Photonic Network Communications (2021) 42:93–104
https://doi.org/10.1007/s11107-021-00933-z

ORIGINAL PAPER

A novel bandwidth allocation scheme for OTSS-enabled flex-grid intra-datacenter networks

Lin Wang[1] · Xinbo Wang[1] · Massimo Tornatore[1,2] · Kwangjoon Kim[3] · Biswanath Mukherjee[1]

Received: 19 May 2020 / Accepted: 27 March 2021 / Published online: 24 August 2021

Abstract

Optical circuit switching networks have been recognized as a promising solution for *inter*-datacenter networks. However, for *intra*-datacenter networks, they may fall short in efficiently provisioning traffic requests due to their relatively coarse-grained channel assignment and special intra-datacenter traffic patterns. Optical time slice switching (OTSS) has been recently proposed as an optical-switching technique that can provide flexible and transparent optical circuits by extending the merit of flex-grid switching to the time domain, thus achieving much finer granularity. As OTSS requires nanosecond speed optical switches which are expensive, it might not be economically viable to make a one-time upgrade for the entire datacenter. Thus, we expect fine-grained OTSS-enabled and coarse-grained flex-grid-enabled optical switching techniques to co-exist in the foreseeable future. In this study, we investigate an OTSS-enabled flex-grid (OTSS-FG) architecture for intra-datacenter networks. For scenarios where traffic flows are given, we develop a Mixed Integer Linear Program to study the optimal bandwidth allocation scheme in an OTSS-FG architecture. When traffic flows are generated in real time, by leveraging machine-learning techniques to detect flow types, we propose a flow-aware bandwidth allocation (FABA) scheme and a dynamic version of FABA, called "D-FABA" scheme. Numerical simulations show that proposed bandwidth allocation scheme can outperform benchmark schemes in terms of average delay and blocking probability.

Keywords Optical time slice switching (OTSS) · Flex-grid · Bandwidth allocation · Machine learning

1 Introduction

According to Cisco's forecast in [1], global datacenter (DC) traffic will reach 3.3 ZB per year in 2021, increasing at an annual rate of 31%, and 76% of the traffic are transmitted within DCs, known as "intra-DC" traffic. Currently, intra-DC networks are mostly equipped with electrical switches, which have high power consumption, limited scalability and high latency, resulting in severe challenges for datacenter operators. These challenges are becoming more arduous, as many of modern applications require to exchange huge volume of data for real-time tasks with rigid latency constraints. These tasks or applications usually need to be conducted within a DC, which leads to characteristic traffic patterns.

Studies [2, 3] show that a small proportion of intra-DC traffic is carried by a large proportion of flows. Specifically, these "mice flows" carry less than 1 MB data per flow, but account for 90% of the number of traffic flows. Another type of flows usually carries more than 100 MB data per flow, occupying more than 90% of traffic amount, so-called elephant flows. Mice flows usually carry small-size packets with short duration and high time sensitivity, e.g., transactional traffic, web browsing and search queries. Elephant flows generally carry large-size packets and have long-lasting duration. Examples are bulk data transfer, data backup and virtual machine migration. Hence, intra-DC networks need a high-speed, scalable, low-latency and energy-efficient switching architecture, which, as several studies have anticipated, can be provided by advanced optical switching technologies [4]. For example, Helios architecture [5] adopted optical circuit switching network for coarse-grained intra-DC traffic, while leaving part of the fine-grained switching fabrics operating with electrical solutions. Ref. [6] proposed

✉ Lin Wang
amlwang@ucdavis.edu

[1] University of California Davis, Davis, USA

[2] Politecnico Di Milano, Milano, Italy

[3] Electronics and Telecommunications Research Institute, Gwangju, Korea

another intra-DC architecture, called OSA, which provides on-demand network topology according to traffic patterns, but traffic demands are usually smaller than its minimal channel capacity, leaving space for improving network utilization.

Wavelength-division multiplexing switching networks have been proposed as a promising optical circuit switching technique for intra-DC networks. By using on-demand spectrum assignment and adaptive modulation formats, flex-grid (FG) networks can significantly improve spectrum efficiency and increase network capacity. FG networks enable optical bypass and relax requirement for routers, including buffering and switching of the transit traffic. Reference [7] provided an optical slot switching-based architecture by combining optical circuit switching and elastic, burst-mode transponders, with variable bit rate depending on node count for intra-DC networks. Besides, optical packet switching (OPS) has also attracted attention in intra-DC settings, as it can provide finer granularity (at packet level), but it faces major limitations due to immature technologies for optical buffering [8].

However, all existing proposals for intra-DC optical switching technologies based on optical networks fail to consider the aforementioned special traffic patterns, so they fall short in provisioning intra-DC network traffic efficiently. Existing bandwidth-allocation schemes [9, 10] treat each flow equally, without considering distinct requirements of flows, e.g., mice flows and elephant flows. This might degrade the performance of the entire network. For example, for a typical FG network, channel capacity is typically 10 Gbps or even larger at higher-order modulation. However, as mice flows are short-lived and small-sized, assigning such a large channel to mice flows leads to a waste of precious bandwidth resources, thus decreasing throughput and increasing latency of incoming flows. Therefore, a more fine-grained optical switching technique is needed for intra-datacenter networks.

Optical time slice switching (OTSS) [11, 12] has been recently proposed as a promising solution for fine-grained optical switching, thanks to the introduction of novel technologies such as high-precision network time synchronization [13] and nanosecond speed optical switches [14]. OTSS can provide flexible, fine-grained, transparent optical circuits by dividing an optical transmission channel into repetitive OTSS frames in time domain that can be allocated to OTSS nodes with high-precision network time synchronization and nanosecond speed optical switches, which makes it achieve much finer granularity than channel switching. OTSS shares some similarities with existing time-based optical switching technologies such as fractional lambda switching and time-driven switching (TDS) [15]. But OTSS can provide a fully gridless time slice allocation scheme thanks to high-precision network time synchronization, while TDS has limitation on the minimum time slot it can provide. Besides, OTSS does not require large buffer which could be expensive for large-scale TDS-based DC. While OTSS is a promising solution, it may not be economically viable to make a one-time complete upgrade from FG to OTSS technology for the entire DC, due to the requirement of expensive nanosecond-speed optical switches [14]. However, to satisfy the increasing data requirements in intra-datacenter networks, we envision that fine-grained optical switching techniques such as OTSS may co-exist with coarse-grained optical switching technique, such as flex grid (FG), for intra-DC networks in future.

