' Price Table

Device	Usage	Price
Cisco Nexus 3548	Used for edge and aggregation tier switches	$21600 [11]
Passive Optical Coupler Fabric	Used for replacing edge electronic switches (48 ports)	$480
2x1WFFOC and 1x2 AWG optical splitter	Used for combining two links into one and splitting one link into two	$40
10G LR(1310 nm) Transponder	Used for LR (1-40 km), 10G, transponder (1 wavelength)	$200 [20]
10G LR WDM Tunable Transponder	Used for LR, 10G, transponder (48 wavelengths)	$525 [2]
10G LR(1550 nm) Transponder	Used for LR (40-80 km), 10G, transponder (1 wavelength)	$350 [20]

3 Multiple Channels Distributed Access Protocol

We now describe how the MCDAP works. There are three major types of messages in the MCDAP: the control messages that are carried by the multicast plane; the unicast data messages that are mainly carried by the unicast plane but can also be carried by the multicast plane in sequence when this plane has extra capacity; and the multicast data messages that are only carried by the multicast plane.

The working process of the multicast plane of the MCDAP is similar to that of the HEDAP [14]. We begin with a brief introduction about how the HEDAP works, and then we present the working process of the MCDAP in detail.

The HEDAP protocol can be divided into two phases: the discovery phase and the data transmission phase [14]. The discovery phase is designed to achieve the plug-and-play objective and to discover other ports in POXN. The data transmission phase follows the discovery phase, and it can be further divided into multiple scheduling cycles. Within each cycle, each transmtting port has a chance to send a burst of frames during its assigned time interval. At the end of the burst, each transmitting port also broadcasts the amount of traffic it needs to send for the next data transfer cycle through a REQUEST message. Each port maintains a list of all the discovered ports and their corresponding traffic requests for the next cycle. With this list, all the transmitting ports use an identical algorithm to decide a common scheduling. To accommodate port churns, the discovery phase is allowed to repeat after running the data transmission phase long enough. During the repeated discovery phase, new ports can be discovered, the clock can be re-referenced and resynchronized, and the round-trip time and loopback time can be re-measured.

Inspired by HEDAP, we build the multicast plane of the MCDAP, which alters the original control information that the HEDAP broadcasts through ANNOUNCEMENT, CONFIRMATION, and REQUEST messages. We require that each transmitting port indicates the addresses for unicast traffic and the exact amount of unicast and multicast traffic to each receiving port through the multicast plane by the corresponding control messages. After the discovery phase of the multicast plane ends, each transmitting port also maintains a traffic request list of the first scheduling cycle for all the unicast channels. This list contains the amount of unicast traffic to be sent among all transmitting and receiving ports in addition to the corresponding receiving port addresses. Based on this list, each transmitting port locally runs a common scheduling algorithm, which generates a globally identical scheduling result to guarantee that each transmitting port sends its traffic without any collisions. Notably, while unicast traffic can be carried by the unicast channels in parallel, it can also be carried by the multicast plane in sequence. Therefore, the MCDAP can achieve load balance between the multicast and unicast planes by dynamically allocating unicast traffic to the two planes.

3.1 Discovery Phase for the Multicast Plane

There are two kinds of discovery phases for the multicast plane: the discovery phase at system boot and the discovery phase between data transfer phases.

3.1.1 Discovery Phase at System Boot

Upon system activation, each port starts with sending an ANNOUNCEMENT message through the multicast plane after a random back-off time, which lets them be detected by the other ports. The ANNOUNCEMENT message contains the mac address of the transmitting port and the time stamp of when the message was transmitted along with the discovery window time period. This message also contains information regarding the amount of multicast traffic to be transmitted in the first data transfer cycle and a list containing the amount of unicast traffic as well as the corresponding receiving port mac addresses for the same cycle.

As a result of the broadcast property of coupler fabrics, every port hears the same information from the multicast plane. Thus, the first ANNOUNCEMENT message received at its own local receiver is also the first message successfully received at all the other ports. We call a port to be a successful port if it successfully receives its own ANNOUNCEMENT message. The first successful port decides the discovery window period. The discovery window starts with the reception of the first ANNOUNCEMENT message and lasts as specified in that message [14]. Once the announced period expires, no other ports are allowed to send message.

The first discovered port will broadcast a CONFIRMATION message to summarize all the discovered ports.

After successfully receiving the CONFIRMATION message, each transmitting port generates two identical scheduling lists for the following data transfer cycle: one contains the amount of multicast traffic as well as its transmitting mac address for the multicast plane, and the other contains the amount of unicast traffic as well as the corresponding receiving and transmitting port mac addresses for the unicast plane. With these lists, each transmitting port locally runs a global identical multicast and unicast scheduling algorithm for each respective plane, which allocates time intervals for each transmitting port such that traffic collisions for both planes can be avoided.

3.1.2 Discovery Phase between Data Transfer Phases

After running the data transfer phase for the multicast plane long enough, a discovery phase follows to add new ports. During this phase, only the newly joined ports broadcast an ANNOUNCEMENT message to allow it to be discovered by the existing ports. Once the discovery window ends, the current clock-reference port sends a CONFIRMATION message that contains information on all the existing and newly joined ports as well as information (e.g., the start time of the next discovery phase) regarding the following data transfer and discovery phases. Hence, a newly joined port must wait to receive the CONFIRMATION message, which means it must wait at least one data transfer phase before it can join the running network.

3.2 Data Transfer Phase for the Multicast Plane

An identical scheduling algorithm locally decides when a port can transmit its multicast traffic and how much it can transmit. All the discovered ports share the same wavelength for the multicast plane. Thus, each port can only send data in sequence during its assigned time interval. At the end of its transmission, every port broadcasts its traffic request for the next cycle for both the multicast and unicast planes via a REQUEST message. The information carried by a REQUEST message pertinent to the unicast plane is a list that contains the exact amount of unicast traffic for the next data transfer cycle and the corresponding transmitting and receiving port mac addresses. Moreover, to achieve load balance between the two planes, when the multicast plane has extra bandwidth capacity, transmitting ports in the MCDAP send parts of the unicast traffic through this plane in sequence.

When a port dies, all the other ports do not need to do anything other than schedule their future cycles assuming that the dead port do not have anything to send. If the dead port is the clock-reference port, all the other ports still behave the same way until the next discovery phase during which all the other ports will remove the failed port from their lists [14]. The second port on their lists then becomes the clock-reference port. If it is desired that all other ports can detect the dead port as soon as possible, we can configure all ports to have their every discovery phases to rediscover all ports instead of discovering new ports only. Then a dead port can be detected and removed in the next discovery phase after its failure.

3.3 Idle Phase for the Unicast Plane

The unicast plane begins by entering the idle phase at the same time as the multicast plane begins its discovery phase. During the idle phase, the ports must wait for the successful reception of the ANNOUNCEMENT and CONFIRMATION messages from the multicast plane. The idle phase for the unicast plane ends at the same time as the discovery phase for the multicast plane, and the multicast plane eventually enters another discovery phase after running the data transfer phase for the multicast plane long enough. Meanwhile, the unicast plane enters another idle phase.

