Machine-Learning-Based Prediction for Resource (Re)allocation in Optical Data Center Networks

Sandeep Kumar Singh and Admela Jukan

***Abstract*—Traffic prediction and utilization of past information are essential requirements for intelligent and efficient management of resources, especially in optical data center networks (ODCNs), which serve diverse applications. In this paper, we consider the problem of traffic aggregation in ODCNs by leveraging the predictable or exact knowledge of application-specific information and requirements, such as holding time, bandwidth, traffic history, and latency. As ODCNs serve diverse flows (e.g., long/elephant and short/mice), we utilize machine learning (ML) for prediction of time-varying traffic and connection blocking in ODCNs. Furthermore, with the predicted mean service time, passed time is utilized to estimate the mean residual life (MRL) of an active flow (connection). The MRL information is used for dynamic traffic aggregation while allocating resources to a new connection request. Additionally, blocking rate is predicted for a future time interval based on the predicted traffic and past blocking information, which is used to trigger a spectrum reallocation process (also called defragmentation) to reduce spectrum fragmentation resulting from the dynamic connection setup and tearing-down scenarios. Simulation results show that ML-based prediction and initial setup times (history) of traffic flows can be used to further improve connection blocking and resource utilization in space-division multiplexed ODCNs.**

***Index Terms*—Defragmentation; Holding time; Machine learning; Optical data center networks; Prediction; Space-division multiplexing.**

I. Introduction

The growth of traffic in cloud data centers (DCs), which offer application-based centralized services (e.g., online services, storage, backups, platform, infrastructure-as-a-service, and data replication), require more and more network capacity, as the projected traffic from cloud to users/devices will likely rise by 3.3-fold (15.3 zettabytes per year) by 2020 [1]. At the same time, optical data center networks (ODCNs), i.e., intra- and inter-ODCNs, are emerging as a viable solution to cope with the increase in data center network (DCN) traffic and also to mitigate the power consumption of its counterpart, i.e., electrical DCNs. Moreover, DC traffic can be categorized into short-, medium-, and long-lived based on the flow completion time, which ranges from a few milliseconds to hundreds of seconds. A majority (more than 80%) of the DC traffic flows are mice or short-lived flows that are mostly latency-sensitive; however, more than 80% of the entire traffic volume is carried by long-lived or elephant flows [2,3], which require high throughput; they are also mostly latency insensitive and deadline agnostic (e.g., virtual machine migration) (see Table I). In another study, limited to Facebook's services [4], the authors found that many flows are long-lived but not heavy, and traffic remains stable over even subsecond intervals. On the other hand, inter-DC traffic can be classified into two categories [5]: *background* DC-to-DC (e.g., data replication) or *client-triggered* (e.g., video streaming), which exhibit varying trends over the day but could be predicted using learning methods [6]. Thus, given the DCN traffic forecast and acknowledging its periodicity, there is a need for applying learning algorithms for better resource utilization both in ODCNs as well as in servers.

Artificial intelligence (AI), although in its infancy, is emerging as a technology to realize "smart" optical networking, where AI can help in network control and management functions like flexible resource allocation, fault detection, failure prediction, and performance monitoring [7]. Among many AI-based technologies, machine learning (ML) is the most preferable technique for learning, classification/identification, and prediction. Many variants of brain-inspired neural networks, such as deep/recurrent/convolutional neural networks, are often used as ML tools that learn from input sets and from the past in order to classify/identify test objects and predict the future [7]. As modern applications, especially cloud-based, are heterogeneous and diverse in nature, holding (service) time information (HTI) of applications is not known in advance, and are heavy-tailed distributed [2,3]. Furthermore, intra- and inter-data center traffic follows a repetitive pattern during a week or a month. Thus, ML-based techniques can be handy in predicting or forecasting future traffic in terms of mean arrival rate and mean HTI of traffic flows, and other network-related issues such as connection blocking. Although the prediction of mean HTI is useful in admission control, it is not sufficient. The reason is that, when a new service request arrives, it could be established with other flows having similar remaining (residual) lives such that the termination of these neighboring flows leads to

Manuscript received January 2, 2018; revised February 26, 2018; accepted April 7, 2018; published May 14, 2018 (Doc. ID 318574).

The authors are with the Department of Electrical and Computer Engineering, Technische Universität Carolo-Wilhelmina zu Braunschweig, Braunschweig 38106, Germany (e-mail: sandeep.singh@tu-bs.de).

https://doi.org/10.1364/JOCN.10.000D12

reduced spectrum fragmentation [8]. At the same time, the residual life of an active (heavy-tailed distributed) flow or connection is a function of already spent time (i.e., active lifetime) and its mean HTI. Thus, in absence of known mean HTI, its prediction becomes necessary to utilize the mean residual life (MRL) information. Furthermore, DC traffic exhibits the heavy-tailed characteristic, i.e., holding times of some flows are quite large, which make the most of the existing works [9–13] inapplicable for routing and spectrum allocation (RSA) of time-varying traffic in ODCNs because they assume an exponentially distributed HTI model. Due to this reason, to the best of our knowledge, none have utilized MRLs of active connections in RSA in ODCNs.

To address the above issue, in this paper we use an ML-based prediction model to predict time-varying mean HTI of three classes of services, distinguished by their bandwidth demands and mean service times (HTI), and use them to estimate MRLs of active flows whose holding times are assumed to be unknown and lognormally (heavy-tailed) distributed. However, the question is how to best use the predicted and estimated traffic-related information in (re)allocation of resources (transponders, fiber-core, and spectrum) in ODCNs. To answer this question, we propose a novel ML-based resource allocation and reallocation scheme in ODCNs that use the predicted mean HTI, MRLs of active flows, and known HTI of departed connections[1] to allocate resources to a new connection request. More precisely, we propose cost metrics that utilize the predicted and estimated traffic information as well as requirements of a new connection request to either provision it over an existing lightpath by means of traffic aggregation or route it over a new lightpath. Additionally, the knowledge of past and predicted blocking information is used to trigger an on-demand defragmentation, which is essential in reducing spectrum fragmentation in multicore fiber links used as interconnects in ODCNs.

The remainder of the paper is as follows. We describe the related work and our contribution in Section II. Section III describes how to estimate MRLs of connections and explains whether to route a new request along a lightpath or to create a new lightpath based on the cost functions. Section IV explains an ML model for traffic prediction. The ML-based connection admission and an on-demand defragmentation are discussed in Section V. We evaluate the performance of the proposed schemes in Section VI. Finally, we conclude the paper in Section VII.

II. Related Work and Our Contribution

The data center workloads (see Table I) are diverse in nature: they are short- and long-lived, latency sensitive or insensitive, and deadline aware or unaware, which, in turn, require dynamic resource allocation. Inefficient resource (e.g., spectrum) handling often leads to higher resource consumption and higher blocking. Therefore, aggregation of mice flows and splitting of elephant flows over multiple spectrum paths could help better utilize the network bandwidth. As high-speed switching technologies mature, top-of-rack (ToR) switches can have either electrical or optical (NIC) interfaces (1 GbE, 10 GbE, 40 GbE, and 100 GbE), which represent a technological limit on application-specific bandwidth demands. Therefore, in optical DCNs, aggregation of traffic flows will lead to better bandwidth utilization and lower service blocking probability. While certain types of data center applications are characterized by known holding times, for others it can be estimated or predicted. Therefore, several dynamic scheduling algorithms [14–16] devise new resource models for EON as well as WDM-based DCN architectures, under the assumption that holding times are known *a priori*. In [15], flex-grid optical networks have been used to support inter-data center communication for bitrate-based applications. The holding time is assumed to be known *a priori* for each request; thus, a holding-time aware scheduling strategy is developed for both immediate and advanced reservation schemes (i.e., IR and AR). Similarly, the HTI-based strategy has been included in addition to the frequency dimension to optimally utilize the flexible bandwidth in EONs for both data volume- and bitrate-based applications [16].

The time-varying traffic model for RSA and bandwidth defragmentation has also been considered in EONs [9–12]. At the same time, it is important to note that time-varying traffic problems have been addressed in two forms in the EON literature: time-varying bandwidth demands, and time-varying mean arrival rate and mean holding time. While [9] and [12] propose integer linear programming (ILP) and heuristic solutions to deal with the RSA problem considering time-dependent bandwidth demands, [10,11] proposed heuristic solutions for (re)configuration of lightpaths in a time-varying traffic scenario. However, these works do not employ the predictive or exact knowledge of time-varying mean arrival rate or HTI. Recently, [17] used a neural network to predict arrival time and holding time of future connections for a given traffic load and showed that the predictive knowledge of traffic could be used for designing an efficient RSA scheme. However, for each given load, a fixed (100 K) number of requests were generated to evaluate blocking probability, which do not truly characterize the time-varying traffic because they ignore the correlation of spectrum occupancy in consecutive time intervals for time-varying loads. Furthermore, the prediction of individual connection-specific information could be inaccurate and computationally complex in a time-varying traffic condition. Thus, in this paper we use machine learning to mainly predict time-varying mean holding time ($\overline{h}_k(t)$) of traffic classes k, $k = \{1, 2, 3\}$, and utilize this information in designing an efficient resource (re)allocation scheme.

Other studies not primary related to DC applications are usually based on the classical dynamic traffic model, where requests are characterized by random arrivals and holding times, both described by their probability density function. Tornatore *et al.* [18] used an HTI-based strategy to establish WDM lightpaths and dynamically assign new requests onto a lightpath, which has the longest remaining life.

[1]In this paper, traffic flow and connection are used interchangeably.

TABLE I
DIVERSITY OF INTRA- AND INTER-DATA CENTER APPLICATIONS[a]

	Application Examples	Short-Lived (ms)	Long-Lived (s)	Latency-Sensitive	Latency-Insensitive	Deadline-Aware	Deadline-Unaware
Intra-DC	Web search, IM, SR, haptics	✓		✓		✓	
	Data mining, storage, analytics		✓		✓		✓
	Email, web services, robotics	✓		✓		✓	✓
	Backups, downloads, test-beds		✓		✓		✓
	YouTube, games, VR, smart city	✓	✓	✓	✓	✓	✓
Inter-DC	VM migration, data replication, LB		✓		✓		✓
	Search, file sharing, video streaming	✓	✓	✓	✓	✓	✓

[a]IM, instant messaging; SR, sensor readings; VR, virtual reality; VM, virtual machine; LB, load balancing.

If the precise value of the holding time is unknown, the authors estimate the remaining life using the holding time distribution. Due to complexity, the authors' approach was limited to an exponential holding time distribution. As a consequence, the traffic history, which is given by the initial setup times of active applications, is of no importance due to the memoryless property of the exponential distribution. But, as mentioned above, the DC traffic is of high burstiness; thus, the exponential assumption is inappropriate. The same strategy was used by Zong *et al.* [13] for load balancing in ODCNs. However, as most DC traffic is generally heavy-tailed distributed, e.g., lognormal, utilization of its history has not been taken into account.