Therefore, in this study, we consider an OTSS-enabled flex-grid (OTSS-FG) architecture for intra-DC networks and propose two novel schemes for allocating bandwidth resources in time and channel domains. In this way, we can make use of OTSS technology to provide fine-grained optical switching for mice flows to improve network performance while still using FG switching for elephant flows to control the total cost. Our proposed bandwidth-allocation schemes reserve a set of channels as OTSS channels to accommodate mice-flow traffic, while leaving other coarse-grid channels, such as FG channels, for elephant-flow traffic. For scenario when traffic flows are given, we develop a Mixed Integer Linear Program (MILP) to mathematically model the optimal bandwidth allocation scheme in OTSS-FG architecture. However, in practice, when traffic flows are generated in real time, in order to achieve flow-aware bandwidth allocation, the scheduling scheme must be equipped with the capability to distinguish between mice flows and elephant flows. We employ the classification methods for flow detection that were proposed in our previous work [16], i.e., C4.5 decision tree, which was shown to be able to achieve 95% accuracy for elephant-flow detection. Armed with the function of flow detection, we design a flow-aware bandwidth allocation (FABA) scheme for OTSS-FG architecture to treat traffic flows differently and assign OTSS or FG bandwidth resources properly. We also provide a dynamic version of FABA (called D-FABA) to adaptively adjust bandwidth allocation scheduling based on current traffic flow status so as to further improve the performance of the OTSS-FG architecture. We conduct simulations to study the performance of the MILP for optimal bandwidth allocation, for the proposed FABA scheme, and D-FABA scheme, in terms of blocking probability and average latency, to prove our proposed bandwidth allocation scheme can outperform benchmark schemes on the OTSS-FG architecture.

The rest of the study is organized as follows: Section II describes the envisioned OTSS-FG architecture and evaluates several machine-learning-based flow-detection methods. Section III introduces mathematical models for three scenarios of OTSS channel and FG channel in static network case and studies the performance improvement of OTSS-FG architecture. Section IV gives two scheduling schemes, i.e.,

FABA for static traffic and D-FABA for dynamic traffic. Section V conducts numerical study on the performance of the proposed scheduling schemes via simulations. Section VI concludes this study.

1.1 OTSS-enabled flex-grid architecture

A. State-of-the-Art Switching Techniques

Flex-grid (FG) optical networking has been proposed as an optical switching technique to enhance flexibility in optical spectrum assignment [9, 10]. However, considering intra-DC traffic flow characteristics in intra-DC, FG switching falls short in efficient usage of bandwidth resources in intra-DC networks, especially for mice-flow traffic. Mice flows carry less than 1 MB data, while advanced FG switching can minimally provide 6.25 GHz channel, which is too large for mice flow and would lead to a waste of bandwidth resources. Thus, even finer granularity is needed to serve some small traffic flows in intra-DC system. As described in Introduction, OPS can provide finer granularity (at packet level), but it faces major limitations due to immature technologies for optical buffering for large-scale datacenter, so we do not consider OPS. In this study, therefore, we consider OTSS, a recently proposed optical switching technology that exploits time domain to provide more transparent and fine-grained connections [11, 12]. In OTSS, the optical transmission channels are organized into repetitive OTSS frames in time domain. Each OTSS frame contains one or several variable-length time slice(s) for data transmission, and each time slice occupies one timeslot. When a time slice arrives at a switching node, the pre-set (periodic) control signals are sent to OTSS fabric at precise time to direct the time slice to the expected output port. To guarantee high-precision timing, time synchronization of all OTSS nodes is required. Reference [13] reported a high-precision network time synchronization with an accuracy of 65 ns realized under 13 synchronization hops over commercial transport networks. Commercial fast switches (e.g., (Pb,La)(Zr,To)O_3 (PLZT) switch) make OTSS a reliable technology, and experimental demonstration has been conducted in [14]. It is expected that this accuracy can be further reduced below 10 ns in the next few years. The high-precision network time synchronization and nanosecond speed optical switches make it possible for OTSS to achieve much finer granularity than channel switching of FG network.

B. OTSS-enabled flex-grid (OTSS-FG) switching architecture

Considering characteristics of mice flows, i.e., requiring small bandwidth, low latency and low blocking probability, OTSS switching is a promising solution, thanks to its fine-grained time slice(s). But, as mentioned before, considering the requirement of expensive nanosecond fast optical switching, it is not realistic to apply OTSS to all traffic. Therefore, we apply conventional FG switching, which can provide comparatively coarse-grained channels with lower-price optical switch, on those flows containing large amounts of data, long-lasting transmission and low time sensitivity, i.e., elephant flows.

To integrate the merits of OTSS and FG switching, we propose an OTSS-FG architecture for intra-DC networks as shown in Fig. 1. As shown in Fig. 1a, a fiber consists of multiple (k) channels. We reserve n out of k channels to transmit mice flows through OTSS switching (called "OTSS channel"), while we apply FG switching technology on the $k - n$ channels to transmit elephant flows (called "FG channel"). OTSS channels are divided into repetitive OTSS frames in time domain that can be allocated to OTSS nodes as shown in Fig. 1b (i.e., λ_4, λ_5). To realize fine-grained frames in time domain, OTSS requires high-precision network time synchronization provided by a network controller through control messages in Fig. 1c.

We also assume that a centralized controller provides the following functions:

- Collect elephant traffic flow information from end servers. To avoid high monitoring overhead, which will consume significant switch resources, and/or have long detection times, we detect elephant flows at the end servers [17]. We do this by observing the end servers' socket buffers, which provide more efficient visibility of flow behavior shown on the left side of Fig. 1a. Once an elephant flow is detected, an end server notifies the centralized controller using in-band signaling with low overhead.
- Provide accurate network-scale global time synchronization which is required by OTSS to guarantee that all switches have the same time coordinate to perform switching operations at specific time. To realize this, we define the format of control message sent from centralized controller to network switches. As shown in Fig. 1c, the control message contains OTSS frame length, operation time and type, fiber ID, channel, traffic ID, source and destination information.
- Provide routing and channel assignment for elephant flows transmitting on FG channels. We can apply conventional FG routing and channel assignment schemes in [18].
- Provide routing and timeslot assignment (RTA) for flows on OTSS channels. Figure 1b demonstrates repetitive OTSS frame format which contains a fixed number of timeslots. As OTSS can provide flexible, fine-grained, transparent optical circuits by using timeslots, it can be regarded as optical circuit switching in temporal domain. We need to