3.4 Data Transfer Phase for the Unicast Plane

Once the CONFIRMATION message has been successfully received by all the discovered ports through the multicast plane, each port generates two separate traffic request lists: one for the multicast plane and the other for the unicast plane. Running global identical algorithms locally, a port can compute their start times, scheduling orders, and assigned time intervals for the multicast and unicast planes respectively. Different from the multicast plane, the unicast plane needs to compute which receiving port it can transmit during an assigned interval.

Every port estimates its data transfer cycle period for the multicast and unicast planes, respectively. If there is extra bandwidth capacity for the multicast plane, the corresponding ports will use the multicast plane to transmit parts of their unicast traffic in sequence to achieve load balance.

As opposed to the multicast plane, the unicast plane consists of multiple channels dedicated to different receiving ports. However, if multiple transmitting ports communicate with the same receiving port at the same time, collisions can occur. Thus, an algorithm must be designed for distributing traffic among different channels of the unicast plane as well as to optimize the bandwidth utilization.

4 Algorithm for the Unicast Plane of the MCDAP

We now describe how the scheduling algorithms for the unicast plane of the MCDAP work. In this paper, we assume

that each transmitting port has multiple queues, with a dedicated queue for each receiving port. Every port stores its unicast traffic to a specific receiving port in its corresponding unicast queue. In addition, different unicast queues have different amounts of unicast traffic that must be sent to different receiving ports during a data transfer cycle.

In POXN/MP, a transmitting port sends traffic from different unicast queues through different wavelengths during different allocated time intervals. The wavelength used by one transmitting port for one time interval is recycled and assigned to another transmitting port during the next time interval. The receiving port then receives traffic through a single dedicated wavelength. This wavelength-port association is referred to as a run time decision.

As discussed above, todays tunable transponders can typically support tuning speeds on the 5-μs time scale, which is nearly four times greater than the transmission time of the maximum Ethernet frame size (1.2 μs) for a 10 Gbps link. If traffic transmission is scheduled on a packet-by-packet basis, frequent tuning may lead to bandwidth waste of approximately 80%. Moreover, scheduling traffic on a packet level will require that packet level information be available to all transmitting ports within each cycle, which will introduce more overhead. Additionally, it costs more energy for a tunable transmitter to tune more frequently. Therefore, our top priority is to minimize tuning time.

One way to minimize tuning time is to send traffic to the same receiving port as a burst. In the following discussion, we always assume that a transmitting port will send traffic to the same receiving port continuously as a concatenated burst. It must finish sending one unicast queue completely before it can begin sending traffic from another unicast queue. When a sending port is sending to a receiving port, it will not be interrupted by other ports until it finishes. Using this approach, we minimize the impact of the tuning time and simplify our scheduling algorithms.

Due to the randomness of traffic, there are several situations that may cause bandwidth to be wasted. First, a transmitting port has finished sending all its unicast queues but must wait for other ports to finish all their unicast queues before a new cycle can start. This is called Type 1 mismatch.Second, called Type 2 mismatch, a transmitting port has unicast queues to be sent to receiving ports, but none of the receiving ports are available. Thus, the transmitting port has to wait until one of the receiving ports is available. Third, a transmitting port has traffic to send and the corresponding receiving port is available, but the transmitting port has to tune its wavelength to the receiving wavelength.

An illustrative transmission scheduling of a 3-port POXN/MP network is depicted in Fig. 5 as an example. The rectangles marked with T_{ij}^{Q} describe different unicast queues belonging to different transmitting ports. Let T_{ij}^{Q} be the transmission time of packets in the unicast queue that will be sent

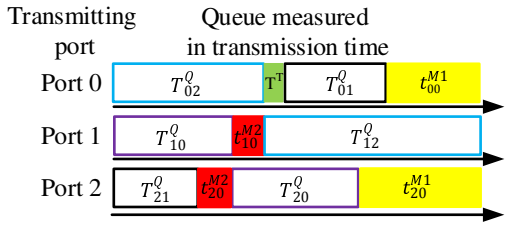

Fig. 5 A sample transmission scheduling of a 3-port POXN/MP

from transmitting port i to receiving port $j(0 \leq i, j \leq 2)$. For example, T_{02}^{Q} denotes the transmission time of the packets, which belong to transmitting port 0 and will be sent to receiving port 2. Let T_{ik}^{Mx} indicate the k-th $(0 \leq k)$ Type $x(x = 1 or 2)$ mismatch that transmitting port experiences. $\sum T_{ik}^{Mx}$ is the total Type x mismatches that transmitting port i suffers for a data transfer cycle. The yellow rectangles marked with T_{ik}^{M1} denote the Type 1 mismatch. For instance, close to the end of the data transfer cycle, transmitting port 0 and port 2 have finished their transmission, but port 1 still has data to be sent. In this case, transmitting port 0 and port 2 experience their first $(k = 0)$ Type 1 mismatch, T_{00}^{M1} and T_{20}^{M1}, respectively. The red rectangles marked with T_{ik}^{M2} represent the Type 2 mismatch. For example, transmitting port 2 suffers its first Type 2 mismatch T_{20}^{M2} when it finishes its data burst to receiving port 1. In this case, transmitting port 2 becomes available and has data to be sent to receiving port 0; however, receiving port 0 is not available, so transmitting port 2 has to wait until port 1 finishes. The third scenario is represented by the green rectangle, labeled T^{T}, which denotes a tunable transponders constant tuning time.

The Type 1 and Type 2 mismatches have to be addressed by designing scheduling algorithms that minimize the idle time caused by the mismatch of transmitting and receiving ports. The tuning time can be minimized by proactively scheduling and tuning the wavelength for a burst transmission whenever possible. Compared with the transmission time of a burst, the tuning time is typically small.

To minimize the aforementioned mismatches, we first formulate them as a mathematical programming problem and then propose several heuristic solutions in the following subsections. These solutions try to optimize traffic distribution among different wavelengths for the unicast plane and enable dynamic unicast traffic allocation.

4.1 Problem Descriptions and Formulation

The problem to be addressed can be summarized as follows: Maximize overall throughput under the constraints that traffic from one transmitting port to one receiving port must be sent in a burst, a receiving port can only receive traffic from

one transmitting port and a transmitting port can only send traffic to one receiving port at any time.

We will focus on one data transfer cycle of a phase because the optimization will be performed on a cycle-by-cycle basis. For convenience of analysis, we assume that time starts from the beginning of a data transfer cycle. Given an N port POXN/MP network, assume that at the beginning of the data transfer cycle, the unicast queue sizes in each transmitting port are known by the MCDAP described earlier. We denote these initial unicast queue sizes as T_{ij}^{Q}, where a unicast queue size is measured as the transmission time of the packets in a unicast queue, i is the index of a transmitting port and j is the index of a receiving port, and $(0 \leq i, j \leq N-1)$. Let $\delta_{ijt} = 1$ indicate that transmitting port i is sending to receiving port j at time t; otherwise,$\delta_{ijt} = 0$. Let t_{ij}^{S} denote the starting time of port i sending to port j and t_{ij}^{F} denote the corresponding finishing time. We assume the tuning time is negligible after proactive tuning. If the tuning time is not negligible, it can be counted as part of a data burst in the worst case. Our goal is to minimize the period of a data transfer cycle as below:

Objective:

$$min(\max_{i,j} t_{ij}^{F})$$

s.t.