A. Our Contribution

This paper deals with the problem of resource allocation and reallocation of time-varying traffic in ODCNs. We propose an ML-based resource (re)allocation scheme, taking into account the past and future of network load characteristics. In this study, we adopt an SDN-based intra-ODCN architecture [14] and inter-ODCN architecture [13,19], whose ToR switches and DCs, respectively, are interconnected by multicore fiber networks, and support the flexibility and constraints (e.g., capacity constraint and spectrum continuity and contiguity constraint) of space-division multiplexing-based elastic optical networks (SDM-EONs). Our contributions are multifold, as follows:

- We consider a multiclass dynamic service model and use an ML-based neural network model to predict the time-varying mean HTI of three classes of services whose holding times are unknown and lognormally distributed.
- We propose cost metrics for connection admission that incorporate the predicted mean HTI, estimated MRLs of active flows, and mean HTI of departed connections in order to assign a minimum cost to a path that minimizes the spectrum fragmentation in spectral and time domains.
- We show that only HTI prediction is not enough in ODCNs; the estimation of MRLs of active connections using their history (passed times) and prediction of connection blocking for future time intervals are also important factors in resource (re)allocation, especially in ODCNs whose traffic is mostly heavy-tailed distributed.
- We show that the latency-insensitive connection requests could be delayed to reduce overall connection blocking.
- More importantly, we show that, when a prediction goes unexpectedly badly, the resources could be inefficiently allocated, leading to higher connection blocking.

III. ESTIMATION OF MRL AND COST FUNCTIONS

In this section, we first show how to use an HTI distribution function for estimation of mean lifetime of traffic flows, then illustrate its uses in traffic aggregation and provisioning of new lightpaths in ODCNs. In general, there are two kinds of application requests: one with *known* HTI and other with *unknown* HTI. Let us assume that a connection request j with *known* HTI advertises its value h_j at the start of service t_{S_j}, as shown in Fig. 1. For example, a fixed-size data volume has to be transmitted at a given interface speed, or a video streamed at fixed bandwidth is active for a predefined time. Thus, at any time t during the service of the request j, we know its remaining life $r_j = h_j - (t - t_{S_j}) = h_j - d_j$, where we define d_j as its history. The other types of applications do not advertise their HTI (*unknown* HTI). In this case, we assume a random value of the holding time, h'_j, but drawn from a known probability distribution $F_H\{t\} = P\{H \le t\}$, which is used to derive its mean value $E\{H\}$ and the distribution of its residual life $r_j = h'_j - d_j$, which is given by $F_{\mathcal{R}}(t, d_j) = P\{H \le d_j + t | H > d_j\}$, where a continuous and positive random variable $\mathcal{R}$ represents the residual life. Note that the residual life distribution depends on the connection's history d_j. The mean residual life (MRL) of a connection j with history d_j is defined as $\bar{r}_j(d_j) = \int_0^\infty (1 - F_{\mathcal{R}}(t, d_j))dt$, or without calculating the residual life distribution, the MRL value $\bar{r}_j(d_j)$ is given by $E[H - d_j | H > d_j] = [\int_{d_j}^\infty (F_H^c(t))dt]/F_H^c(d_j)$,

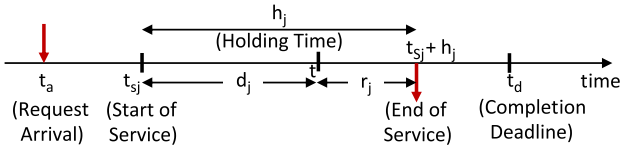

Fig. 1. Various traffic parameters of an application request.

where $F^c_H(t) = 1 - F_H(t)$ is the so-called survival function [20].

A. MRL of Lognormally Distributed Holding Time

As requests with known holding times have been well investigated, in this paper our focus is on DC applications whose lifetimes are mostly unknown but derived from a heavy-tailed distribution, e.g., a lognormal [2,3] function with parameters μ and σ. Lognormal cumulative distribution function (cdf) is given[2] by $F_H(t) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{\ln t - \mu}{\sigma\sqrt{2}}\right)\right], t > 0, \sigma > 0$, and the corresponding probability distribution function (pdf) is $f_H(t) = \exp\left[-\left(\frac{\ln t - \mu}{\sigma\sqrt{2}}\right)^2\right]/t\sqrt{(2\pi\sigma^2)}$. The mean lifetime is $E\{H\} = e^{\mu+\sigma^2/2}$, and the coefficient of variation (CoV) is $c_v = \sqrt{e^{\sigma^2} - 1}$. In general, for a heavy-tailed distribution function such as lognormal, which does not have a closed-form residual distribution, calculating MRL requires double integrals to be solved by some approximations (e.g., trapezoidal rule), which are mostly time-consuming and error-prone using computer simulation. Thus, as suggested in [20,21], the MRL value for any distribution, including lognormal, satisfying $t \times F^c_H(t) \to 0$ as $t \to \infty$ can alternatively be calculated as

$$\bar{r}_j(d_j) = \frac{1}{F^c_H(d_j)} \int_{d_j}^{\infty} t f_H(t) \mathrm{d}t - d_j, \tag{1}$$

where $f_H(t)$ is the pdf of holding time H. The value of $\int_{d_j}^{\infty} t f_H(t)\mathrm{d}t$ for lognormally distributed variable H is given as below, after substituting $y = \frac{\ln t - \mu}{\sigma\sqrt{2}}$ and $y(d_j) = y(t = d_j)$:

$$\begin{aligned}\int_{\mathrm{d_j}}^{\infty} t f_H(t)\mathrm{d}t &= \frac{1}{\sqrt{\pi}} \int_{y(\mathrm{d_j})}^{\infty} \exp\left[-y^2 + y\sigma\sqrt{2} + \mu\right]\mathrm{d}y, \\ &= \frac{\exp[\mu + \sigma^2]}{2} \times \frac{2}{\sqrt{\pi}} \int_{y(\mathrm{d_j})}^{\infty} \exp\left[-\left(y - \frac{\sigma}{\sqrt{2}}\right)^2\right]\mathrm{d}y, \\ &= \frac{E[H]}{2} \times \left[1 - \text{erf}\left(\frac{\ln d_j - (\mu + \sigma^2)}{\sigma\sqrt{2}}\right)\right].\end{aligned}$$

Finally, using the value of above term in Eq. (1), the MRL of a lognormally distributed holding time of a request j with history (passed time) d_j is given by Eq. (2):

$$\bar{r}_j(d_j) = E[H] \times \frac{1 - \text{erf}\left(\frac{\ln d_j - (\mu+\sigma^2)}{\sigma\sqrt{2}}\right)}{1 - \text{erf}\left(\frac{\ln d_j - \mu}{\sigma\sqrt{2}}\right)} - d_j. \tag{2}$$

The dependence of MRL $\bar{r}_j(d_j)$ on mean holding time $E[H]$, history d_j, and CoV c_v is illustrated in Fig. 2, where the numerical (theoretical) value of Eq. (2) is verified with MATLAB simulation. As can be seen, the relationship between the MRL and the passed time (history) is nonlinear and depends on the mean $E[H]$ and CoV c_v (or equivalently μ and σ). The heavy-tailed nature of lognormal distribution can also be seen even for smaller $E[H] = \{1,10\}$ because MRL does not die with increasing passed time. It should be noted that, when mean holding time is small $0 < E[H] < 1$, computing Eq. (2) could produce wrong MRL values for higher passed times. Furthermore, the simulation for the scenario $E[H] = 0.1$ and $c_v = 1$ does not generate lognormally distributed random variables with higher lifetimes; thus, the MRL obtained by the simulation is only shown (in green) for lower values of passed times.

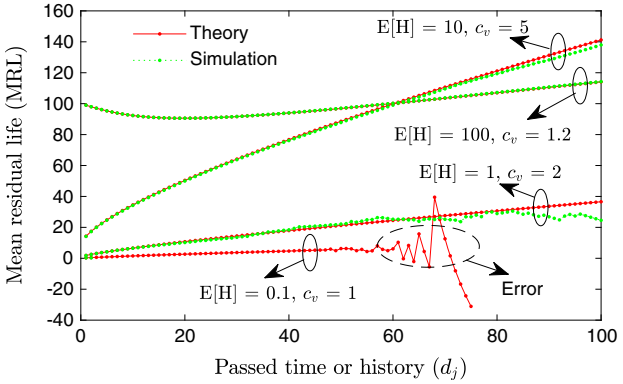

Fig. 2. Comparison of MRLs obtained by Eq. (2) and simulation for various mean holding time $E[H]$ and CoV (c_v) for lognormal distribution.

Now, let us see how HTI and MRL are useful in traffic flow aggregation and provisioning of new lightpaths in ODCNs. Consider an example in Fig. 3, where the spectrum occupancy of a two-core fiber link is shown, when a new traffic flow request R arrives between nodes A and D with demand of b_k gigabytes per second (Gb/s). There are mainly two possibilities to accept the request R: creating a new lightpath over a candidate spectrum path,[3] which satisfies the spectrum continuity, contiguity constraint, and bandwidth requirement of R, and an aggregation of traffic flow R over an existing lightpath between same source-destination pair. Let us evaluate these two scenarios one by one.

B. Provisioning a New Lightpath

Upon arrival of a new request, the control plane tries to provision the best route and resources (transponder, spectrum, fiber cores, etc.), if possible; otherwise, it blocks it when free resources are unavailable and a delayed reservation is not accepted by the connection request. Provisioning of resources is done by the network control and management system based on the requirement of connection requests, and it is often policy-driven. However,

[2]The error function $\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\mathrm{d}t$.

[3]In this paper, we denote a continuous and contiguous free spectrum slice(s) of a route where a connection could be established as a *spectrum path*, and a *lightpath* is setup over a spectrum path to admit a new connection request.