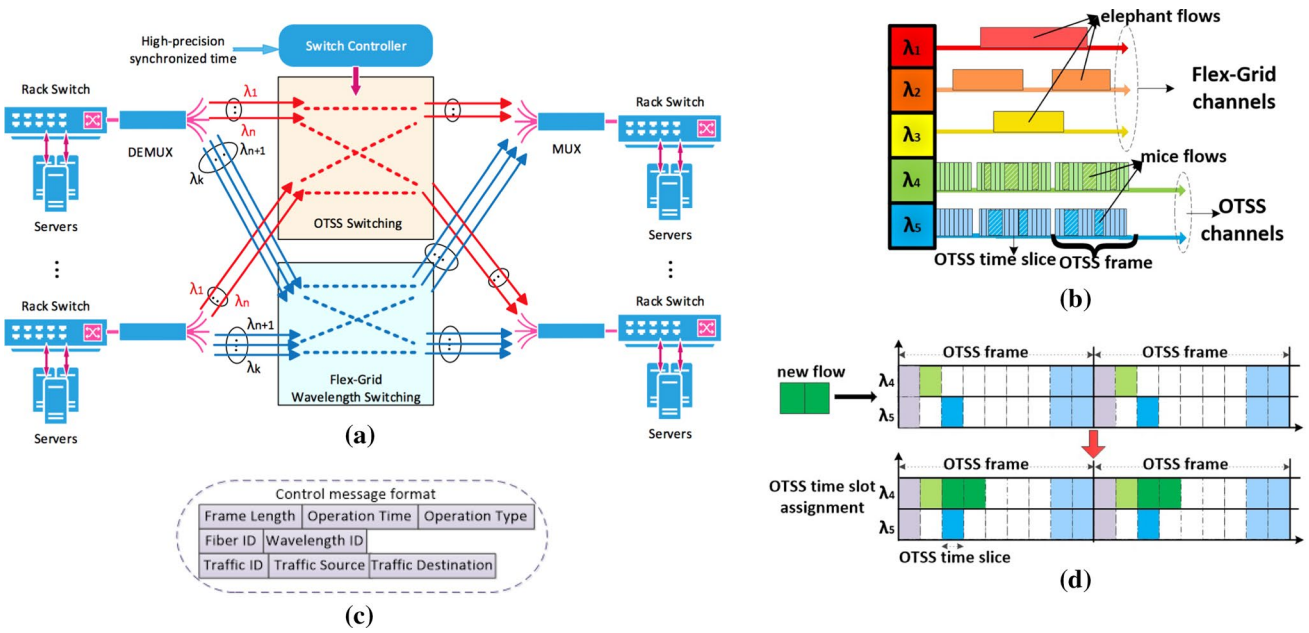

Fig. 1 **a** OTSS-enabled flex-grid network architecture; **b** channel assignment; **c** control message format; **d** OTSS frame timeslot

devise new RTA schemes which will be described in Section IV.

C. Machine learning-based flow detection

As introduced in Section I, we assume that traffic flows in a datacenter mainly consist of mice flows and elephant flow. To satisfy different requirements of mice flows and elephant flows, the OTSS-FG architecture should be able to classify an incoming flow and detect if it is an elephant flow or mice flow. Therefore, a real-time flow detection method must be devised.

In [16], we studied machine learning methods of flow detection for intra-DC networks. We achieved 95% accuracy of detecting elephant flows by applying C4.5 Decision Tree (C4.5) machine learning method. Besides, C4.5 can achieve fast detection and needs low computation requirements; thus, it is very suitable for large-scale intra-DC architectures. We use sliding windows as in [17] to obtain packets information from traffic flows and apply machine learning methods (e.g., C4.5) to detect elephant flows at each end server. We envision that our flow-detection technique can enable a flow-aware architecture for the intra-DC network.

2 Mathematical modeling For OTSS switching

In this section, we use a mathematical model to compare the throughput achievable in three intra-DC network scenarios, i.e., FG architecture, OTSS-FG flow-unaware architecture, OTSS-FG flow-aware architecture. Specifically, we assume that a flow-aware architecture can differentiate if a flow is mice flow or elephant flow, and thus it is possible to assign mice flows to OTSS channel, while a flow-unaware architecture will assign mice flows randomly to either an OTSS channel or a FG channel. This will allow us to study how flow-awareness (flow detection) can help improve the performance of OTSS-FG architecture.

We mathematically model the FG channel and the OTSS channel for the three scenarios. For FG channel, since there are extensive studies on how to formulate its ILP model (e.g., in [18]), we do not present it here, but we directly apply it in our simulations and treat the scenario of FG architecture as a benchmark. For the scenarios of OTSS-FG flow-unaware architecture and OTSS-FG flow-aware architecture, we assume that they both consist of

OTSS channels and FG channels, and their only difference is the input, i.e., we assign mice flows only to OTSS channel in the flow-aware architecture, while we assign mice flows randomly to both OTSS and FG channels in the flow-unaware architecture.

A. Mathematical model for one OTSS channel

The following MILP mathematically models optimal bandwidth allocation on one OTSS channel (i.e., as mice flows occupy small portion of traffic in intra-DC networks, and one OTSS channel is enough to accommodate all mice flows, we do not consider multiple OTSS channels allocation in this model for simplicity purpose).

2.1 Given

- $G(N, E)$: network topology in a unidirectional graph, where N and E denote set of nodes and fiber links.
- R: set of traffic requests.
- s_r, d_r, b_r: source, destination and required bandwidth of traffic request$r$,$r \in R$. Here, bandwidth is calculated in terms of the number of timeslots.
- η_1, η_2: parameters for optimization sequence.
- (i, j): fiber link.
- Max: a maximum number.

2.2 Variables

- $\lambda_{(i,j)}^{r,t}$: binary variable, which equals 1 if request r occupies timeslot t on fiber link (i, j).
- ρ_r: binary variable, which equals 1 if request r is accepted.

Then, network throughput can be described as:

$$T = \sum_{r \in R} \rho_r * b_r \tag{1}$$

The total used network resources can be described as:

$$R = \sum_{r \in R} \sum_{t \in T} \sum_{(i,j) \in E} \lambda_{(i,j)}^{r,t} \tag{2}$$

2.3 Objective

We consider network throughput as the most important metric for running a network, so we maximize network throughput first and then minimize resource usage, i.e.,$\eta_1 \gg \eta_2$ So, the objective is to maximize

$$\eta_1 * T - \eta_2 * R \tag{3}$$

$$\sum_{j \in N} \sum_{t \in T} \lambda_{(i,j)}^{r,t} - \sum_{j \in N} \sum_{t \in T} \lambda_{(j,i)}^{r,t} = \begin{cases} \rho_r * b_r, i = s_r \\ -\rho_r * b_r, i = d_r \\ 0, \text{otherwie} \end{cases} \quad \forall r \in R \tag{4}$$

$$\sum_{t \in T} \lambda_{(i,j)}^{r,t} \lambda_{(i,j)}^{r,t} = \lambda_{(j,k)}^{r,t} \times 1 - 1 \geq 0 \quad \forall r \in R, \forall t \int T, \forall (i,j), (j,k) \in E \tag{5}$$

$$\sum_{r \in R} \lambda_{(i,j)}^{r,t} \leq 1 \quad \forall t \int T, \forall (i,j), (j,k) \in E \tag{6}$$

Equation (4) is the flow-conservation equation for request routing. For a request traversing multiple fiber links, it should obey the time-slice continuity constraints, which means the same time slices should be adopted along fiber links of the routing path, as Eq. (5) constraint. Note that $\langle \cdot \rangle \times 1$ represents the logical calculation which returns 1 if the expression inside the angle brackets is true. If link (i, j) is occupied by request r at timeslot t (i.e., $\lambda_{(i,j)}^{r,t} = 1$), then link (j, k) must also be occupied by request r at timeslot t to satisfy Eq. (5). On the other hand, if link (i, j) is not occupied by request r, (i.e., $\lambda_{(i,j)}^{r,t} = 0$), then $\lambda_{(j,k)}^{r,t}$ can either be 0 or 1. However, as the objective wants to minimize resource usage in Eq. (3), $\lambda_{(j,k)}^{r,t}$ will be 0. Note that we do not consider propagation delay of optical intra-DC networks in this study.[1] Equation (6) ensures that a time slice of a link can be used only once.