$$\delta_{ijt} = 0, \forall t \leq t_{ij}^{S}, i, j \leq N-1, i \neq j \tag{3}$$

$$\delta_{ijt} = 0, \forall t \geq t_{ij}^{F}, i, j \leq N-1, i \neq j \tag{4}$$

$$t_{ij}^{F} - t_{ij}^{S} = T_{ij}^{Q}, \forall i, j \leq N-1, i \neq j \tag{5}$$

$$\delta_{ijt} = 1, \forall t : t_{ij}^{S} \leq t \leq t_{ij}^{F}, \forall i, j \leq N-1, i \neq j \tag{6}$$

$$\sum_{j} \delta_{ijt} \leq 1, \forall t \geq 0, i \leq N-1, i \neq j \tag{7}$$

$$\sum_{i} \delta_{ijt} \leq 1, \forall t \geq 0, j \leq N-1, i \neq j \tag{8}$$

where equations (3) and (4) define the starting and ending times of bursts, equations (5) and (6) define burst lengths, equation (7) is the constraint imposed by a receiving port and (8) is the constraint by a transmitting port.

The above formulation is a complex programming problem [11]. It typically takes some time for each transmitting port to process. A transmitting port can only start conducting the calculation after it receives all the requests for the next data transfer cycle. If it finishes processing before the end of the current data transfer cycle, it will not introduce any overhead. Otherwise, it will introduce extra overhead, leading to lower efficiency. Therefore, the time for this processing should be much smaller than a data transfer cycle, which is typically less than 1 ms for highly dynamic traffic patterns. In our numerical examples given later, the period of a data cycle can be as small as 150 μs. To complete the optimization process in such a brief time is almost impossible unless a significant amount of computing power is deployed. Thus, it is necessary to look at heuristic solutions.

4.2 Shortest Queue First Algorithm

One of the main empirical studies of datacenter traffic characteristics shows that flow sizes of 80% of the datacenter traffic are considerably small (i.e., less than 10 KB). However, less than 20% of the total bytes are contributed by a few large flows [4]. Based on these observations, we assume that at the beginning of a data transfer cycle, the shorter unicast queues belonging to different transmitting ports may resemble each other in size. If each transmitting port tries to send its shortest unicast queue with an available receiving port first, a considerable number of transmitting ports will complete their data bursts at roughly the same time. This will allow these transmitting ports to recycle the wavelengths to other receiving ports at roughly the same time and, therefore, minimize Type 2 mismatch. To be consistent, when two transmitting ports try to send to the same receiving port, the transmitting port with the smaller unicast queue will send first. We name this algorithm Shortest Queue First (SQF).

As an example, we still use the aforementioned 3-port POXN/MP network to demonstrate the transmission process of the SQF algorithm. The high-level diagram of the transmission process of the SQF algorithm is depicted in Fig. 6. To be consistent, we use the same notations as in Fig. 5. Moreover, let t_{ijt}^{Q} denote the transmission time of the remaining packets in the unicast queue that will be sent from transmitting port i to receiving port $j(0 \leq i, j \leq 2)$ at time t. For instance, $t_{01t_1}^{Q}$ denotes the transmission time of the remaining packets in the unicast queue that will be sent from transmitting port 0 to receiving port 1 at time t_1.

Fig. 6 depicts snapshots of the remaining unicast queues of all transmitting ports at different times. The time coordinates denote the times when ports begin transmitting to the desired receiving ports (which correspond to events occurring in the event list). The unicast queue measured in transmission time coordinates describe the status of the remaining unicast queues for all transmitting ports corresponding to different event times. We assume that each transmitting port orders its unicast queues according to their sizes. The port coordinates denote the transmitting ports.

The SQF algorithm consists of the following steps:

Step 1: At the beginning of each cycle, each transmitting port will try to select the shortest unicast queue to send first. If there is a collision with other transmitting ports, the transmitting port with the shortest unicast queue size will win, and all other transmitting ports that have collisions will try their second-smallest unicast queues with different receiving ports. This process continues until a transmitting port finds the non-colliding shortest unicast queue; otherwise, the transmitting port will be idle and wait for a receiving port to become available. A transmitting port can figure out collisions beforehand because every transmitting port has the

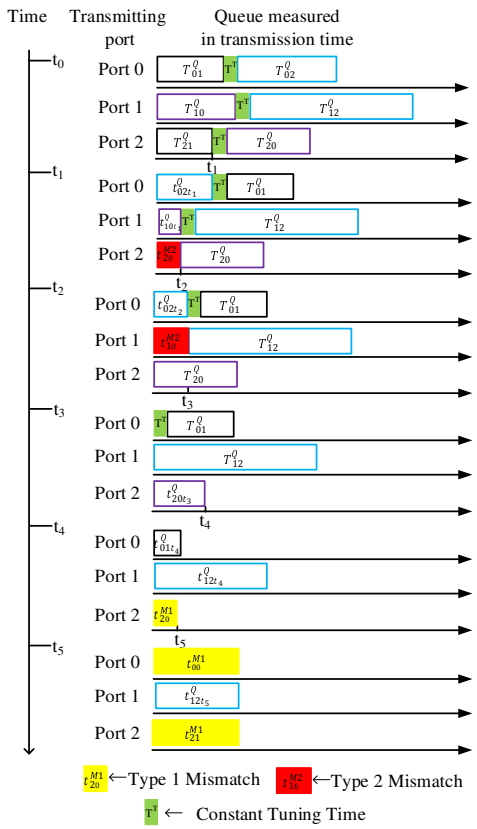

Fig. 6 Transmission process of a 3-port POXN/MP using the SQF algorithm

traffic burst information, learned through the MCDAP, about all other transmitting ports.

In this example, at the beginning of the data transfer cycle (t_0), transmitting port 2 selects its unicast queue 1 to send first. Simultaneously, transmitting port 0 selects its unicast queue 2, although unicast queue 1 is smaller. Meanwhile, transmitting port 1 selects its unicast queue 0 as its first data burst. It is assumed that transmitting port 2 will complete its data transmission of packets in unicast queue 1 at t_1.

Step 2: Once a transmitting port has completed the data transmission for its first selected unicast queue, it begins to select its next shortest unicast queue to an available receiving port as its second data burst. If there is no available receiving port at this moment, the transmitting port will be idle and wait for an available receiving port.

As depicted in Fig. 6, transmitting port 2 will first finish sending traffic from its selected unicast queue 1 to receiving port 1 for this data transfer cycle at t_1. Then, transmitting port 2 only has unicast queue 0 that needs to be sent to receiving port 0. However, at this moment, receiving port 0 is not available. Therefore, transmitting port 2 will be idle and suffer from Type 2 mismatch from until transmitting port 1 completes its data burst to receiving port 0 at t_2, where $t_2 = t_1 + t^Q_{10_{t_1}}$. During this idle time, transmitting port 2 can start tuning to the wavelength of receiving port 0 so that the tuning time does not incur extra overhead. This is what we call proactive tuning.

Step 3: Following the same rule as mentioned in step 2, all the transmitting ports complete their corresponding data bursts for the current data transfer cycle.