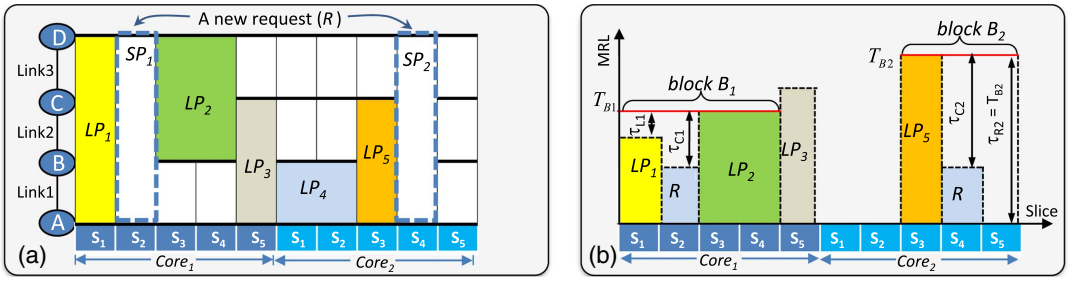

Fig. 3. (a) Spectrum occupancy of a three-link path with two-core fiber and five slices per core. (b) Spectrum occupancy of link 2 along spectral, core, and time. This figure shows the provisioning of a request R along other lightpaths, taking their MRLs into consideration in blocks B_1 and B_2.

irrespective of the policies, network operators like to maximize revenue, which is mainly done by serving more requests and fulfilling their needs. Therefore, we propose a method to create new lightpaths over spectrum paths, which could satisfy the requirements of new requests and also reduce spectrum fragmentation in SDM-EONs such that more requests could be accepted into networks. For this purpose, we utilize a cost function, originally proposed in [8], which takes into account the fragmentation factor in both spectral and time domains and selects the best (minimum cost) spectrum path. To illustrate this, consider an example in Fig. 3(a), where there are three possible spectrum paths (SPs) in a three-link linear network, which could satisfy the bandwidth demand of the new request R, assuming that one spectrum slice is sufficient to satisfy its demand. Notice that only two spectrum paths, SP_1 and SP_2, over slots S_2 and S_4 in fiber cores 1 and 2, respectively, are shown, and a third SP over slot S_5 in core 2 is not shown. Furthermore, to choose the best SP, which could minimize spectrum fragmentation in both spectral and time domains, we assign a cost to each candidate SP using Eq. (3), which takes MRLs of neighboring lightpaths into account:

$$\text{cost}(\text{sp}_i) = \sum_{\forall l \in \text{sp}_i} \phi^l(\text{sp}_i), \quad \text{where}$$

$$\phi^l(\text{sp}_i) = \frac{(\tau_{L_i} + \tau_{C_i} + \tau_{R_i})/T_{B_i}}{|B_i|}. \tag{3}$$

In Eq. (3), the fragmentation factors ($\phi^l(\text{sp}_i)$) of all links traversed by a spectrum path SP_i are taken into account to define the cost of SP_i. The factor $\phi^l(\text{sp}_i)$ is calculated as the average value of normalized differential remaining lives of left (τ_{L_1}), center (i.e., incoming, τ_{C_1}), and right ($\tau_{R_1} = 0$) lightpaths [see Fig. 3(b)]. The normalization constant is the maximum value of MRLs, i.e., $T_{B_i} = \max(\bar{r}_{L_i}, \bar{r}_{C_i}, \bar{r}_{R_i})$ of lightpaths associated with a block B_i. Note that a block B_i in Fig. 3(b) is associated with a group of neighboring lightpaths around the possible spectrum allocation of the request R over a candidate spectrum path SP_i. It is important to mention how the MRL of a lightpath ($\bar{r}_{lp_i}$) is calculated using MRLs of traffic flows (connections) that it serves. Suppose a lightpath (LP_i) is provisioned at time t_0, and at time $t > t_0$ it serves n traffic flows having unknown lifetimes $H_1, H_2, \ldots, H_n$ but known history (passed times) $\vec{d} = \{d_1, d_2, \ldots, d_n\}$. Thus, the MRLs of these flows can be obtained by Eq. (2) as $\bar{r}_1(d_1), \ldots, \bar{r}_n(d_n)$. We define the MRL of a lightpath as the maximum value of MRLs of all connections served by it,[4] i.e., $r_{lp_i} = \max(\bar{r}_1(d_1), \ldots, \bar{r}_n(d_n))$.

To visualize these parameters, let us see Fig. 3(b), where the spectrum occupancy of link-2 is shown across core, spectral, and time dimensions. Here, two possible spectrum allocations of the new request R are also shown together with other lightpaths in block B_1 and B_2. The differential remaining life of lightpaths in a block B_i (i.e., τ_{L_i}, τ_{C_i} and τ_{R_i}) are calculated as the difference between the maximum MRL of lightpaths in a block B_i and the MRL of an individual lightpath, i.e., $\tau_{L_i/C_i/R_i} = T_{B_i} - r_{L_i/C_i/R_i}$. Note that the center lightpath (actually an SP) in a block B_i refers to a possible spectrum assignment of the new request; when the spectrum is free to the left or right of a spectrum path SP_i, then τ_{L_i} or τ_{R_i} is considered the same as the normalization constant T_{B_i}. This ensures that cost(sp_i) is lower for those SPs that fit into the gap between two lightpaths as well as the minimum cost SP ensuring that an incoming request and its neighboring lightpaths have minimum variance of MRLs; thus, they will depart together and eventually will minimize fragmentation in both spectral and time domains.

C. Aggregation of Traffic Flows

Aggregation of traffic flows at the electrical layer is the easiest solution to accommodate a new traffic flow request. Thus, it is advisable to route a class-k request R along an existing not fully utilized lightpath, if the lightpath's residual (free) bandwidth is sufficient to satisfy the request's demand (b_k Gb/s), between the same source-destination pair in order to reduce resource consumption and connection blocking. As shown in Fig. 3(a), there is a lightpath LP_1 between nodes A and D, that are assumed

[4]The MRL of a lightpath can be precisely calculated by finding the mean value of the residual life distribution, i.e., $\max(\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n)$. But the solution requires a double integral to be solved, which is complex and time-consuming.

to carry some traffic flows (say, $A_1, A_2, \ldots, A_n$) with aggregated bandwidth demand of B Gb/s. Let us assume that each spectrum slice can support a maximum of C Gb/s with a given modulation format. Therefore, the lightpath LP_1 can accommodate the traffic flow request R easily at the electrical layer if $B + b_k \le C$. In other words, the residual bandwidth of the lightpath LP_1 after allocating the demand b_k is non-negative, i.e., $rb(lp_1) = C - B - b_k \ge 0$. Otherwise, there is another option to expand the bandwidth of lightpath LP_1 by assigning an additional orthogonal frequency-division multiplexing (OFDM) subcarrier to the corresponding transponder, as the slice S_2 is free, and the flow request R could be carried over the expanded lightpath LP_1 if $B + b_k \le 2C$. At the same time, we assume that the transponder's capacity is two or more slices. Nevertheless, when there are more candidate lightpaths that could provision a request, then it is wise to devise the knowledge of their residual (free) bandwidth to select the best one, and the cost function is given by Eq. (4):

$$\text{cost}(lp_i) = p_i \times \left[(1 - \hat{r}b(lp_i)) \times \varepsilon + \frac{|\overline{h}_k^e - \overline{h}_k^p|}{\max(\overline{h}_k^e, \overline{h}_k^p)} \right]. \quad (4)$$

The cost of a candidate lightpath, which could accommodate a new request R, defined in Eq. (4) has two factors apart from a multiplying hop-length factor (p_i) and a constant $\varepsilon = 10^{-1}$. The first factor $(1 - \hat{r}b(lp_i))$ assigns lower cost to a lightpath that has more normalized residual bandwidth ($\hat{r}b(lp_i)$), which is normalized with regard to the lightpath's bandwidth. Thus, aggregation of requests over a not fully utilized lightpath gets lower cost than its expansion. The second factor takes into account the normalized difference between estimated ($\overline{h}_k^e$) and predicted ($\overline{h}_k^p$) mean HTI of the same class-k traffic flows that the new request R belongs. Here, it is important to mention that we store the HTI of the last few departed connections of every class in fixed size first-in first-out (FIFO) queues and calculate its mean value, which is termed as *estimated* mean HTI ($\overline{h}_k^e$). Note that traffic classes are defined by their bandwidth demands. Because the exact knowledge of HTI is not known for individual flows, the ML-based prediction is used to find in advance the *predicted* mean HTI ($\overline{h}_k^p$) of class-k traffic flows (connections). Thus, the difference factor ($|\overline{h}_k^e - \overline{h}_k^p|$) translates the accuracy of prediction into a weight in the cost function in Eq. (4). We explain a prediction mechanism in the next section.

IV. Predicting Time-Varying Traffic with ML

Due to the properties and abilities, e.g., learning and self-X, where X is the configuration, healing, and optimization, AI is well suited for the role of predicting traffic, classifications, and making decisions. Many AI techniques and statistical and learning methods have been widely used to predict time-varying traffic and to perform cognitive lightpath provisioning. Among the linear regression models, an auto-regressive integrated moving average (ARIMA) model and its variants were mostly used for time-series data prediction. A hidden Markov model as a statistical learning method has also been applied to predict short-term traffic on freeways [22]. However, nonlinear regression models, especially learning methods such as support vector machine and neural networks, were evolved for predicting long time-series bursty data with a nonlinear arrival rate, which make them suitable for data center traffic prediction [6]. Because machine-leaning algorithms based on neural networks can extract informative features from raw data, many variants of it, such as deep/recurrent/convolutional neural network methods, have been utilized to predict (also classify) time-series data [7]. The prediction accuracy of these methods comes at the cost of complexity. Recently, [6,19] applied backpropagation neural networks to predict traffic in EONs to efficiently establish and manage lightpaths. However, backpropagation and other neural network models could suffer from exploding and vanishing gradient problems, where the gradient of error functions decreases quickly without improving the learning process. Thus, Hochreiter and Schmidhuber [23] proposed a recurrent neural network architecture called long short-term memory (LSTM) to overcome these error backflow problems. Due to capability of learning long-term dependencies, the LSTM network is useful in classification, processing, and prediction problems. Apart from an input and an output layer, a basic LSTM network has a hidden layer with one or more LSTM blocks, where each block has a memory cell for storing a value for long or short time periods and gates (input, output, and forget) for acting on signals they receive using activation functions.

In this paper, we use the LSTM network model to predict time-varying traffic. It should be noted that ML-based control and management would work efficiently under *periodic* or *known* circumstances or traffic patterns. However, under unexpected traffic patterns, a prediction could be worse. To illustrate this fact, we first generated a time-varying mean and variance and used lognormal distribution to generate a set of data because the actual traces of DC traffic are not available in the public domain. The time-varying mean values (data) can be obtained using the superposition of sinusoidal functions, as given by Eq. (5), where α is an amplitude constant, β_k and ϕ_k are frequency-dependent constants, and n is the number of frequency components (ω_k):

$$m(t) = \alpha + \sum_{k=1}^{n} \beta_k \times \sin(\omega_k t + \phi_k). \quad (5)$$

We used 67% data for training and the remaining 33% for validation and testing of the LSTM network model, which consists of an input layer, a hidden layer with four LSTM blocks, and an output layer for prediction. Note that for the time-series data prediction, the target is the same as the input data. To highlight the accuracy of prediction, Fig. 4 illustrates different RMSEs obtained during training and two sets of testing. Traffic profiles of "Real Data" are shown in blue for all three phases (training, testing 1, and testing 2). As can be seen, RMSE is low for test data set 1,

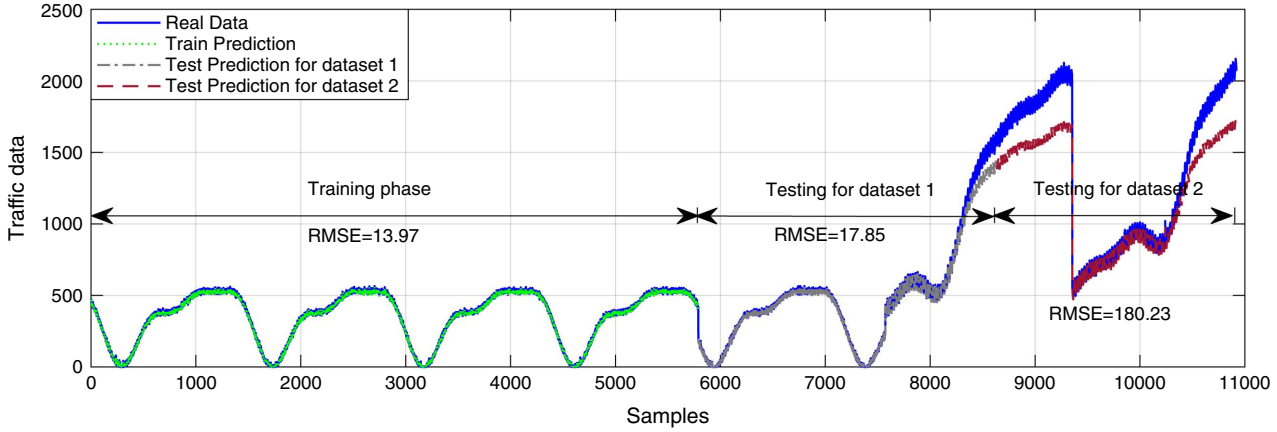

Fig. 4. Data sets used for training and testing and their prediction. Only a fraction of data from different sets are shown here to highlight RMSEs.