B. *Model Linearization*

Equation (5) is nonlinear and makes the problem a mixed integer quadratic constraint program, which is hard to solve. We can linearize it with extra variables and constraints to reduce it to a mixed linear integer program (MILP) as follows.

2.4 Linearization variables

$u_{(i,j)}^r$: binary, which equals 1 if $\sum_{t \in T} \lambda_{(i,j)}^{r,t}$ is positive. $u_{(i,j)}^r$ can be calculated as:

$$\sum_{t \in T} \lambda_{(i,j)}^{r,t} / \text{Max} \leq u_{(i,j)}^r \leq \sum_{t \in T} \lambda_{(i,j)}^{r,t} \forall r \in R, (i,j) \in E \tag{7}$$

[1] Generally, a traffic flow traversing multiple fibers will have propagation delay. Thus, one traffic flow should occupy different time slices on different fibers (i.e., time slice shift). Considering the short-reach optical intra-DC networks (fiber links are usually a few hundreds of meters), the time slice shift can be regarded as negligible and we do not consider propagation delay in this work.

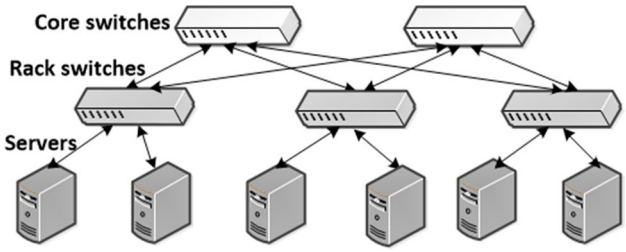

Fig. 2 A fat tree datacenter topology

$\chi^{r,t}_{(i,j),(j,k)}$: binary, which equals $\left(\left\langle u^{r}_{(i,j)} \times \lambda^{r,t}_{(i,j)} = \lambda^{r,t}_{(j,k)} \right\rangle \times 1\right)$

.

Then, $\chi^{r,t}_{(i,j),(j,k)}$ can be calculated as:

$$\chi^{r,t}_{(i,j),(j,k)} \geq u^{r}_{(i,j)} + \left(\left\langle \lambda^{r,t}_{(i,j)} = \lambda^{r,t}_{(j,k)} \right\rangle \times 1 - 1\right)$$

$$\forall r \in R, t \in T, (i,j), (j,k) \in E \tag{8}$$

$$\chi^{r,t}_{(i,j),(j,k)} \leq u^{r}_{(i,j)}, \forall r \in R, t \in T, (i,j), (j,k) \in E \tag{9}$$

$$\chi^{r,t}_{(i,j),(j,k)} \leq \left(\left\langle \lambda^{r,t}_{(i,j)} = \lambda^{r,t}_{(j,k)} \times 1 \right\rangle\right), \forall r \in R, t \in T, (i,j), (j,k) \in E \tag{10}$$

2.5 Linearization constraints

Equation (5) can be linearized into Eq. (11), accompanied by Eqns. (8)-(10)

$$\chi^{r,t}_{(i,j),(j,k)} - u^{r}_{(i,j)} \geq 0, \forall r \in R, t \in T, (i,j)(j,k) \in E \tag{11}$$

C. Performance evaluation for MILP model

We solve the MILP problem using a commercial IBM CPLEX platform. We adopt a small-scale fat-tree datacenter topology as shown in Fig. 2 due to scalability limitation of the MILP. A fiber supports 4 channels. Each channel has a capacity of 10 Gbps. The frame length of OTSS is set to be 20 *ms*, and the minimum time slice is 2 *ms*.

We simulate three architectures, i.e., FG, flow-unaware OTSS-FG and flow-aware OTSS-FG. FG is treated as a benchmark to study the performance improvement of the other two architectures. Considering problem complexity, we give optimal solution to a special case of OTSS-FG architecture which contains only one OTSS channel and three FG channels. We compare the two OTSS-FG architectures with a conventional FG architecture containing four FG channels. Traffic requests are generated uniformly between edge switch pairs as an input to the simulation. Specifically, mice flows are randomly assigned to any channel in FG architecture and flow-unaware OTSS-FG architecture, while dedicatedly assigned to flow-aware OTSS-FG architecture.

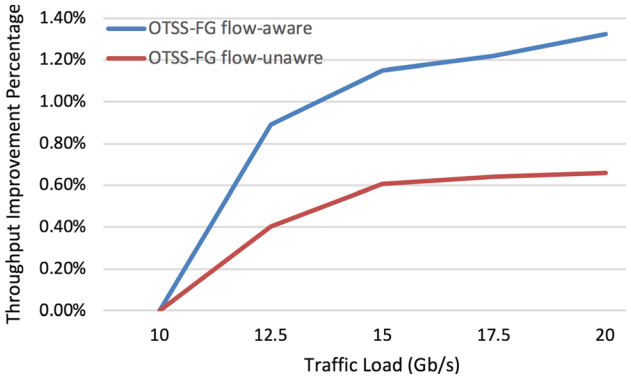

Fig. 3 Network throughput improvement vs. traffic load

Figure 3 shows throughput improvement percentage obtained by comparing OTSS-FG flow-aware and OTSS-FG flow-unaware architecture to FG architecture, under different traffic loads. For each traffic load, we pre-generate all flows as input of which 10% are elephant flows occupying more than 90% of total bytes, while 90% are mice flows occupying just total 10% of bytes. We see that, at low traffic loads (i.e., up to load of 10 Gb/s), there is no throughput improvement for OTSS-FG flow-aware and OTSS-FG flow-unaware models. This is so because all requests can be satisfied, and throughput almost equals to traffic load, meaning there is almost no blocked traffic. As traffic load grows, OTSS-FG flow-aware achieves higher throughout than the other. This is so because OTSS-FG flow-unaware model does not differentiate flow types, and thus mice flow might be scheduled to transmit on FG channel, which can be blocked by elephant flows and leads to low channel utilization. To maximize throughput, some mice flows might be blocked by elephant flows. Similarly, as there is no OTSS channel in FG model and Eq. (3) aims to maximize total throughput, more mice flows will be blocked by elephant flows leading to lower throughput than OTSS-FG architectures.