This step is illustrated from t_2 to t_5. At t_2, transmitting port 1 completes its remaining data burst to receiving port 0. As a result, receiving port 0 is available and transmitting port 2 can begin its next data burst to receiving port 0 immediately. Meanwhile, transmitting port 1 only has unicast queue 2 that needs to be sent to receiving port 2. However, at this moment, receiving port 2 is busy, so transmitting port 1 experiences Type 2 mismatch from t_2 until transmitting port 0 completes its data burst to receiving port 2 at t_3. Transmitting port 1 can certainly take this opportunity to tune its wavelength to receiving port 2 as an act of proactive tuning.

At t_3, transmitting port 0 finishes its data burst to receiving port 2, so receiving port 2 is available and transmitting port 1 can send its unicast queue 2 to receiving port 2. Meanwhile, receiving port 1 is also available; however, transmitting port 0 must wait for a constant tuning time T^T to tune its tunable transmitter to receiving port 1.

At t_4, transmitting port 2 completes all its data bursts for this data transfer cycle. Then, transmitting port 2 becomes idle. Thus, transmitting port 2 suffers from the Type 1 mismatch from t_4 until t_5. At t_5, transmitting port 0 completes its data burst and experiences Type 1 mismatch until at the end of the current data transfer cycle.

4.3 Longest Queue First Algorithm

When we employed the SQF scheduling algorithm, we observed that the unicast queues belonging to the last completed transmitting ports tend to have a larger average unicast queue size at the end of a data transfer cycle. Moreover, during this period, some transmitting ports have already completed their data transmissions and are now waiting for the remaining un-finished transmitting ports. This will increase Type 1 mismatch. To minimize this mismatch, we propose another approach in which each transmitting port will try to send the longest unicast queue with an available receiving port first during the data transfer cycle. This algorithm is named Longest Queue First (LQF).

The only difference with the SQF algorithm is that each transmitting port first tries to send traffic from the longest unicast queue to an available receiving port. Intuitively, compared with the SQF algorithm, even though the LQF algorithm can reduce the Type 1 mismatch at the end of the data transfer cycle but tends to suffer from more Type 2 mismatches at the beginning of a cycle. Thus, there may not be a significant performance increase for the LQF algorithm.

These two algorithms show that transmitting ports in the POXN/MP can, without collisions, send unicast traffic in parallel to different receiving ports through the unicast plane. The tuning time can be minimized through proactive scheduling. The SQF algorithm can reduce the impact of Type 2 mismatch. However, it experiences more Type 1 mismatches at the end of a data transfer cycle. In contrast, the LQF algorithm is helpful in minimizing the impact of Type 1 mismatch to some extent, but it suffers from more Type 2 mismatch. To minimize both types of mismatch and to achieve load balance between the two optical planes, we propose a third algorithm.

4.4 SQF with Cut-over Function

Another method to address the Type 1 mismatch toward the end of each cycle is to utilize the two optical planes dynamically. Before the end of each cycle, the last transmitting ports will use both the multicast plane and the unicast plane to send their unicast traffic from un-transmitted unicast queues whenever the multicast plane has extra bandwidth capacity. Therefore, the wasted bandwidth caused by the Type 1 mismatch can be minimized. Moreover, in the MCDAP, the two planes must start their next data transfer cycles at the same time. If the multicast plane finishes its data transfer cycle earlier, it will be idle and wait for the unicast plane to complete, wasting bandwidth for the multicast plane. Thus, this method can also optimize bandwidth utilization for the multicast plane by sending parts of unicast traffic to align the two planes when the multicast plane has extra capacity. We achieve this by designing a function named cut-over, which can dynamically distribute traffic between the two planes. The cut-over function is helpful to optimize overall system bandwidth utilization.

To elaborate on the cut-over function, it is necessary to consider the multicast plane. The cut-over function can be achieved according to the following operations:

First, every transmitting port estimates the period of its current data transfer cycle for both the multicast and unicast planes, which can be done by running the HEDAPs scheduling algorithm for the multicast plane based on the multicast traffic burst information and the SQF algorithm for the unicast plane based on the unicast traffic burst information.

Second, each transmitting port compares these two data transfer cycle periods. When the data transfer cycle for the unicast plane is longer, the corresponding transmitting ports move a certain amount of unicast traffic from its unicast queues to the multicast queue. This may result in the order of data transmission for a unicast queue being different. Therefore, out-of-order delivery may occur for some packets. The out-of-order delivery problem can be addressed by running a hash function in every transmitting port, similar to ECMP [24]. The amount of unicast traffic should try to make the two planes finish their cycles at nearly the same time.We name this amount of unicast traffic cut-over traffic.

Third, each transmitting port updates the scheduling order for the two planes. The cut-over unicast traffic will be sent by the multicast plane in sequence immediately after the completion of the original multicast traffic transmission.

After running the cut-over function, each transmitting port follows the new scheduling order so that the two planes will finish at nearly the same time. We name this new algorithm Shortest Queue First with Cut-over Function(SQF/CF).

5 Numerical Results

We evaluate the MCDAP in two parts through simulations. In the first part, we focus only on the efficiency of the proposed algorithms for the unicast plane (the numerical results are presented in subsection 5.1). In this paper, efficiency refers to the ratio of the mean transmission time of unicast packets for a transmitting port divided by the period of a data transfer cycle for the unicast plane. In the second part, we implement the MCDAP protocol using the best suitable algorithm, and we test the performance of our protocol at a system level.

5.1 Algorithm Efficiency Analysis

We evaluate our proposed algorithms over a 48-port POXN/MP system through simulations. The port number 48 was chosen as an example because this port number is common to the widely used electronic switches in datacenters. In the setup environment, all 48 transmitting ports work at a 10 Gb/s wavelength line rate. For simplicity, we assume that the number of packets in a dedicated unicast queue follows a uniform distribution. Moreover, simulations in subsection 5.1 were achieved based on burst-by-burst basis. In this paper, the number of packets in a queue also denotes the corresponding queue length. Every packet has an equal size of 1024 bytes. This size of 1024 has been used to ensure consistency with the setup parameters in the HEDAP [14]. In addition, the tunable transmitter has a constant tuning time that is equal to the transmission time of 6 packets. We use 6 because the corresponding transmission time is nearly equivalent to 5 μs. All simulation results shown are with 95% confidence intervals.

An idealized result is calculated as the upper bound for the algorithm efficiency under the following assumptions: any transmitting port has the same amount of unicast traffic to be sent to all other receiving ports during a data transfer cycle; except for the constant tuning time, there are no Type 1 or Type 2 mismatches between any two adjacent scheduled data bursts; and all the transmitting ports start and finish their data bursts at the same time. Thus, for an N-port

POXN/MP, the upper bound efficiency can be computed as:

$$E = \frac{(N-1) \times T_{ij}^{Q}}{(N-1) \times T_{ij}^{Q} + (N-2) \times T^{T}} \quad (9)$$

where T^T denotes a tunable transponders constant tuning time, T_{ij}^{Q} denotes the transmission time of packets in a dedicated unicast queue that will be sent from transmitting port i to receiving port j. In this case, all the unicast queues have the same queue length. When we compute the upper bound of each case, the value of T_{ij}^{Q} is set to the maximum queue length of a corresponding uniform distribution for simulations in subsection 5.1. Although this upper bound is looser than the result calculated using the formulations in subsection 4.1, it is more feasible in real system. As we will show later, the results of our best scheduling algorithm can be quite close to the upper bound.