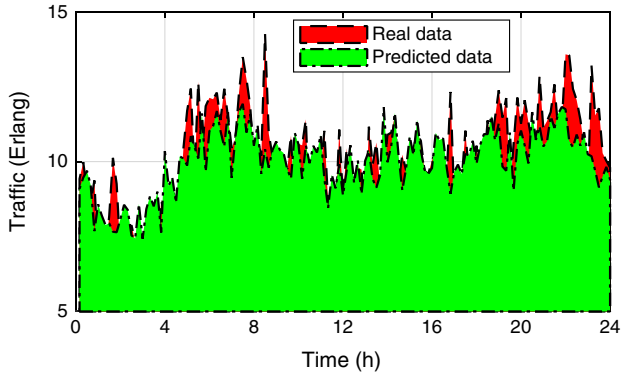

Fig. 5. Time-varying real and predicted traffic during a day.

and it is quite large for test data set 2, because the traffic profile used for testing data set 1 follows the profile of the real data set used for training, but testing data set 2 has a different profile. Thus, it is essential to train ML-based models with different possible traffic profiles in order to obtain a good prediction, i.e., RMSE is close to zero. Figure 5 shows a clearer example of real and predicted data of time-varying traffic during a day. We observe that, although the predicted values differ from exact values, the LSTM model is able to predict the traffic profile. As described in the previous section, when exact knowledge of traffic is unavailable, the predicted mean HTI values for a time-interval can be used to calculate the MRLs of traffic flows given their history and the costs of candidate spectrum paths and lightpaths for allocating resources to a new request.

V. Dynamic Resource (Re)Allocation in ODCNs

In this section, we propose a dynamic ML-based resource provisioning scheme to route intra- or inter-DC flows and (re)allocate the resources, e.g., transponders, fiber cores, and spectrum on-the-fly. This scheme, named as ML-based resource (re)allocation (in short ML-RAR), answers which lightpath or spectrum path should be selected to provision a new class-k request. To this end, the ML-RAR scheme updates fixed-size FIFO queues with the known HTI of departed flows to calculate the variable $\overline{h}_k^e$, i.e., estimated mean HTI of class-k traffic, and uses pre-computed predicted mean HTI for the current time interval, i.e., $\overline{h}_k^p$. Additionally, whenever a new request arrives, the MRL of each candidate lightpath is computed using the knowledge of lifetimes (history) of flows it serves, as described in Section 3. We describe the ML-RAR scheme in Algorithm 1, which is explained below.

First, K-shortest (fiber-level) paths are computed for all source-destination $(s-d)$ pairs. Then, at any given time, two possible events could occur: connection request arrival or departure. When a request arrives between a source s and destination d with demand b_k Gb/s, then the ML-RAR scheme tries to allocate resources and admit it in the network. To achieve this, all possible candidate lightpaths and spectrum paths over allowed cores on its route are found, and their costs are placed in sets $\mathbb{LP}$ and $\mathbb{SP}$, respectively. Note that Eqs. (3) and (4) are used to find costs for spectrum paths and candidate lightpaths, respectively. Among these two sets, a minimum cost path, either an lp_i or sp_j, is selected for resource allocation of R i.e., $\mathbb{RA}(R)$. When a lightpath is selected for this purpose, then demand b_k is aggregated along other flows of lp_i; if needed, a required number of additional subcarriers is added to lp_i, provided if the lightpath's bandwidth does not exceed the transponder's capacity. On the other hand, if a spectrum path sp_j is selected, then a new lightpath is provisioned over sp_j with bandwidth $\lceil b_k/C \rceil + g$ slices, i.e., including guard band g, where C is the capacity per slice. Therefore, a given number of guard bands g separates two adjacent lightpaths in the same fiber core to minimize inter-channel interference within a fiber core, which is a major problem in EON.

However, it is not always possible to find either a lightpath or a spectrum path to admit a new request

due to reasons such as resource unavailability or spectrum fragmentation. In the latter case, even though enough resources are free on the routing path of a connection request, the free spectrum on different links along its path is not aligned as a continuous and contiguous block. Thus, in the absence of a traffic splitting mechanism, one transponder could not provision a lightpath all-optically from source to destination of the new request. Therefore, reconfiguration of resources to traffic flows has been used in the past as an efficient and effective method to realign spectrum, which is popularly known as defragmentation (DF) [8,10,24–28]. At the same time, although defragmentation could also cause traffic disruption, it can be minimized by reducing the number of reconfigurations per DF operation and performing the fast retuning of transceivers and optical switches.

When a network operator allows reconfiguration to be performed, and if the request could not be accepted normally by methods in Steps 3 to 9, then an intelligent reconfiguration process, described in Algorithm 2 as a $\mathcal{DF}-\mathcal{AI}$ scheme, could be triggered (Step 11). We call it intelligent because the DF process is triggered only when predicted blocking (ϕ_B) for the current time interval is equal or more than a threshold (γ_{th}). This avoids excess reconfiguration of lightpaths. For simplicity, the predicted blocking for time-slot Δ_t is calculated as $\phi_B(\Delta_t) = \text{Tr}_p(\Delta_t) \times BP(\Delta_{t-1})/\text{Tr}(\Delta_{t-1})$, where $\text{Tr}_p(\Delta_t)$ is predicted traffic in time-slot Δ_t, $BP(\Delta_{t-1})$ and $\text{Tr}(\Delta_{t-1})$ are the known blocking and traffic, respectively, in previous time-slot Δ_{t-1}. When the DF criteria is satisfied in Step 1 of Algorithm 2, the reconfiguration process tries to set free $\lceil b_k/C \rceil + g$ number of slices to establish a new lightpath for R. For example, in Fig. 3(a), if we assume that the required number of slices to accept the request R between node-pair A to D is $\lceil b_k/C \rceil + g = 3$ slices, then it could not be accepted without shifting some of the lightpaths. Thus, we use an auxiliary graph construction approach [24,27] to find a solution for reallocating some of the lightpaths using a maximum independent set (MIS) algorithm proposed in [29]. The MIS algorithm could find a solution to shift all lightpaths within a reserved spectrum window (for R) to a new nonoverlapping set of spectrum paths outside the reserved window. Using the MIS algorithm, one possible solution to create a new lightpath over spectrum window 2–4 in core-1 is to shift LP_2 from slices 3 and 4 in core-1 to slices 4 and 5 in core-2. However, even a reconfiguration process could not always guarantee the acceptance of a new flow request; hence, if resources cannot be allocated to R, i.e., $\mathbb{RA}(R) = \emptyset$ even after the reconfiguration phase, then it is blocked.

The novel feature of the ML-RAR scheme is that it not only takes prediction into cognizance but also extracts the information from departed connections. When a class-k connection finishes its service, its resources are freed up. Interestingly, most of the requests do not advertise HTI at the time of their arrivals. However, when they depart, the controller could calculate for how long they have been served, i.e., HTI becomes known for departed connections. Therefore, in Steps 16 to 21 in Algorithm 1, fixed size (say 100) FIFO queues store the last 100 departed connections of every class and updates estimated mean HTI $\overline{h}_k^e$, which is utilized in defining the costs of lightpaths in Eq. (4).

Algorithm 1: ML-based resource (re)allocation (ML-RAR)

1: **Given**: Current time-slot Δ_t: network state (NS), guard band g, predicted traffic (λ and $\overline{h}_k^p, \forall\, k$), estimated mean HTI $\overline{h}_k^e, \forall\, k$, predicted blocking ϕ_B, and DF threshold γ_{th}.
2: **if** a connection request $R(s,d,b_k)$ arrives **then**
3: Find all candidate LPs and SPs on allowed cores.
4: Put costs of candidate LPs in $\mathbb{LP}$ and SPs in $\mathbb{SP}$.
5: $\mathbb{RA}(R) \leftarrow min\{cost(lp_i), cost(sp_j)\}, \forall\, i \in \mathbb{LP}, j \in \mathbb{SP}$
6: **if** $lp_i \in \mathbb{RA}(R)$ **then**
7: Accept R and perform $\mathbb{RA}(R)$ along lp_i.
8: **else if** $sp_j \in \mathbb{RA}(R)$ **then**
9: Create a new lightpath over sp_j for $\mathbb{RA}(R)$.
10: **else if** Reconfiguration is allowed **then**
11: $\mathbb{RA}(R) \leftarrow \mathcal{DF}-\mathcal{AI}(R, NS, \phi_B, \gamma_{\text{th}}, g)$.
12: **end if**
13: If $\mathbb{RA}(R) = \emptyset$, block this request;
14: Otherwise, set up a new lightpath for $\mathbb{RA}(R)$.
15: **else** // if a class-k connection (A_k) departs
16: Free resources allocated to A_k.
17: **if** class-k FIFO queue is full **then**
18: Remove an HTI element from this queue.
19: **end if**
20: Add HTI of A_k to the class-k queue.
21: Update $\overline{h}_k^e$.
22: **end if**

Algorithm 2: $\mathcal{DF}-\mathcal{AI}(R, NS, \phi_B, \gamma_{\text{th}}, g)$

1: **if** $\phi_B \geq \gamma_{\text{th}}$ **then** // $|S|$:= # of slices per core
2: Try to set free $\mathbf{b_k} = \lceil b_k/C \rceil + g$ consecutive slices on the route of R as follows. $\mathbb{RA}(R) \leftarrow \emptyset$.
3: **for** $i = 1 : |S| - \mathbf{b_k} + 1$ **do**
4: **if** $\mathbb{RA}(R) = \emptyset$ **then**
5: Reallocate all lightpaths in spectrum interval $[i, i + \mathbf{b_k} - 1]$ on the route of R to other SPs.
6: **else**
7: **return** $\mathbb{RA}(R)$
8: **end if**
9: **end for**
10: **return** $\mathbb{RA}(R)$
11: **end if**