3 Flow-aware bandwidth allocation scheme for OTSS-FG architecture

A. Flow-Aware Bandwidth Allocation Scheme

As mice flows require only small bandwidth and are typically latency-sensitive, OTSS switching technology is a promising solution for mice flows, thanks to its fine-grained connections multiplexed in temporal domain. We can evolve current FG networks by reserving a portion of spectrum for OTSS and adding fast optical switches like PLZT, while other parts of channels still apply

conventional FG channel/spectrum-slot routing by channel-selective switch as in Fig. 1b. On the other hand, as an elephant flow requires large bandwidth to transmit data, we can reserve conventional FG switching channels for elephant flows.

For FG channels, we need to solve the routing and channel assignment problem. As shown in Fig. 2, there can be multiple routes for two servers connecting with different rack switches, so we need to consider routing for intra-DC networks. Fixed routing provides only one fixed pre-calculated routing for each pair of nodes and can result in high blocking probability. On the other hand, adaptive routing, which needs to be calculated on each node based on current network state, will require high computation and a longer response time, and is not practical in large-scale datacenter networks. Thus, we apply fixed-alternate routing enabled by K-shortest path algorithm, which considers multiple fixed routes for a connection request and tries to establish any one of the routes. Since the bandwidth requirement of flows can vary significantly, we need to assign channel bandwidth accordingly, e.g., for a request that needs 10 Gbps bandwidth, we must assign at least 2 FG channels if the channel capacity is 6.25 Gbps or 1 FG channel with 12.5 Gbps capacity. For channel assignment, we use first-fit (FF) approach, which tries to pack all of the in-use wavelengths toward the low end of the channel space.

We also need to solve the routing and timeslot assignment problem for OTSS channel. Considering the efficiency of fixed-alternate routing explained above, we apply the same routing strategy for OTSS enabled by K-shortest path algorithm. After deciding the routing, we apply a first-fit (FF) method to choose the timeslots for a new traffic flow request as shown in Fig. 4.

Employing flow-detection methods described in Section II.C, we can design a bandwidth allocation scheme, called flow-aware bandwidth allocation (FABA) scheme, to accommodate mice flows and elephant flows. As shown in Fig. 1a, when a flow is generated by a traffic request at an end-host, a virtual layer in operating system will use sliding window [17] to collect the first n packets of the flow and then apply machine learning algorithm (i.e., C4.5 decision tree or NBD) to detect whether this is an elephant flow. If this virtual layer detects an elephant flow, it will mark this flow and send an in-band signal to the centralized controller shown in Fig. 1a. The centralized controller then obtains the knowledge of which flow is elephant flow and shares this information with all rack switches. So each rack switch can transmit elephant flows on FG channels by using fixed-alternate routing and first-fit algorithm (iterate at most K times) to calculate valid path and assign available channel. On the other hand, if a flow is not marked as an elephant flow, the rack switch will assume it as a mice flow and assign

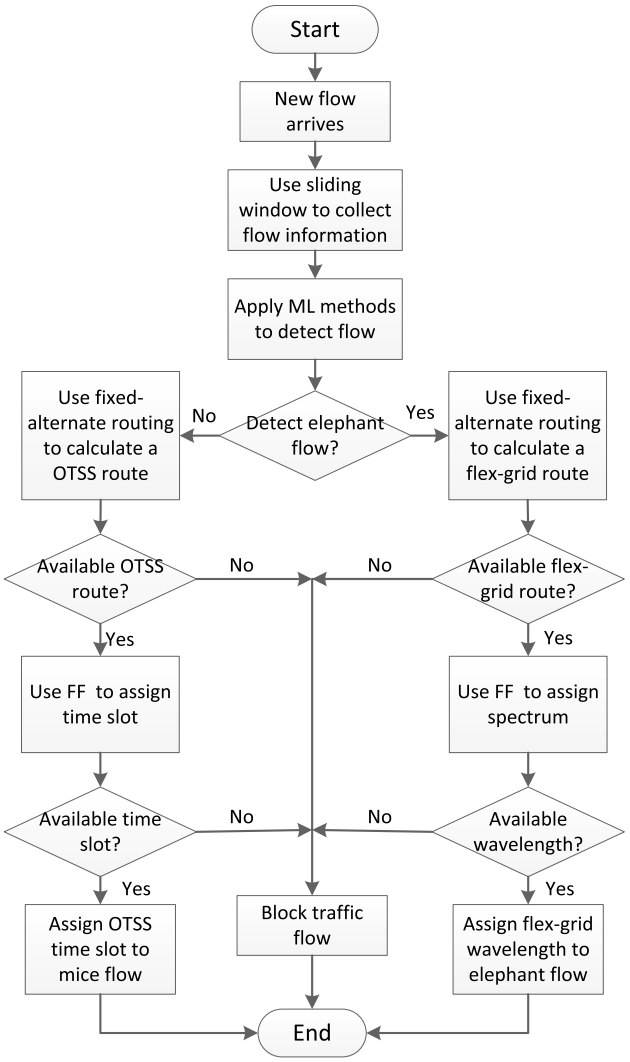

Fig. 4 Flowchart of FABA scheme

OTSS timeslots to provide fine-grained connection by using fixed-alternate routing and first-fit algorithm to find valid route and timeslots.

B. Dynamic Flow-Aware Bandwidth Allocation Scheme

In FABA, we transfer elephant flows only on FG channels and transfer all mice flows on OTSS channels. We apply OTSS switching on m out of n channels based on the knowledge of datacenter traffic characteristics. For example, if only 10% data are carried in mice flows, we can apply OTSS switching on 10% bandwidth resources. But traffic flows keep changing all the time. Sometimes, elephant flows might be majority flows with only few of mice flows. In this case, all FG channels could be very busy with transferring data and thus block new elephant flows, while OTSS channels might be quite empty leading