Discussions in section 4 show that three factors mainly affect the efficiency of our proposed algorithms. The first is the relative overhead introduced by the tuning time, which is mainly affected by the mean of the unicast queue length. The second is the Type 2 mismatch, which is mainly influenced by the variation of the unicast queue length. The third factor is the Type 1 mismatch, which is mainly affected by the transmission time of the multicast traffic. Following this, we study how the mean of the unicast queue length, the variation of the unicast queue length, and the transmission time of the multicast traffic affect the algorithms efficiency. This can be tested through three different scenarios.

5.1.1 Mean of the Unicast Queue Length

In the first scenario, the unicast queue length follows a uniform distribution. Moreover, the mean of the unicast queue length is different, increasing from 45 to 65 in increments of 5. The standard deviation of each unicast queue length is set to 14.43. In addition, the multicast queue length for each transmitting port is set to 5. We will study the algorithm efficiency of different multicast queue lengths in a later subsection. We test the efficiency of the aforementioned algorithms and depict the results in Fig. 7. Their results are compared with the upper bound.

In Fig. 7, the horizontal coordinates denote the mean of the unicast queue length. The vertical coordinates denote algorithm efficiency. The integrated zoom-in figure in Fig. 7 is used to show the confidence interval. To keep the other figures orderly, we only present the zoom-in figure in Fig. 7. Fig. 7 shows how the algorithm efficiency changes with the mean of the unicast queue length. We see that when the mean of the unicast queue length becomes larger, the efficiency for all the proposed algorithms increases. This is due to the impact of the tuning time becoming smaller. Consequently, the algorithm efficiency will be higher. Fig. 7 also

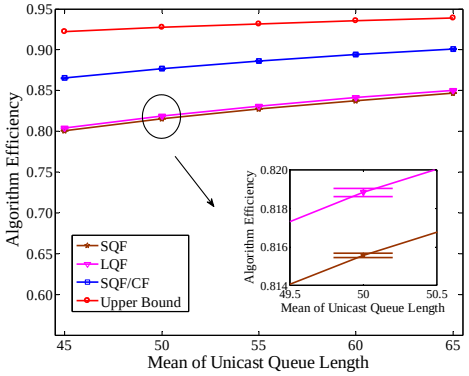

Fig. 7 Algorithm efficiency for SQF, LQF, and SQF/CF with different means of unicast queue length but a constant standard deviation of unicast queue length. Simulation results shown are with 95% confidence intervals..

shows that there is no significant difference between the efficiency of the LQF and SQF algorithms. This is most likely due to the tradeoff between Type 1 and Type 2 mismatches.

However, there is approximately an 8% relative increase in algorithm efficiency after adding the cut-over function compared with the LQF and SQF algorithms. This is a result of shortening the Type 1 mismatch at the end of a data transfer cycle. Moreover, Fig. 7 shows that the efficiency of the SQF/CF algorithm under this simulation environment approaches the upper bound, with an approximately 7% relative difference when the mean of the unicast queue length is set to 45. Thus, there will be little practical gain on computing the bound using the optimization formulation in subsection 5.1.

5.1.2 Variation of the Unicast Queue Length

In the second scenario, the unicast queue length still follows a uniform distribution. Moreover, the mean of the unicast queue length is set to 65, which is the rightmost point in Fig. 7. The standard deviation of each unicast queue length increases from 8.66 to 20.21 in increments of 2.88. In addition, the multicast queue length for each transmitting port is set to 5. We test the efficiency of the aforementioned algorithms and depict the results in Fig. 8 Their results are compared with the upper bound again. In Fig. 8, the horizontal coordinates represent the standard deviation of the unicast queue lengths. The vertical coordinates represent the algorithm efficiency.

Similar results have been observed in the second scenario. There is nearly no efficiency difference between the SQF and LQF algorithms. Moreover, the cut-over function can increase algorithm efficiency by nearly 9% when the standard deviation of the unicast queue length is 20.

Fig. 8 shows how the algorithm efficiency changes with the variation of unicast queue length. When the variation

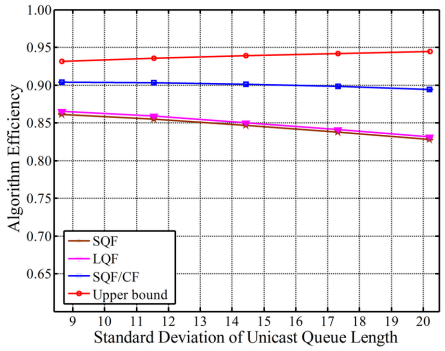

Fig. 8 Algorithm efficiency for SQF, LQF, and SQF/CF with a constant mean of unicast queue length but an increasing standard deviation of unicast queue length. Simulation results shown are with 95% confidence intervals.

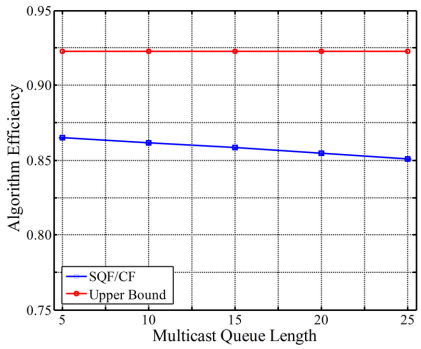

Fig. 9 Algorithm efficiency for the SQF/CF algorithm. Simulation results shown are with 95% confidence intervals.

of unicast queue length becomes larger, the efficiency for all the proposed algorithms decreases. This is because when the variation of unicast queue length is growing larger, both Type 1 and Type 2 mismatches will increase. Therefore, the algorithm efficiency will decrease when the variation of unicast queue length increases.

5.1.3 Transmission time of multicast traffic

In the third scenario, the unicast queue length still follows a uniform distribution. We set the mean and standard deviation of the unicast queue length to 45 and 14.43, respectively. In addition, the multicast queue length of a transmitting port increases from 5 to 25 in increments of 5. In this case, we test the efficiency of the SQF/CF algorithm and depict the result in Fig. 9. The result is compared with the upper bound again.

In Fig. 9, the horizontal coordinates represent the multicast queue length for a transmitting port. The vertical coordinates represent the algorithm efficiency. Fig. 9 shows how

Table 2 Time specification for each operation period

Operation period	Time
Transmission time for control messages T^D	0.1014, μs
Inter-port guard interval T^I, which includes laser off and on, automatic,gain control (AGC), clock and data recovery (CDR), and code-group alignment, intervals [14]	2 μs
Intra-port processing time T^C	10 ns
Worst-case propagation delay from a port to the coupler fabric or from the coupler fabric to a port	5 μs
Inter-frame,gap	9.6 ns
Second moment of the per-frame service time	0.3615 $(\mu s)^2$
Tunable transponders constant tuning time	4.916 μs

the algorithm efficiency changes with the transmission time of multicast traffic. When the multicast queue length is set to 25, the efficiency of the SQF/CF algorithm has an approximately 8% relative difference compared with the upper bound. We see that the efficiency of the SQF/CF algorithm decreases when the transmission time of the multicast traffic increases. This is because there will be less cut-over unicast traffic that can be sent by the multicast plane when the multicast plane has more multicast traffic to transmit. In this case, less reduction in Type 1 mismatch will occur, and the algorithm efficiency will decrease correspondingly.