The time and space complexity of Algorithm 1 with reallocation can be given by analyzing four major parts: i) finding the k-shortest spectrum paths and lightpaths; ii) calculating costs; iii) finding a nonconflicting solution using an MIS algorithm; and iv) storing HTI of departed connections. We pre-compute k fiber-level paths for all $N(N-1)s-d$ node pairs in a $|N|$-node network, which requires at max $k \times |N|^2 \times |L|$ memory spaces. Assuming that a lightpath or a spectrum path occupies at least one spectrum slice, the maximum number of lightpaths or spectrum paths (i.e., edges of an auxiliary graph) between a pair of nodes $s-d$ with $|L|$ fiber-links with ζ fiber cores per link, and $|S|$ spectrum slices per fiber core is $|L| \times |\zeta| \times |S|$. Thus, the time complexity for

finding spectrum paths (SPs) or lightpaths (LPs) over k-shortest fiber paths between a pair of nodes $s-d$ is $O(k+|L|\times|\zeta|\times|S|+|N|\times|\zeta|\times|S|\log(|N|\times|\zeta|\times|S|))$, and they (LPs and SPs) would require $2k|L|\times|\zeta|\times|S|$ memory spaces. The costs of a lightpath, accommodating a maximum of c' connections, and a spectrum path can be calculated by Eqs. (3) and (4) in the order $O(c'+2|L|)$ and the minimum among them in $O(|L|\times|\zeta|\times|S|)$. As the reallocation process in Algorithm 2 is limited to within the same core, the time complexity of running an MIS algorithm [29] can be achieved in $O(2^{0.0076|L|\times|S|})$ to find a possible nonconflicting reallocation solution. However, the running time of a parallel MIS algorithm [30] can be achieved in $O((\log(|L|\times|S|))^4)$ but requires $O((|L|\times|S|/\log(|L|\times|S|))^3)$ of memory spaces. The departure of a connection and updating an FIFO queue is a simple process; thus, time and space complexity is $\sim O(c)$, where c is a constant. Therefore, the overall time and space complexity for running the Algorithm 1 with reallocation is the sum of the computation times and spaces required of all major parts; for a large network, it can be approximated as $O(|L|\times|\zeta|\times|S|(|L|+|N|\log(|N|\times|\zeta|\times|S|))+(\log(|L|\times|S|))^4))$ and $O(|N|^2\times|L|+|L|\times|\zeta|\times|S|+(|L|\times|S|/\log(|L|\times|S|))^3)$, respectively.

VI. Performance Evaluation

In this section, we evaluate the performance of the ML-RAR scheme with the time-varying traffic in terms of connection blocking, used transponders, and the number of reconfigurations being performed. We consider an intra- and an inter-ODCN topology in Figs. 6 and 7 based on the architectures [14] and [13,19], respectively, for the simulation. The simulation results are obtained using a discrete event simulation written in Java and evaluated for 24 h. Furthermore, to obtain the steady-state simulation results, we simulate each traffic value for an hour and start the next hour simulation with the previous network state (i.e., spectrum occupancy). Furthermore, we discard the results of an extra run hour at the beginning to get the network into a steady state. For the intra-ODCN, we consider only three hot ToR switches per aggregate optical switch, and each ToR switch is connected to several servers. For simplicity, the number of servers per ToR switch is not taken into account for the simulation. However, traffic is changed after every 1 h such that enough requests are generated for every evaluation time-interval (1 h) between

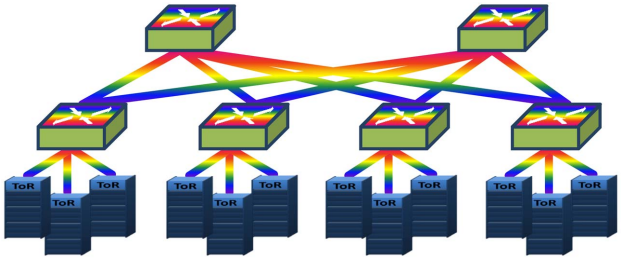

Fig. 6. Intra-ODCN topology with three ToR switches per aggregation switch.

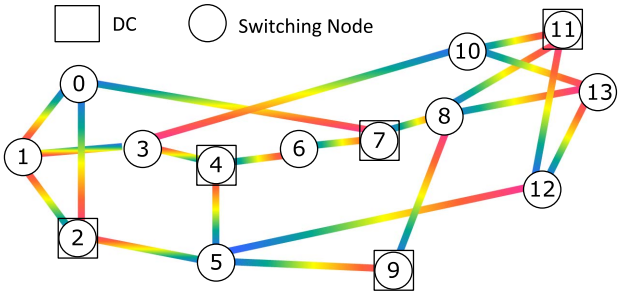

Fig. 7. Simple inter-ODCN topology, where five DCs are interconnected with 14-node NSFNET topology.

ToR switches in intra-ODCN or between DCs in the case of inter-ODCN topology. In inter-ODCN, each DC is directly connected with an optical switch, and requests are generated between any two DCs. The ToR switches in intra-ODCN (Fig. 6) and DCs in inter-ODCN topology (Fig. 7) are interconnected by multicore fiber networks, where each link is bidirectional (consists of two unidirectional multicore fibers), and each fiber consists of three cores, while each core has a total of 200 slices. We assume that a spectrum slice (mini-grid) occupies 12.5 GHz, and a spectrum slice can support aggregated flows with a total demand of $C=25$ Gb/s using the QPSK modulation format. Additionally, we assume that the transponder's capacity is 10 slices, i.e., a lightpath can be established over a maximum of 10 slices. It should be noted that, when a new lightpath is created to serve an incoming demand (b_kGb/s), the lightpath (logically) occupies $\lceil b_k/C\rceil+g$ consecutive slices, where g is the given number of guard bands (in slices), which generally help reduce inter-channel interference between two adjacent lightpaths within a fiber core. However, we do not evaluate the effect of any kind of crosstalk within and among fiber cores in this paper.

In intra-ODCN simulation, we assume that each ToR-level Ethernet switch consists of 54 ports on the server side with three different line rates: 40 ports at 10 Gb/s, 10 ports at 40 Gb/s and four ports at 100 Gb/s. Thus, we consider three different classes of application requests with bandwidth demands $b_1=10$ Gb/s, $b_2=40$ Gb/s, and $b_3=100$ Gb/s (see Table II). As the configuration time in optical DCs is in the order of milliseconds, we focus on application requests with demand $b_k\geq 10$ Gb/s, which can be

TABLE II
Simulation Parameters for Different Traffic Classes

Parameters	Value (for different line rates)		
	10 Gb/s	40 Gb/s	100 Gb/s
Total arrival rate per source (requests per second)	*exponential*: mean $\in\{5, 10\}$, variable		
Arrival ratio (proportional distribution)	40	10	4
Holding time (in seconds)	*lognormal*: time-varying, $c_v\in\{0.25,1.25\}$		

dynamically routed by setting up lightpath(s) upon their arrival, and whose bandwidth requirements are comparable with line rates of Ethernet ports of ToR switches. The same bandwidth classes and their arrival ratio (i.e., 40:10:4) are considered in the case of inter-ODCN simulation. We first pre-compute the two shortest fiber-level paths between all ToR to ToR switches in intra-ODCN topology and between DCs in inter-ODCN topology for routing of these three bit-rate-based traffic flows (connections). Unless otherwise stated, we assume that all three classes are latency-sensitive; thus, they need to be routed immediately over pre-established lightpaths, if available, or over spectrum paths by setting up new lightpaths.

The connection arrival events are generated in each time slot $\Delta_t = \{1, 2\ldots, 24\}$ with exponential (parameter: λ per second) distribution, and a source-destination $(s-d)$ node pair (i.e., ToR–ToR in intra- and DC–DC in inter-ODCN) is selected uniformly. The overall request arrival rate (λ) is uniformly divided among traffic sources, i.e., among the number of ToR switches in the intra-ODCN, and among the DCs in the inter-ODCN. The arrival rate per class at any source node (i.e., λ_s^k) is proportional to the number of ports per class, which is in the ratio of 40: 10: 4. Moreover, each connection departs after finishing its tasks, i.e., after its holding time, which is lognormally distributed with parameters: mean (real) HTI $\overline{h}_k(\Delta_t)$ and coefficient of variation (CoV) c_v. It should be noted that requests' arrival and departure events are generated at each source based on the real traffic per class (i.e., $f(\lambda_s^k, \overline{h}_k)$). To show the effect of HTI and its heavy-tailed characteristics, we fix the mean arrival rate per source $\lambda_s = 10$ for inter-ODCN and $\lambda_s = 5$ for intra-ODCN topology for the entire evaluation period (24 h). The effect of time-varying arrival rate has also been evaluated at the end. The real and predicted mean HTI in the intra-ODCN case (see Fig. 8) is also scaled to half that of inter-ODCN, such that blocking plots are visible and remain below the acceptable limit (10%) in most parts of the evaluation period for the ML-RAR scheme. Thus, the time-varying class-k traffic load is defined as $Tr_{\text{type}}^k(\Delta_t) = N \times \lambda_s^k(\Delta_t) \times \overline{h}_k^{\text{type}}(\Delta_t)$, where traffic type could be real or predicted based on whether mean HTI is real or predicted, and the number of traffic generator sources

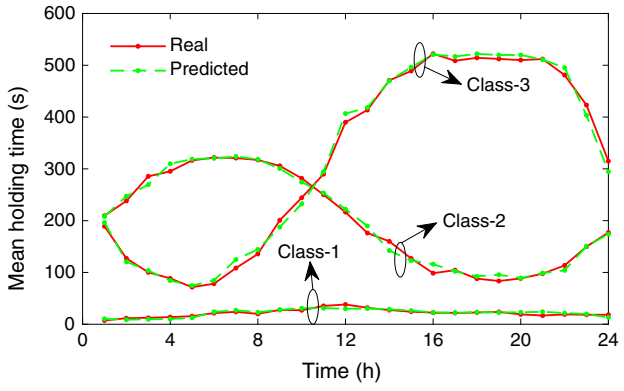

Fig. 8. Real and predicted mean HTI of three traffic classes in Intra-ODCN.

in the intra-ODCN and the inter-ODCN are the number of ToR switches $(N = 12)$ and the number of DCs $(N = 5)$, respectively. Furthermore, we calculate MRLs of connections using Eq. (2), which is a function of the already served time, i.e., the traffic history, its mean, and CoV (or equivalently, μ and σ). It should be noted that, for each class of traffic, we maintain a separate FIFO queue to store HTI of last few (100) departed connections, in order to calculate estimated mean HTI $\overline{h}_k^e$. Unless otherwise stated, arrival rates are fixed as $\lambda_s = 5$ for intra- and $\lambda_s = 10$ for inter-ODCN topology, and connection requests are latency-sensitive and require immediate reservation. Connections are allowed to route over all fiber cores on their paths.