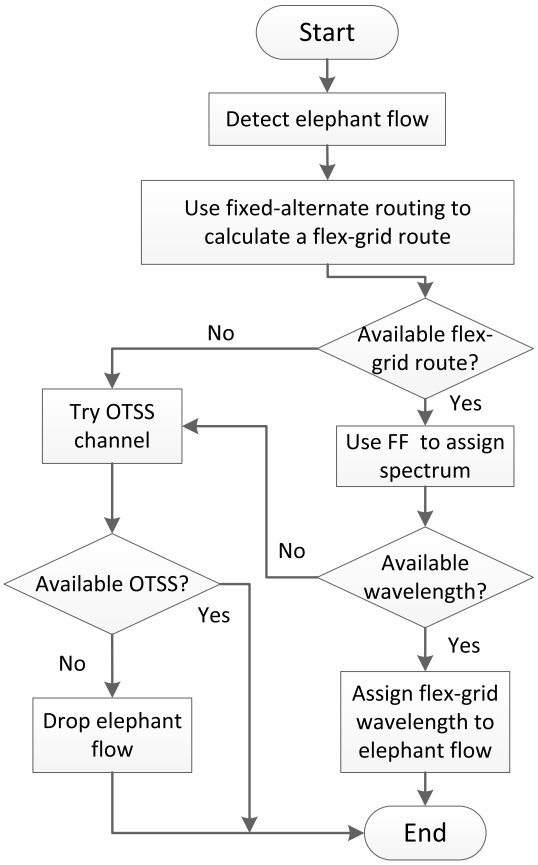

Fig. 5 Flowchart of D-FABA scheme

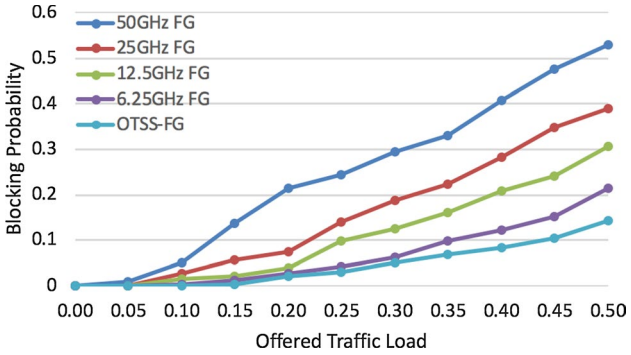

Fig. 6 Blocking probability for OTSS-FG and FG networks

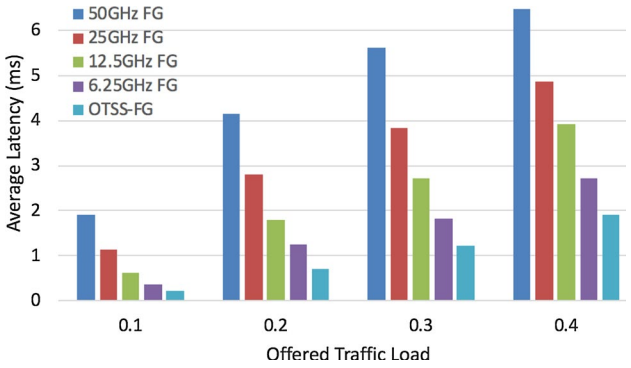

Fig. 7 Average latency for OTSS-FG and FG networks

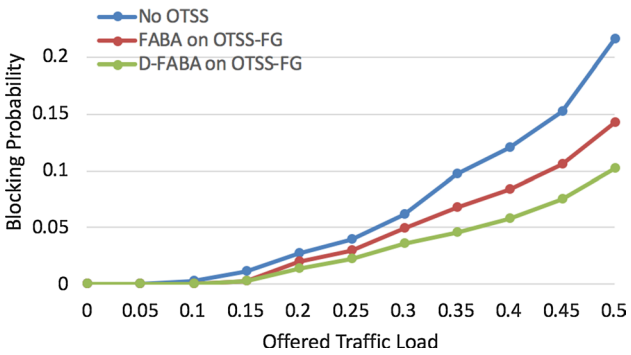

Fig. 8 Blocking probability for D-FABA, FABA and FG

to a waste of bandwidth resources. Thus, we want to design a dynamic version of FABA (called D-FABA) to adaptively adjust bandwidth allocation scheduling based on current traffic flow status. When there are no available FG channels for an elephant flow, we will search whether there is any available OTSS channel. This way, if OTSS channel is not busy, it can help with FG channel to transfer some elephant flows so as to decrease the blocking probability of elephant flows and improve total network performance.

D-FABA is shown in Fig. 5. Similar to FABA, D-FABA also uses sliding window to collect flow information and detect elephant flows at OS layer in end server. If a flow is considered as a mice flow, we apply the same OTSS routing and timeslot allocation algorithm to assign OTSS channel for this flow. If a flow is detected as an elephant flow, we first try to find FG route for this flow. If FG route exists, we search for available FG spectrums for this flow. Otherwise, if there is neither available FG route nor FG spectrum for this flow, we will ask for help from OTSS channels by searching for available OTSS route and OTSS timeslot.

4 Numerical results

We conduct simulations for dynamic traffic scenario, where flows are generated in real time under 3-layer flat-tree topology similar to Fig. 2. The first layer consists of 5 core switches, and each core switch connects with 10 rack switches. Each rack switch contains 8 servers (i.e., total 400 servers). We assume that propagation delay is negligible in our simulation. Spectral efficiency is 1 b/s/

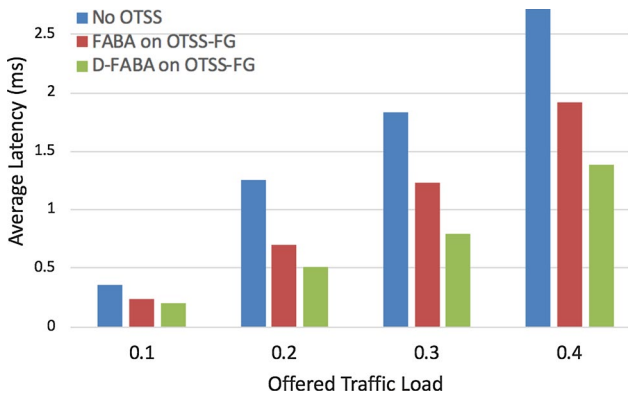

Fig. 9 Average latency for D-FABA, FABA and FG

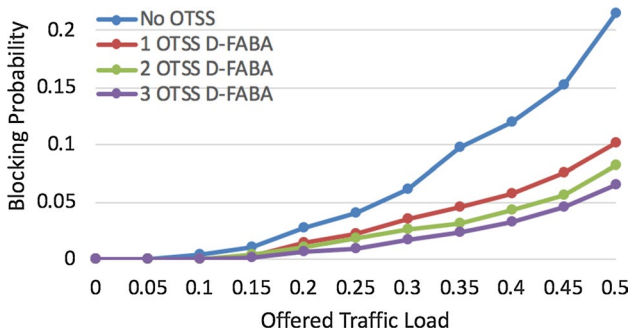

Fig. 10 Blocking probability for D-FABA with different number of OTSS channels

Fig. 11 Average latency for D-FABA with different number of OTSS channels

Hz. On OTSS, we set OTSS frame to be 1 *ms*, and the smallest time slice to be 10 μs. We iterate 50 independent instances so that all plotted values fall within a 5% confidence interval with 95% confidence level in Figs. 6, 7, 8, 9, 10 and 11. Traffic flows in DC, as opposed to voice traffic, display a high burstiness and extreme variability over a wide range of timescales [19, 20]. This burstiness feature induced by self-similar traffic can be expressed as a superposition of multiple independent and identically distributed (i.i.d.) ON/OFF sources, with a heavy-tailed distribution of active/inactive phases. Thus, we apply Pareto distribution to model traffic flows in DC [20]. Specifically, periods of flows generation are modeled by matching ON/OFF periods, corresponding to data generation/nongeneration instances, which simulates traffic behavior found in real datacenters. The lengths of these ON/OFF periods are characterized by heavy-tailed random Pareto distribution. Simulations are conducted with different normalized offered traffic loads, defined as the ratio of data generated by all servers to the total capacity of OTSS-FG architecture. For a certain offered traffic load, we adjust different Pareto parameters for mice flows and elephant flows so that, on average, 90% generated flows are mice flows which only occupy 10% of total data size, while 10% flows are elephant flows contributing 90% data size.