5.2 System Throughput and Packet Delay Analysis

In this subsection, in order to analyze the MCDAPs performance at the system level (at which we take the MCDAP overhead into consideration.), we evaluate our protocol using the SQF/CF algorithm over an 8-port POXN/MP network using the OPNET Modeler. All the ports are connected to the same coupler fabric by a pair of 1 km fibers. We chose 8 to ensure consistency with the simulation environment in the HEDAP [14]. Moreover, simulations in this subsection were undertaken based on packet-by-packet basis. It would be rather complicated and infeasible to simulate a 48-port POXN/MP as in previous subsection. Thus, considering the simulation time, we chose 8-port POXN/MP in the following simulations.

We set up the values of the simulation parameters based on the existing EPON technology, and we employ the same setup environment as in the HEDAP [14]. It is assumed that all the transponders operate at 10 Gb/s. The values set for the system simulation parameters are given in Table 2. In this simulation model, we assume that the frame arrival rates are the same for all the transmitting ports. All the control messages (i.e., ANNOUNCEMENT, CONFIRMATION, and REQUEST messages) are 128 bytes in length. All the mentioned parameters are deterministic without variation. The discovery phase is triggered every 20000 scheduling cycles.

We first calculate the theoretical upper bounds for the efficiencies of both the HEDAP and the MCDAP. These upper bounds can be used as guidance to evaluate the performances of the two systems under more-realistic traffic scenarios. To this end, the upper bounds are calculated by considering overheads caused by protocols and physical constraints only. To be consistent with the computation in POXN [14], we set the maximum data transfer cycle to 144.96 μs for both the HEDAP and the MCDAP.

The overhead for the HEDAP consists of an inter-port guard interval, the transmission delay of REQUEST message T^D, twice 1-km propagation delays from the coupler fabric to a transmitting port, and a constant processing time T^C. Based on the parameters in Table 2, it can be calculated that the overhead time for the HEDAP within a data transfer cycle is 12.11 μs. This leads to a maximum system efficiency of $1 - 12.1114/144.96 = 0.916$. Because ports can only send sequentially in the HEDAP, the maximum per-port efficiency will be 0.115.

The overhead for the unicast plane of the MCDAP is composed of an inter-port guard interval and twice 1-km propagation delays from the coupler fabric to a transmitting port. It can be calculated that the maximum per-port overhead for the unicast plane is 12 μs. Therefore, the per-port maximum efficiency will be 0.917. Each port of the unicast plane of the MCDAP has approximately 8 times the efficiency of the port in the HEDAP.

We now consider the effects resulting from random traffic. We assume that both POXN and POXN/MP are under gated service with a maximum cycle period of 144.96 μs, which is consistent with the calculation of the upper bound. Frames arrive at each transmitting port following a Poisson process. Moreover, frames have an Ethernet format with minimum and maximum frame sizes of 64 and 1518 bytes, respectively. The frame size at each transmitting port follows the truncated geometrical distribution. The mean value of the distribution is 624.47 bytes. Thus, we obtain the mean value of the per-frame transmission time as. We use λ to denote the frame arrival rate of a transmitting port, which also represents its throughput when the system is stable. The ρ denotes the offered load to a transmitting port, which also represents the efficiency of each port.

Intuitively, the bandwidth efficiency for the MCDAP is determined by traffic load and the amount of multicast traffic. In what follows, we study how the traffic load and the amount of multicast traffic affect the mean packet delay.

Fig. 10 shows how the throughputs λ for both the HEDAP and the MCDAP systems change with the offered load ρ to a transmitting port. Fig. 10 shows that the λ of the HEDAP linearly increases until the λ of the HEDAP approaches 1.02 Gbit/s. Then, the λ of the HEDAP remains constant even though ρ increases. The maximum achievable λ of the HEDAP is achieved when ρ is 0.103, which nearly approaches its

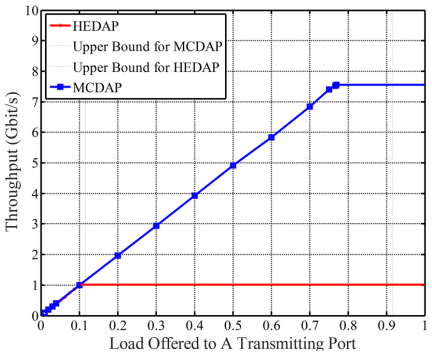

Fig. 10 Algorithm efficiency for the SQF/CF algorithm. Simulation results shown are with 95% confidence intervals.

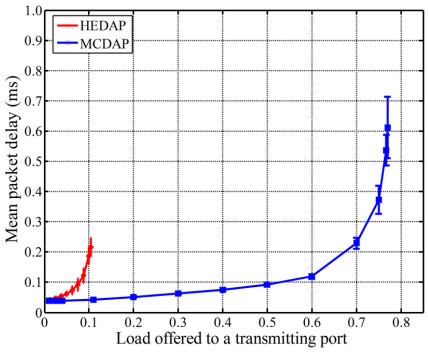

Fig. 11 Mean packet delay for the HEDAP and the MCDAP with different offered load ρ. Simulation results shown are with 95% confidence intervals

computed upper bound for maximum efficiency, with an approximately 10% relative difference. However, the maximum achievable of the MCDAP is approximately 7.56 Gbit/s, when ρ is 0.767, which is smaller than its upper bound for the maximum efficiency, with an approximately 16% relative difference. The maximum λ of the MCDAP is nearly 7.4 times that of the HEDAP because the MCDAP supports parallel communications among different receiving ports. Furthermore, the transmitting ports in the MCDAP can send traffic through the two planes simultaneously, which can make its throughput even larger. Notably, although the theoretical per-port maximum efficiency of the unicast plane of the MCDAP is 8 times the maximum efficiency of the HEDAP for each port, the actual maximum efficiency that can be achieved is 7.4 times, which is an approximately 7% difference. This results because the three situations discussed at the beginning of Section 4 may cause efficiency loss.

Fig. 11 presents how the mean packet delay in the system changes with the offered load ρ of both the HEDAP and MCDAP. Of the total load offered to a transmitting port in the MCDAP, 99% is set to be unicast traffic. The remaining

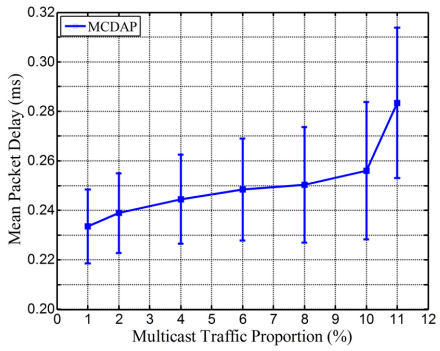

Fig. 12 Mean packet delay time of the MCDAP with different proportions of multicast traffic to total traffic. Simulation results shown are with 95% confidence intervals.

1% of the total load is multicast traffic. We will discuss the impact of the proportion of multicast traffic to total traffic on mean packet delay later. Fig. 11 shows that the maximum offered load to a transmitting port for both the HEDAP and the MCDAP are no greater than 0.103 and 0.767, respectively.