To evaluate the performance of the ML-RAR scheme, we compare it with a first-fit (FF) spectrum allocation policy, which is simple yet effective in a dynamic environment [8,11,25,27]. However, it should be noted that the FF policy considered here as a benchmark is different from the conventional first-fit policy, which tries to find a first (lowest indexed) spectrum path to route every new request. In contrast, the FF policy considered here first tries to find a first available lightpath that has sufficient free bandwidth to aggregate bandwidth demand of a new request; otherwise (i.e., when no lightpath is found), it tries to allocate a lowest indexed spectrum path among all cores to a new request. Additionally, the ML-RAR scheme has been evaluated for three different scenarios: *known*, *predicted*, and *unknown* traffic. When traffic is known, the exact (real) mean HTI, i.e., $\overline{h}_k(\Delta_t)$, is used in the cost functions in Eq. (3) (where Δ_t is omitted for simplicity) to calculate MRLs of connections and lightpaths, and the predicted mean HTI $\overline{h}_k^p(\Delta_t)$ is the same as $\overline{h}_k(\Delta_t)$ in Eq. (4). For the predicted traffic scenario, $\overline{h}_k^p(\Delta_t)$ is used to calculate MRLs and the cost functions in Eq. (3) and Eq. (4), respectively. Interestingly, for the unknown traffic case, neither exact nor predicted mean HTI is known; therefore, the $\overline{h}_k^p(\Delta_t)$ term in Eq. (4) is ignored, i.e., $\overline{h}_k^p(\Delta_t) = 0$, and the MRL of a connection is calculated as if HTI is exponentially distributed, i.e., the MRL is independent of the passed time (memoryless property), and it is equal to the mean value of the unknown HTI. For the unknown mean HTI per class, we averaged the exact time-varying HTI values over 24 h and used this constant mean $(= \frac{1}{24}\sum_{\Delta_t=1}^{24} \overline{h}_k(\Delta_t))$ as the MRL of every connection in all time slots. Thus, the performance metrics (e.g., blocking) obtained for unknown traffic help us to show how important the MRL is, especially in allocating resources to applications with heavy-tailed HTI distribution. Moreover, in all three traffic scenarios, the estimated mean HTI $(\overline{h}_k^e(\Delta_t))$ is calculated by extracting the HTI from departed connections and used in Eq. (4) to calculate the cost of a candidate lightpath.

The connection blocking probability in a multiclass traffic scenario with different bandwidth demands does not present the true information of blocking in EONs; thus, bandwidth blocking ratio (BBR) is often shown in the literature as an alternative blocking measure. Therefore, we now show the BBR obtained for the ML-RAR scheme (without resource reallocation) under three different scenarios

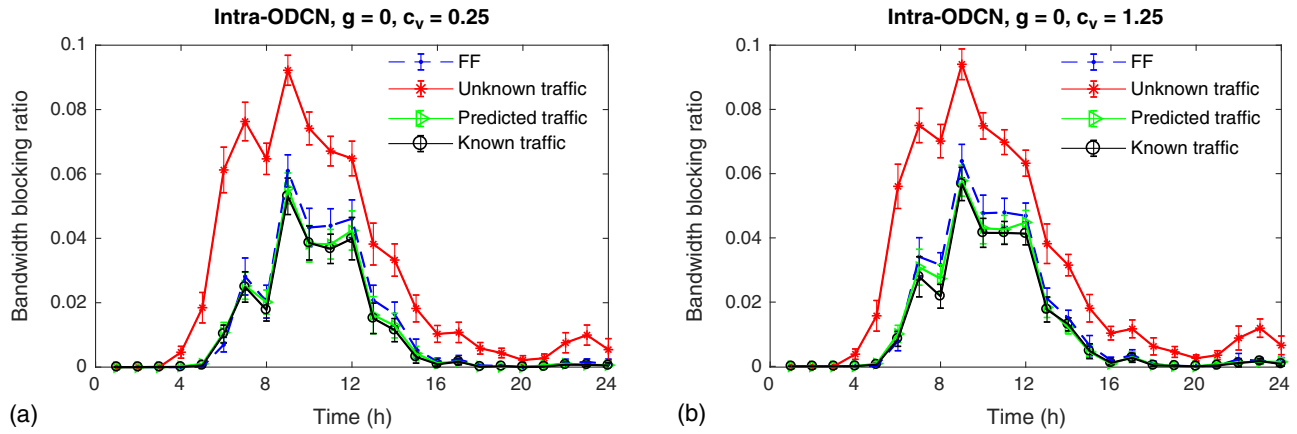

Fig. 9. BBR for FF policy and for various traffic scenarios in intra-ODCN topology are shown for $g = 0$ slice and CoVs: (a) $c_v = 0.25$ and (b) $c_v = 1.25$.

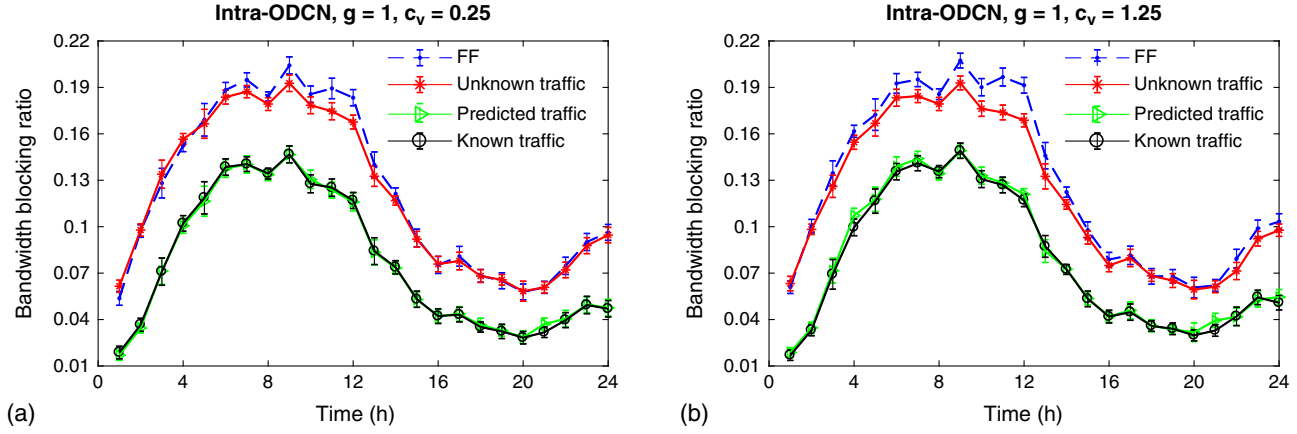

Fig. 10. BBR for FF policy and for various traffic scenarios in intra-ODCN topology are shown for $g = 1$ slice and CoVs: (a) $c_v = 0.25$ and (b) $c_v = 1.25$.

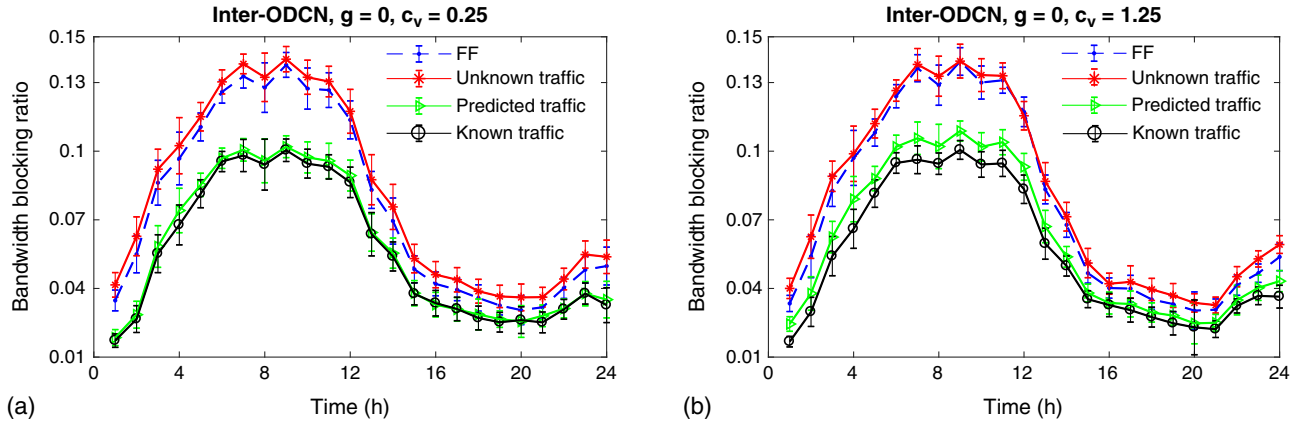

Fig. 11. BBR for FF policy and for various traffic scenarios in inter-ODCN topology are shown for $g = 0$ slice and CoVs: (a) $c_v = 0.25$ and (b) $c_v = 1.25$.

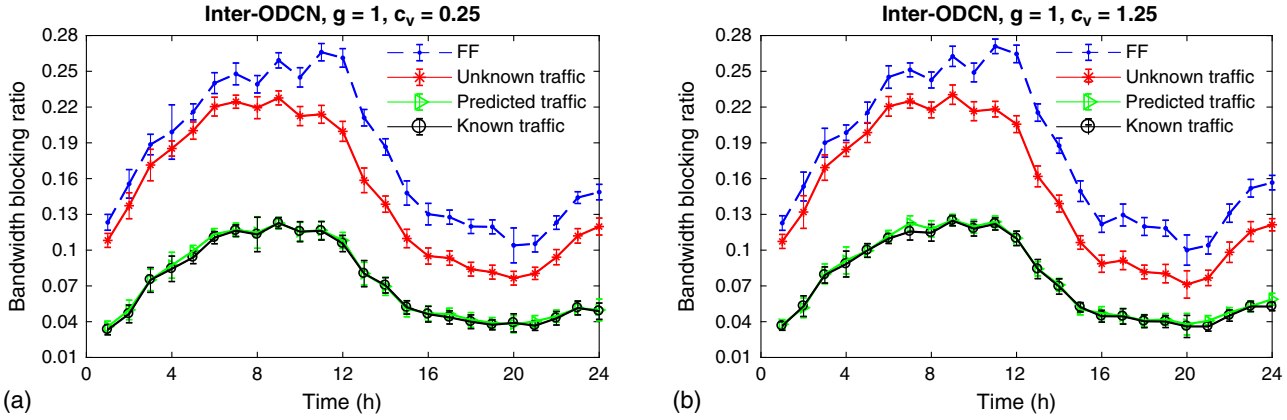

Fig. 12. BBR for FF policy and for various traffic scenarios in inter-ODCN topology are shown for $g = 1$ slice and CoVs: (a) $c_v = 0.25$ and (b) $c_v = 1.25$.