Based on the traffic flow classification results obtained in [16], we obtain 95% accuracy to correctly detect elephant flows in following dynamic simulations. In other words, for each generated elephant flow, it has 5% probability to be wrongly detected as a mice flow. Considering the quality of DC network, if a traffic flow waits too long to be assigned resources, it will lead to congestion for later-coming flows and reduce the total network performance. Thus, we should set a reasonable upper latency limit to drop flows properly, which is set to 10 *ms* according to [21]. In other words, traffic flows failing to start transmission within 10 *ms* will be dropped. Simulation experiments are conducted on an 8-core × 86-64bit Intel(R) Core (TM) i7-4770 CPU @ 3.40 GHz.

Figure 6 shows the bandwidth blocking probability of an OTSS-enabled network compared to conventional FG network with different slot granularity. In OTSS-FG architecture, 25 GHz is reserved for OTSS switching to accommodate reliability-related traffic, while 225 GHz is reserved for 6.25 GHz FG switching. For fairness, we also reserve 250 GHz for conventional FG networks as benchmark to compare its performance with OTSS-FG. On a conventional FG optical network, four grid sizes are considered (50 GHz, 25 GHz, 12.5 GHz, 6.25 GHz). At light traffic load (offered traffic load < 0.3), OTSS-FG performs similar to FG, because a small number of traffic flows are unlikely to be blocked by others. But with increasing offered traffic load, we find that OTSS can help endure heavier traffic burden than conventional FG networks. This is because OTSS can provide much finer grained slot to efficiently transmit mice flows so as to decrease the possibility of blocking elephant flows.

Figure 7 depicts the relationship between the average latency and offered traffic load. We can see that OTSS-FG network could always achieve lower average latency compared with conventional FG networks. This is because OTSS

channel can provide much finer grid to transmit mice flows more efficiently so that fewer mice flows need to wait and elephant flows do not need to be delayed by mice flows. This confirms the observation in Fig. 6.

In Fig. 8, we study the bandwidth blocking probability of D-FABA and FABA in OTSS-FG architecture network (same as described in Fig. 6) compared to No-OTSS network (i.e., conventional FG with 6.25 GHz). We find that, at light offered traffic load (< 0.2 Erlang), D-FABA performs almost the same as FABA, because there is enough bandwidth for elephant flows and few elephant flows need to be transmitted on OTSS channel. But, at high traffic load, more elephant flows compete for FG channels. As D-FABA could help transmit some elephant flows on OTSS channels, these elephant flows will not be blocked (which may have been blocked in FABA scheme).

Figure 9 shows how D-FABA can improve average latency performance on OTSS-FG network compared with FABA and No-OTSS. We can see that both FABA and D-FABA perform better than No-OTSS all the time. At high offered traffic load, more elephant flows arrive requiring bandwidth, as D-FABA can transfer some of the competing elephant flows onto OTSS channels, so D-FABA achieves smaller average latency than FABA. This also confirms our observation in Fig. 8.

In Fig. 10, we compare the blocking probability of No-OTSS (i.e., conventional FG with 6.25 GHz) and OTSS-FG networks with different number of OTSS channels. We apply D-FABA scheme on OTSS channel(s) and each OTSS channel is 25 GHz. We see that three OTSS channels always obtain better performance than one and two OTSS channels in OTSS-FG networks. This is because D-FABA can use OTSS channels to help transmit elephant flows. Thanks to finer grids provided by OTSS, the more numbers of OTSS channels, the better performance OTSS-FG can achieve. But we further notice that, at light traffic load (< 0.3 Erlang), three OTSS channels perform similar as one and two OTSS channels. This is because, in our simulation, although mice flows occupy 90% of total number of flows, they only contribute to 10% of the entire data volume in the network. In other words, mice flow only needs 10% of total channel bandwidth to transmit which equals to 25 GHz out of 250 GHz. Increasing the number of OTSS channels may not provide significant performance improvement. Thus, considering the expensive cost of fast optical switches, in reality, we need to trade-off between the number of OTSS channels and network performance. For example, in our simulation, reserving 25 GHz for OTSS and 225 GHz for FG, which align to mice and elephant traffic flow ratio, might be the most efficient solution in terms of cost and network performance.

Figure 11 shows how average latency performs as traffic load increases for No-OTSS networks and D-FABA scheme-applied OTSS-FG networks with different number of OTSS channels. We also see that, although 3 OTSS channels perform best in terms of latency, it does not achieve much better result than 1 or 2 OTSS channels, especially for light traffic load, which confirms our analysis in Fig. 10.

5 Conclusion

In this study, we investigated an OTSS-enabled flex-grid (OTSS-FG) architecture for intra-datacenter networks. We proposed a flow-aware bandwidth allocation scheme and exploited machine-learning techniques to detect mice flow and elephant flow. We also developed a Mixed Integer Linear Program to mathematically model the optimal bandwidth allocation scheme in OTSS-FG architecture. OTSS-FG flow-aware architecture can achieve higher throughput than FG architecture and OTSS-FG flow-unaware architecture. Numerical simulations show that the proposed flow-aware bandwidth allocation scheme can outperform a benchmark scheme in terms of average delay and block probability.

Acknowledgement This work was supported in part by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) [B0117-16-1008, Development of datacenter Optical Networking Core Technologies for Photonic Frame based Packet Switching]. It was also supported in part by National Science Foundation Grant No. 1716945.