When ρ approaches the load limit (0.767) for the MCDAP, the mean packet delay of the MCDAP system sharply increases with a slight increase in ρ. Moreover, when the ρ of the HEDAP reaches its load limit, the mean packet delay of the MCDAP still increases slightly as ρ increases. This indicates that when packets in the HEDAP system are experiencing a high average queuing delay, the proposed MCDAP still works under a fully gated service. Thus, packets experience less delay in the MCDAP compared with the HEDAP under the same ρ.

Next, we study the impact of different proportions of multicast to total traffic on the mean packet delay for a transmitting port. The incast and broadcast traffic are transmitted through the multicast plane in sequence in the MCDAP system. Hence, in this simulation, we assume that the total traffic only consists of multicast and unicast traffic. Fig. 12 shows the impact of the proportion of the multicast to total traffic on the mean packet delay. We set ρ to be 0.70, where mean packet delay begins to increase dramatically after this point in Fig. 11. The horizontal coordinates represent the proportion of the multicast to total traffic. They reveal that when the proportion of the multicast to total traffic increases to a value larger than 11%, the mean packet delay will increase dramatically. This is because when the proportion of multicast to total traffic increases, the requested bandwidth during a data transfer cycle for the multicast plane increases accordingly. Hence, there will not be any extra bandwidth capacity remaining in the multicast plane. This will reduce the efficiency of the adopted SQF/CF algorithm.

Meanwhile, the unicast plane will finish the current data transfer cycle earlier because the amount of unicast traffic decreases accordingly. However, the unicast plane needs to wait until all the connected ports receive all the REQUEST messages through the multicast plane before it can start the next data transfer cycle. Therefore, the unicast plane needs to wait for the multicast plane to complete. This leads to an increase in the idle time between the two adjacent data transfer cycles for the unicast plane, which results in an increase in mean packet delay when the multicast traffic increases.

Moreover, Fig. 12 shows that when the proportion of multicast to total traffic is less than 10%, the mean packet delay of the MCDAP system remains nearly constant. This results because the multicast plane in this case has extra bandwidth capacity so that it can transmit the cut-over unicast traffic to achieve load balance. Thus, both the unicast and multicast planes finish their data transfer cycles at nearly the same time, which helps reduce wasted bandwidth. Hence, it can be seen that the MCDAP enables dynamic traffic allocation to accommodate dynamic traffic patterns.

6 Related Work

In this section, we highlight our contributions by comparing our work with existing research. Our main contributions can be summarized in three aspects. First, we proposed a high-throughput passive optical cross-connection network with multiple planes, where different planes are designed for different traffic patterns. Second, we proposed a distributed protocol that enables collision-free transmission and dynamic traffic allocation between the two planes. Third, we formulated our scheduling problem as a mathematical programming problem and then designed heuristic algorithms that not only enable transmissions without collisions, but also optimize bandwidth utilization for the unicast plane. The following paragraphs discuss works related to these aspects.

6.1 Switch Fabrics

Recently, many attempts have been made to develop best practices for datacenter networks utilizing different optical technologies [9–14]. One major trend is the use of optical circuit switching based on devices, such as MEMS switches and wavelength selective switches [8, 9, 24, 25]. One of the severe drawbacks of optical circuit switching-based designs is the slow reconfiguration time, which makes it unsuitable to accommodate the highly dynamic and changing traffic patterns characterized in datacenter networks. The other drawback is that translating multicast and broadcast traffic into unrelated unicast flows will greatly reduce the effectiveness of the underlying networks for sending redundant traffic. In our work, multicast/incast traffic is transmitted through the multicast plane utilizing the coupler fabrics, where signals are transmitted in a broadcast fashion without the need for

sending multiple copies. Moreover, our network reconfigures an optical point-to-point circuit for unicast traffic utilizing faster tunable transponders. These transponders enable tuning from one wavelength to another at a granularity of 5 μs (or smaller), which is significantly faster than the slow switching time of optical circuit switching-based networks.

A tremendous amount of research has been done in the area of optical burst-switching (OBS) networks (e.g., [26, 27]). OBS networks are based on active optical switches which introduce long switching times in the range of 10 milliseconds [14]. To mitigate this overhead, packets must be accumulated into a burst, therefore leading to more delays, before a lightpath can be setup through a signaling process. This is known as delayed reservation. The active optical switches also make OBS networks very hard to support multicast traffic. In contrast, our switch fabric is based on purely passive optical broadcast devices which can support multicast and broadcast traffic naturally. Our MCDAP protocol also allows fast resource reservation with delays in the sub-millisecond range as shown in Subsection 5.2.

An unconventional approach has been proposed wherein different passive optical gadgets are designed and attached to a relevant ToR optical space switch to accommodate various traffic patterns found in datacenter networks [5]. This approach achieves the goal of providing a physical layer that is intrinsically compatible with mixed traffic patterns. However, different optical gadgets must be employed at the physical layer to address various traffic patterns, which construct a rather complex physical layer. Moreover, the traffic patterns generated within a datacenter are unpredictable. Hence, some deployed optical gadgets designed have to be overprovisioned. Finally, because of the variability of datacenter traffic, it is extremely challenging to achieve flexible reconfigurations through hardware. In our proposal, passive coupler fabrics are used to construct the physical interconnections. On the one hand, a multicast traffic pattern is naturally supported by the multicast plane; on the other hand, the unicast plane enables efficient transmission of unicast traffic. Flexibility is enabled at the link layer by dynamically adjusting the traffic between the two planes.

6.2 Access Protocols

Access protocols are widely used as the building blocks in many access networks, such as Wi-Fi [28], EPONs [29], and cellular networks. Access protocols can be classified into two categories: random and scheduled. Scheduled access protocols can be further divided into two classes: polling and reservation. Many polling and reservation protocols are typically centralized access protocols, where one of the nodes works as the master node to handle resource scheduling and collision avoidance. These protocols are implemented in networks that have a hierarchical physical structure so that the nodes that deploy at the high level of the hierarchy naturally serve as the master. Unfortunately, such a master-slave hierarchy is not suitable for datacenter networks. We designed the MCDAP, which is a distributed access protocol, to address the link layer contention problems of POXN/MP.

6.3 Schedule Algorithm

Given a traffic matrix, Edmonds algorithm [30] is adopted in C-through [10] to determine how to connect the server racks by optical paths in order to maximize traffic volume offloaded to the optical network. The traffic matrix is a graph $G = (E,V)$. V is the vertex set, in which each vertex denotes a rack, and E is the edge set. The weight of an edge $e, w(e)$ is the traffic volume between the racks. A matching is a set of edges where each vertex is incident with at most one edge. Here, a matching represents a cross-connect pattern of an optical switch. The problem of carrying the maximum amount of traffic with an optical switch can be summarized as finding a matching with maximum aggregated weight. The remaining traffic demands will be carried by electronic switches. The approach is optimal in the sense that it can offload traffic from electronic switches to the highest degree. However, the scheduling problem in this paper requires that the POXN/MP carry the demands between all transmitting and receiving ports during a cycle period. This means that a transmitting port needs to be connected to multiple receiving ports at different times during a cycle period, which makes Edmonds algorithm inapplicable. One might think that we can divide a cycle into multiple rounds and apply Edmonds algorithm at the beginning of each round. To do so, a new round can only start after all optical ports have finished transmitting all scheduled amounts of data for the current round so that each round is a fresh start. However, the realignment process at the end of each round will make the POXN/MP less efficient over the period of a cycle because ports finishing earlier have to wait until all ports finish.