(*known*, *predicted*, and *unknown*) and for the FF scheme in Figs. 9–12. The BBR is calculated per hour, and it is defined as the ratio of blocked bandwidth demands and total requested bandwidth demands of all arriving requests during an hour. The BBRs are shown with a 95% confidence interval for different number of guard bands ($g = \{0, 1\}$ slice) and CoVs ($c_v = \{0.25, 1.25\}$) for intra-ODCN in Figs. 9 and 10 and for inter-ODCN in Figs. 11 and 12. After looking at these figures, we make the following observations: (1) even in the absence of exact knowledge of traffic, the ML-based prediction could be efficiently utilized to allocate resources in ODCNs such that BBR is reduced because the predicted traffic scenario outperforms the FF spectrum allocation policy and unknown traffic, and it is quite close to the BBR of the known traffic case; (2) the utilization of traffic history or active lifetimes of connections and the predicted or exact knowledge of traffic are absolutely necessary for an efficient resource allocation scheme, as shown in Figs. 9–12, in which the BBR is quite high in the case of unknown traffic; (3) the MRL effect alone can be seen for higher CoV ($c_v = 1.25$) as compared with $c_v = 0.25$. We observe that increasing the CoV slightly increases BBRs in all traffic scenarios and in both schemes because more connections with larger holding times are generated even for smaller mean HTI if the CoV is large. However, the increment in BBR is more visible in cases other than the known traffic scenario, which either uses prediction (in predicted traffic case) or simply does not take active lifetime of connections (history) into account (in unknown traffic and the FF scheme), which results in inaccurate estimation of MRLs. Finally, although increasing the guard band would reduce interchannel interference (or crosstalk) between adjacent lightpaths within a fiber core and increase security in optical networks, BBR is increased considerably in all scenarios because the guard bands consume the spectrum that is supposed to be utilized by connection requests. Moreover, we observe that the gap between the known and the predicted approaches is less visible when the guard band is increased in both inter- and intra-ODCN topology. This is due to the fact that, at a higher load (considering guard bands as part of demands), the effect of accuracy of prediction diminishes, and guard bands become a dominating factor in both the known and predicted traffic scenarios. Furthermore, the effect of a guard band on BBR is relatively higher in the unknown and the FF policy than the known and predicted scenarios. The reason is that the former scenarios/policies create newer lightpaths, which is evident in utilizing more transponders per source, as depicted in Fig. 13. It is also important to note that the

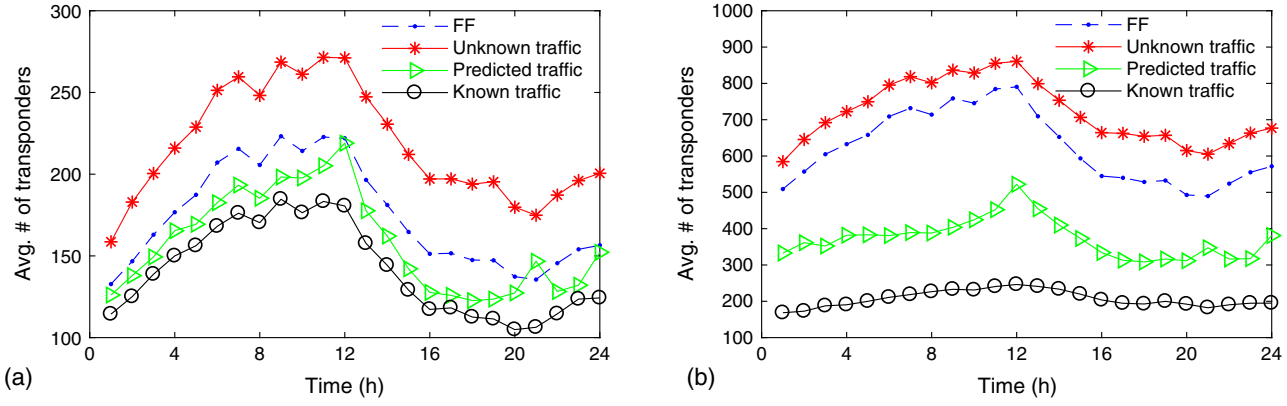

Fig. 13. Average number of transponders per source in (a) intra-ODCN and (b) inter-ODCN topology are shown for $g = 0$ slice, $c_v = 1.25$.

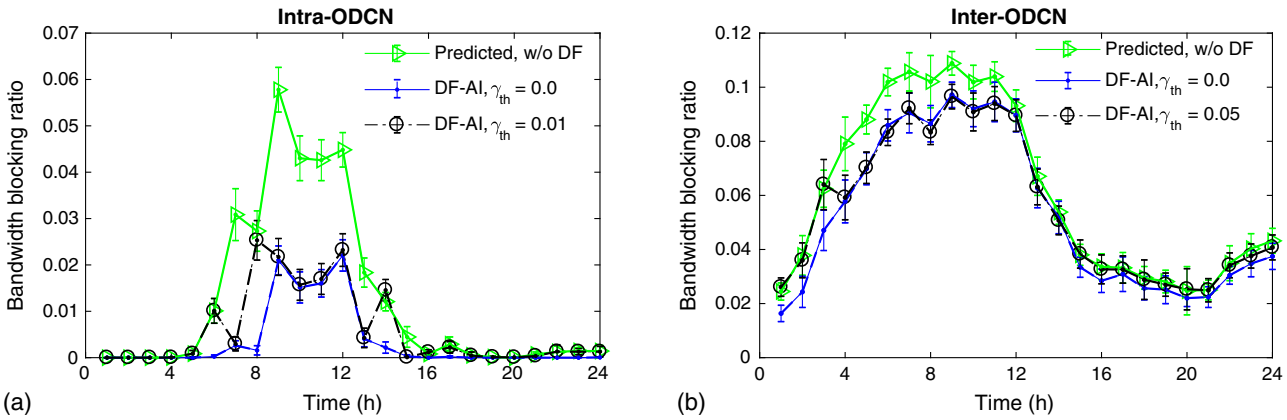

Fig. 14. BBR for predicted traffic and ML-based defragmentation in (a) intra-ODCN and (b) inter-ODCN topology are shown for $g = 0$ slice, $c_v = 1.25$.

inter-ODCN topology uses a much higher number of average transponders per hour than the intra-ODCN, because the offered traffic in the former is four times, and also due to the fact that the inter-ODCN topology provides more disjointed routes than the intra-ODCN topology, which also helps in limiting blocking in the inter-ODCN, despite serving much higher traffic.

It is evident from the previous discussion that traffic prediction is an important factor in handling resources intelligently and efficiently in both types of ODCN networks or in any networks in general. However, as shown in [27], resource reallocation could further improve the BBR performance. Furthermore, an on-demand defragmentation (DF) could be triggered intelligently with the help of predicted BBR for a future time slot. Figure 14 depicts the BBR obtained in the ML-RAR scheme for a predicted traffic scenario with and without enabling the DF-AI feature (explained in Algorithm 2). Furthermore, we introduce a predictive but static DF-AI triggering condition, i.e., whenever the predicted BBR (ϕ_B) crosses the pre-defined threshold (i.e., $\phi_B \geq \gamma_{\text{th}}$), then the DF-AI is triggered. Figure 14 shows BBRs for different thresholds in intra- and inter-ODCN topology, i.e., $\gamma_{\text{th}} = \{0.0, 0.01, 0.05\}$. It should be noted that $\gamma_{\text{th}} = 0$ represents a complex and opportunistic case: Whenever a connection could not be accepted without the reallocation phase (DF-AI), the DF is triggered. Thus, it represents an unintelligent triggering solution leading to the lower bound on the BBR obtained under the ML-RAR scheme with or without the DF phase. Also, we restrict the reconfiguration of lightpaths within the same fiber core, where they are provisioned to reduce the complexity involved in a DF operation and also to limit the traffic disruption caused by reconfigurations of lightpaths over different spectrum, fiber cores, or paths during a DF operation. It can be seen that enabling the DF-AI scheme could reduce blocking by almost an order of magnitude for some time intervals ($\Delta_t = \{6, \ldots, 12\}$), especially in intra-ODCN (likely due to its topology and lesser number of disjoint routes, which cause more spectrum fragmentation). Moreover, the BBR for $\gamma_{\text{th}} = 0.01$ in intra-ODCN and $\gamma_{\text{th}} = 0.05$ in inter-ODCN oscillate between the BBR of predicted traffic without (w/o) the DF case and the DF-AI case with $\gamma_{\text{th}} = 0$. At the same time, it should be noted that a static DF triggering threshold $\gamma_{\text{th}} > 0$ and predicted BBR (ϕ_B) have some flaws, which are reflected on BBRs during time slots $\Delta_t = \{6, 8, 14\}$, wherein the DF-AI is not triggered for $\gamma_{\text{th}} = 0.01$ [see Figs. 14(a) and 15(a)] despite having a moderate load in

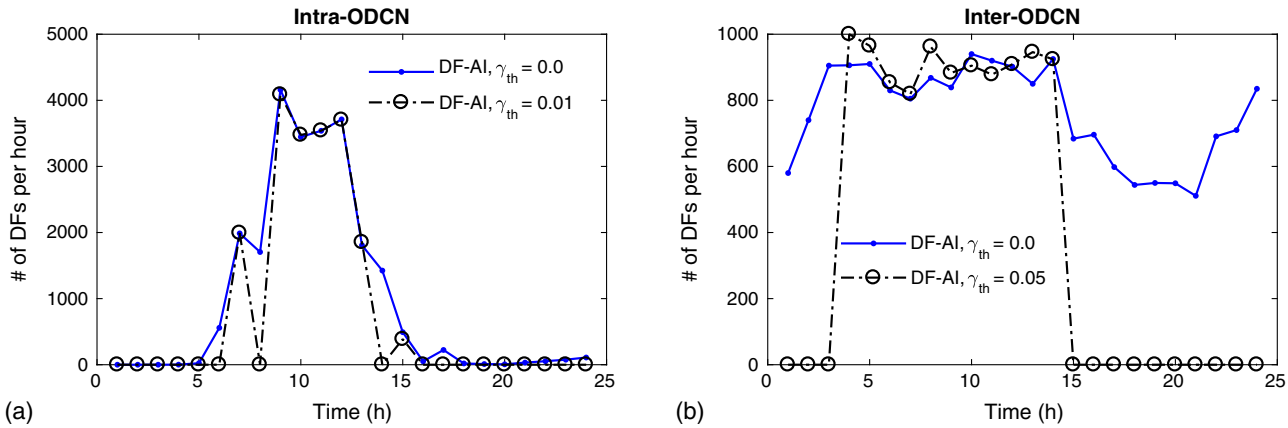

Fig. 15. Number of DFs per hour for predicted traffic in (a) intra-ODCN and (b) inter-ODCN topology are shown for $g = 0$ slice, $c_v = 1.25$.