References

1. Index, C.G.C.: Forecast and methodology, 2016–2021. Cisco Systems Inc, San Jose, CA, USA (2016)
2. Benson, T., et al.: Understanding data center traffic characteristics. Comput. Commun. Rev. **40**(1), 92–99 (2010)
3. Xia, W. et al.: A survey on data center networking (DCN): Infrastructure and operations. IEEE Commun. Surv. Tutor. **19**(1), 640–656 (2017)
4. Pontes, A., et al.: Data center networks (DCN) are facing significant increase. Opt. Switch. Netw. **19**, 10–21 (2016)
5. Farrington, N., et al.: Helios: a hybrid electrical/optical switch architecture for modular data centers. ACM SIGCOMM Comput. Commun. Rev. **41**(4), 339–350 (2014)
6. Chen, K., et al.: OSA: An optical switching architecture for data center networks with unprecedented flexibility. IEEE/ACM Trans. Netw. **22**(2), 498–511 (2014)
7. Mestre, M.A., et al.: Optical slot switching-based datacenters with elastic burst-mode coherent transponders. The European Conference on Optical Communication (ECOC). IEEE (2014)
8. Fiorani, M., et al.: Hybrid optical switching for energy-efficiency and QoS differentiation in core networks. IEEE/OSA J. Opt. Commun. Netw. **5**(5), 484–497 (2013)
9. Yu, X., et al.: Migration from fixed grid to flexible grid in optical networks. IEEE Commun. Mag. **53**, 34–43 (2015)
10. Wosinska, L.: Optical network architectures for datacenters. In: Photonic Networks and Devices, pp. NeW2B.1 (2017). https://doi.org/10.1364/NETWORKS.2017.NeW2B.1

11. Zhong, Z., et al.: Evolving optical networks for latency-sensitive smart-grid communications via optical time slice switching (OTSS) technologies. Opto-Electronics and Communications Conference (OECC) and Photonics Global Conference (PGC), pp. 1–3. IEEE (2017)
12. Hua, N., et al.: Enabling low latency at large-scale data center and high-performance computing interconnect networks using fine-grained all-optical switching technology. Optical Network Design and Modeling (ONDM), pp. 1–4. IEEE (2017)
13. Muqaddas, A.S., et al.: Exploiting time-synchronized operations in software-defined elastic optical networks. Optical Fiber Communication Conference (2017)
14. Dong, G., et al.: Fast and ultra-compact multi-channel all-optical switches based on silicon photonic crystal nanobeam cavities. Conference on Lasers and Electro-Optics (CLEO) (2018)
15. Pattavina, A., et al.: Performance evaluation of time driven switching for flexible bandwidth provisioning in WDM networks. IEEE GLOBECOM (2004)
16. Wang, L., et al.: Scheduling with machine-learning-based flow detection for packet-switched optical data center networks. J of Opt. Commun. Netw. **10**(4), 365–375 (2017)
17. Curtis, A. et al.: Mahout: Low-overhead datacenter traffic management using end-host-based elephant detection. Proceedings IEEE INFOCOM (2011)
18. Chatterjee, B., et al.: Routing and spectrum allocation in elastic optical networks: A tutorial. IEEE Commun. Surv. Tutor. **17**, 1776–1800 (2015)
19. Arrowsmith, D.K., et al.: Datatraffic, topology and congestion. In: Complex Dynamics in Communication Networks, pp. 127–157. Springer (2005)
20. Caida (The Cooperative Association for Internet Data Analysis). Packet size distribution comparison between Internet links in 1998 and 2008. http://www.caida.org/research/traffic-analysis/
21. Institute of Electrical and Electronics Engineers, IEEE Standard Communication Delivery Time Performance Requirements for Electric Power Substation Automation. IEEE Standard 1646–2004 (2005)

Lin Wang received the B.Eng. degree in telecommunication engineering from Beijing University of Posts and Telecommunication, Beijing, China, in 2013, and the Ph.D. in computer science from University of California, Davis, Davis, CA, USA, in 2018. Her research interests include next-generation EPON and optical datacenter networks.

Xinbo Wang received the B.Eng. degree in telecommunication engineering from Beijing University of Posts and Telecommunication, Beijing, China, in 2013 and the Ph.D. in computer science from University of California, Davis, Davis, CA, USA, in 2017. He is currently working as a research scientist at Facebook. Inc. His research interests include cloud radio access network and optical transport network for 5G.

Massimo Tornatore received the Ph.D. degree in Information Engineering Department of Electronics, Information and Bioengineering, Politecnico di Milano, Italy, in 2006, where he is currently an Associate Professor. He also holds an appointment as an Adjunct Professor with the Department of Computer Science, University of California, Davis. He has authored over 280 peer-reviewed conference and journal papers. His research interests include performance evaluation, optimization and design of communication networks (with an emphasis on the application of optical networking technologies), cloud computing and energy-efficient networking. He was a co-recipient of eleven best-paper awards. He is a member of the editorial board of the journal Photonic Network Communications (Springer), Optical Switching and Networking (Elsevier) and the IEEE Communication Surveys and Tutorials.

Kwangjoon Kim was born in Seoul, Korea, in 1958. He received the B.S. and M.S. degrees in physics from Seoul National University, Seoul, Korea, and the Ph. D. degree in physics from the Ohio State University, Columbus, Ohio, USA, in 1981, 1983 and 1993, respectively. He joined ETRI in 1984 and worked on HF communications, until he enrolled in the Ph. D. program in the Ohio State University, where he worked on various linear and nonlinear optical behaviors of conducting polymers. He rejoined ETRI and worked on optical semiconductor devices with quantum wells. His current research interests focus on the WDM optical communication systems and high-speed optical transmission, including all-optical switching network.

Biswanath Mukherjee (S'82–M'84–SM'05–F'07) received the B.Tech. degree from the Indian Institute of Technology, Kharagpur, India, in 1980, and the Ph.D. degree from the University of Washington, Seattle, WA, USA, in 1987. He is a Distinguished Professor and Founding Director of the Institute for Broadband Research and Innovation (IBRI) at Soochow University, P. R. China, and a Distinguished Professor Emeritus at the University of California, Davis, CA, USA, where he was the Chairman of Computer Science during 1997–2000. He was the General Co-Chair of the IEEE/OSA Optical Fiber Communications (OFC) Conference 2011, Technical Program Co-Chair of OFC'2009 and Technical Program Chair of the IEEE INFOCOM'1996 conference. He is Co-Editor of Springer's Optical Networks Book Series. He has served on eight journal editorial boards, most notably IEEE/ACM TRANSACTIONS ON NETWORKING and IEEE NETWORK. In addition, he has the Guest Edited Special Issues of PROCEEDINGS OF THE IEEE, IEEE/OSA JOURNAL OF LIGHTWAVE TECHNOLOGY, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and IEEE COMMUNICATIONS. He has supervised 78 Ph.D.s to completion. He is Winner of the 2004 Distinguished Graduate Mentoring Award, 2009 College of Engineering Outstanding Senior Faculty Award, and 2016 International Community Building Award, and the 2019 Faculty Distinguished Research Award at UC Davis. He is the Co-winner of 15 Best Paper Awards, including the 2018 Charles Kao Best Paper Award for IEEE/OSA Journal on Optical Communications and Networks; five from IEEE Globecom Symposia, four from IEEE ANTS and two from National Computer

Security Conference. He is the author of the graduate-level textbook Optical WDM Networks (New York, USA: Springer, Jan. 2006). He served a five-year term on the Board of Directors of IPLocks, a Silicon Valley startup company (acquired by Fortinet). He has served on the Technical Advisory Board of several startup companies, including Teknovus (acquired by Broadcom). He is the Winner of the IEEE Communications Society's inaugural (2015) ONTC Outstanding Technical Achievement Award "for pioneering work on shaping the optical networking area."