Another problem with Edmonds algorithm is its processing time requirement. Although Edmonds algorithm can be completed in polynomial time [30], it is still too complicated to be executed within a round time, which is extremely small (e.g., submicron-second range), as we discussed earlier.

7 Conclusion

We proposed a novel datacenter network called POXN/MP, an all-optical architecture with high energy efficiency that supports communication parallelism for unicast traffic to address the insufficient bandwidth problems in POXN [14]. Meanwhile, our network efficiently enables multicast/incast traffic transmission through the multicast plane, similar to

[14]. In addition, it achieves dynamic traffic allocation to cater to the fluctuation of datacenter traffic patterns.

The tunable transponders deployed in POXN/MP have a switching time on the order of a few microseconds. Thus, the proposed scheme can be quickly reconfigured to meet traffic fluctuations, as opposed to the slow reconfiguration time of active optical circuit switch-based networks. When taking advantage of passive rather than active optical devices, we achieve low cost and high energy efficiency among interconnected ports. To solve the link layer collision problem, we propose the MCDAP, a fully distributed protocol that enables dynamic traffic allocation, which is suitable to the bursty nature of datacenter traffic patterns.

The scheduling problem for the unicast plane was formulated as a mixed integer programming problem. Three heuristic algorithms were proposed to speed up the computing process.

The simulation results show significant performance increases for both the system throughput and the mean packet delay compared with POXN. Thus, POXN/MP is more suitable for datacenter networks where traffic is bursty with high temporary peaks. The simulation results also show that the SQF/CF is the most promising scheduling algorithm to take advantage of both the unicast and multicast planes and to achieve dynamic load distribution between the two planes.

References

1. J. Dean and G. Sanjay, "MapReduce: simplified data processing on large clusters.", *Communications of the ACM*, vol. 51, no. 1, pp. 107-113, 2008. Volume, page numbers (year)
2. H. Liu, C. F. Lam, and C. Johnson, Scaling optical interconnects in datacenter networks opportunities and challenges for WDM, *2010 18th IEEE Symp. High Perf. Interconnects,* 2010, pp. 11316.
3. 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. 267280.
4. T. Benson,*et al.*, "Understanding data center traffic characteristics." *ACM SIGCOMM Computer Communication Review,* vol. 40, no. 1, pp. 92-99, 2010.
5. H. wang, *et al.*, 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. 5358, Jul. 2013.
6. A. Greenberg, *et al.*, VL2: A scalable and flexible data center network, *in Proc. ACM SIGCOMM Data Commun.*, 2009, pp. 5162.
7. M. Al-Fares, A. Loukissas, and A. Vahdat, A scalable, commodity data center network architecture, *in Proc. ACM SIGCOMM Data Commun.*, 2008, pp. 6374.
8. C. Guo *et al.*, BCube: A high performance, server-centric network architecture formodular data centers, *in Proc. ACMSIGCOMM Conf.*, 2010, pp. 339350.
9. N. Farrington, *et al.*,Helios:Ahybrid electrical/optical switch architecture formodular data centers, *in Proc. ACMSIGCOMM Conf.*, 2010, pp. 339350.
10. G.Wang *et al.*, c-through: part-time optics in data centers, *in Proc. ACM SIGCOMM Conf.*, 2010, pp. 327338.
11. A. Singla, *et al.*, "Proteus: a topology malleable data center network." *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, 2010.
12. C. Kachris and T. Ioannis, "A survey on optical interconnects for data centers." *Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 1021-1036, 2012.
13. C. Kachris, K. Konstantinos and T. Ioannis, "Optical interconnection networks in data centers: recent trends and future challenges." *Communications Magazine*, vol. 51, no.9, pp. 39-45, 2013.
14. W. Ni, C. Huang, and *et al.*, POXN: A new passive optical cross-connection network for low-cost power-efficient datacenters, *Journal of Lightwave Technology*, vol. 32, no. 8, pp. 14821500, 2014.
15. O. A. Lavrova, G. Rossi, and D. J. Blumenthal, Rapid tunable transmitter with large number of ITU channels accessible in less than 5 ns, *in Proc. 26th Eur. Conf. Optical Communications*, pp. 2324, 2000.
16. C. F. Lam, (Ed.), Passive optical networks: principles and practice. Academic Press, 2011.
17. A. Hasegawa and Y. Kodama, "Signal transmission by optical solitons in monomode fiber." *Proceedings of the IEEE*, vol. 69, no.9, pp. 1145-1150.
18. P. Dumon, *et al.*, "Compact wavelength router based on a silicon-on-insulator arrayed waveguide grating pigtailed to a fiber array." *Optics express*, vol. 14, no. 2, pp. 664-669, 2006.
19. Cisco Nexus 3548 and 3524 Switches Data Sheet. Cisco Corp.
20. www.fiberstore.com/c/10g-sfp-plus_63
21. Cisco 10GBASE SFP+ Modules Data Sheet. Cisco Corp.
22. Cisco 10GBASE Dense Wavelength-Division Multiplexing SFP+ Modules Data Sheet. Cisco Corp.
23. C. E. Hopps, Analysis of an equal-cost multi-path algorithm. 2000.
24. C. F. Lam, *et al.*, "Fiber optic communication technologies: What's needed for datacenter network operations." *IEEE Communications Magazine*, vol. 48, no. 7,pp. 32-39, 2010.
25. G. Porter, *et al.*, Integrating microsecond circuit switching into the data center. vol. 43, no. 4, 2013.
26. C. Qiao and M.Yoo, Optical burst switching(OBS)-A new paradigm dor an opitcal internet, *J. High Speed Networks*, vol. 8, pp.69-84, 1999.
27. Y. Xiong, *et al.*, Control Architecture in Optical Burst-Switched WDM Netowrks , *IEEE JSAC*, vol. 18, Oct. 2000, pp. 1838-1851.
28. B. P. Crow, *et al.*, "IEEE 802.11 wireless local area networks." , *Communications Magazine*, vol. 35, no. 9, pp. 116-126, 1997.
29. H. Takagi, "Analysis and application of polling models." In Performance Evaluation: Origins and Directions, *Springer Berlin Heidelberg*, 2000, pp. 423-442.
30. J. Edmonds. "Paths, trees, and flowers." *Canadian Journal of mathematics, vol*. 17, no. 3, pp. 449-467, 1965.

Yang An received the M.A.Sc. in Electronic and Computer Engineering from Carleton University, Ottawa, Canada, in 2016. He is currently a junior Deep Packet Inspection Engineering at Nokia in Ottawa, ON, Canada. He was a Research Assistant at the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, Canada. His research interests include optical network design, simulation and analysis, and datacenter networking.

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, Canada, where he was a System Engineering Specialist. He was a System Engineering 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.

Author's Photo (black-and-white preferred)

Click here to download Author's Photo (black-and-white preferred) yang.jpeg

Author's Photo (black-and-white preferred)

Click here to download Author's Photo (black-and-white preferred) Dr. Huang.png