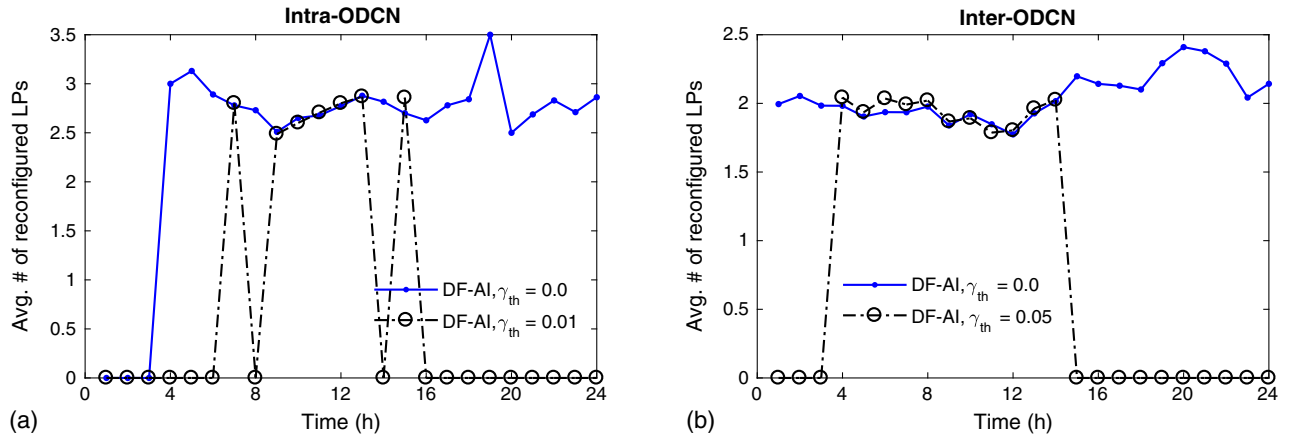

Fig. 16. Number of reconfigured LPs per DF per hour for predicted traffic in (a) intra-ODCN and (b) inter-ODCN are shown for $g = 0$ slice, $c_v = 1.25$.

the intra-ODCN network. The oscillation (a sharp increase and decrease) problem can be overcome if the BBR prediction and blocking threshold are adaptive in nature and consider the following suggestions. First, while predicting the BBR, it must be taken into account whether BBR obtained for the previous interval is obtained with or without a DF-AI scheme. Second, if the BBR obtained in the previous interval is with DF-AI, then the threshold must be slightly lowered for the current interval. Thus, we recommend an intelligent dynamic threshold rather than a static threshold for triggering the DF-AI process. A similar effect is visible in the number of successful DFs in Fig. 15 and the average number of reconfigured lightpaths per hour in successful DF operations in Fig. 16. Here, a successful DF operation represents a DF event that could reallocate some lightpaths to accept a new request. Furthermore, a trade-off between blocking performance and average reconfigured lightpaths can be seen for both topologies. Thus, it is up to network operators to utilize the traffic and corresponding blocking prediction to obtain a balance between blocking and reconfigurations and eventually activate or deactivate some resources (e.g., transponders) in advance.

Until now, the results were presented with the assumption that applications are latency-sensitive and require immediate reservation. In reality, however, elephant or long-lived flows are mostly latency-insensitive. In other words, applications that require larger network bandwidth could be delayed by a few minutes or more, depending on their deadlines. For example, an application requires 3600 gigabytes of data volume to be transferred from one virtual machine to other in different ToRs or DCs, and it needs to be completed within 10 min after the application request arrives at time $t = 0$. Suppose this request is provisioned with a lightpath with a data rate of 100 Gb/s. Therefore, the service time would be 288 s, which means this request could wait for $312 (= 600 - 288)$ s before its start of service. Thus, the effect of the delay of class-3 (100 Gb/s) requests in both ODCN networks is illustrated in Fig. 17 in terms of overall BBR of all classes and average waiting time (delay) of class-3 requests. Note that a class-3 request waits for a maximum of $\tau_d = 100$ seconds; after that, it is blocked if resources are not allocated. For other classes (1 and 2), an immediate reservation is applied. As can be seen, letting some of the class-3 requests wait for a small amount of time results in reducing the overall

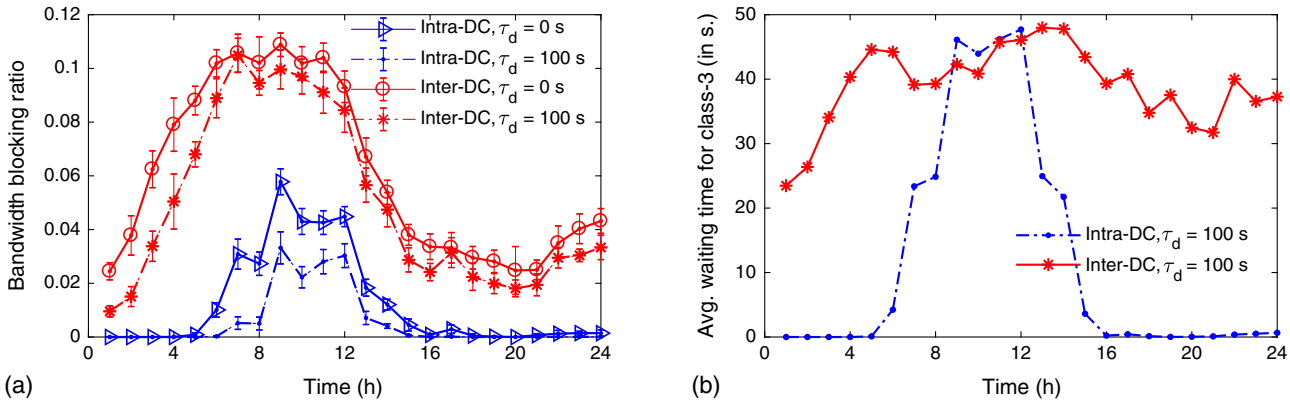

Fig. 17. BBR in (a) intra-ODCN and (b) inter-ODCN topology are shown for $g = 0$ slice, $c_v = 1.25$, for predicted traffic with time-varying arrival rates.

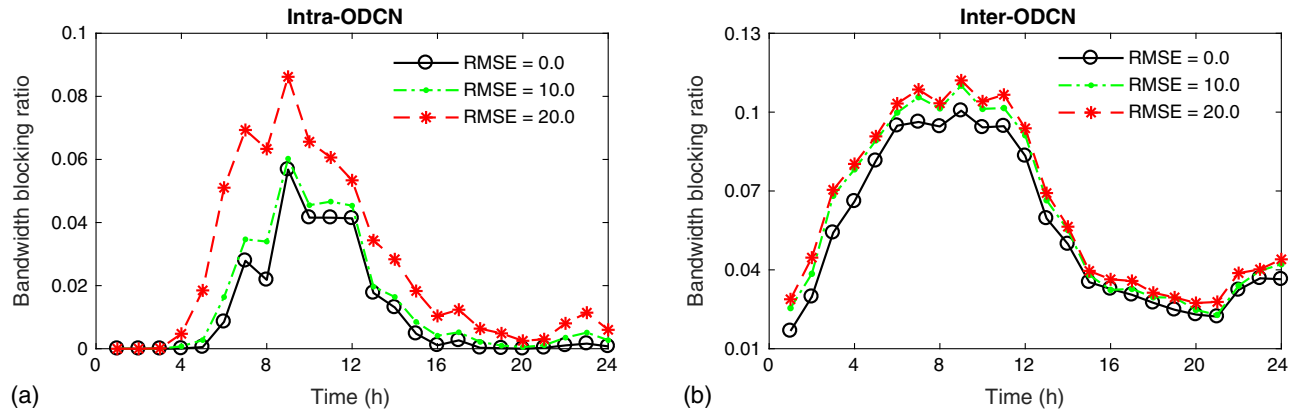

Fig. 18. BBR in (a) intra-ODCN and (b) inter-ODCN topology are shown for $g = 0$ slice, $c_v = 1.25$, for predicted traffic with fixed arrival rates.

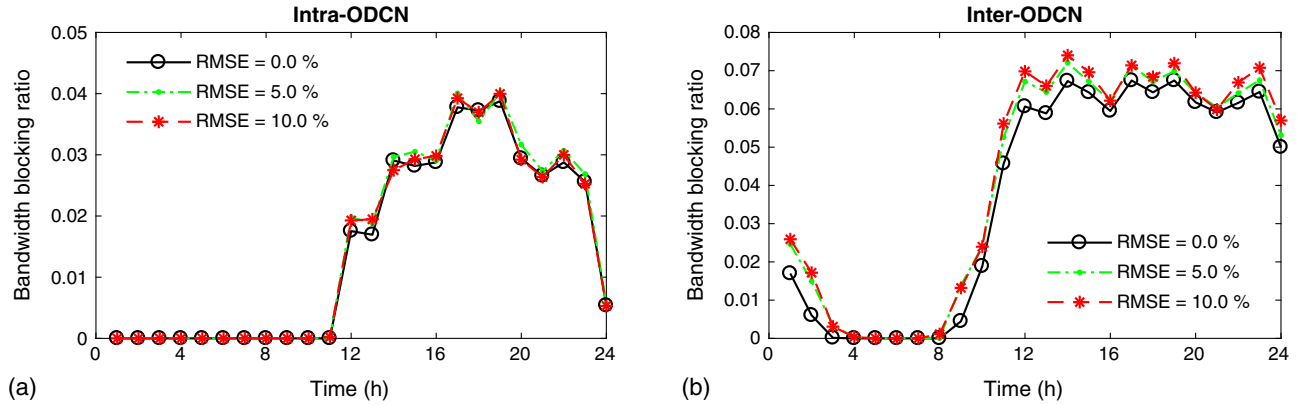

Fig. 19. BBR in (a) intra-ODCN and (b) inter-ODCN topology are shown for $g = 0$ slice, $c_v = 1.25$, for predicted traffic with time-varying arrival rates.

blocking in intra- and inter-ODCNs. The blocking reduction is obtained in both topologies in most of the time intervals where the traffic load is lower and moderate. However, the BBR gain reduces for some time intervals $\Delta_t = \{7, \ldots 12\}$ in the inter-ODCN topology at higher loads, despite the fact that the average waiting time for class-3 requests is relatively high (~40 seconds), as shown in Fig. 17(b). Thus, it is wise to delay the services of some of the latency-insensitive or deadline-agnostic applications or corresponding traffic flows.

Finally, to highlight the importance of prediction accuracy, we calculate the predicted mean HTI by adding absolute RMSE and RMSE in percentage to the exact mean HTI. The exact (real) time-varying mean HTI for intra-ODCN is shown in Fig. 8, and it is doubled for inter-ODCN topology. Again, remember that the real traffic is used to generate connections during simulation, and the predicted mean HTI is used for cost assignment in Eqs. (3) and (4). In Figs. 18 and 19, we present BBR with various RSMEs for intra- and inter-ODCN topology. In particular, Fig. 18 depicts BBR for a fixed arrival rate (5 for intra- and 10 for inter-ODCN topology), and fixed RMSE values are added to the exact mean HTI values to obtain the predicted mean HTI. In contrast, Fig. 19 considers a time-varying arrival rate with the RMSE percentage of the exact mean HTI added to obtain the predicted mean HTI. For intra-ODCN, the arrival rate per source is varied during the evaluation period as $5 - 3\sin(2\pi(\Delta_t - 1)/24)$, where time-interval $\Delta_t = \{1, 2, \ldots, 24\}$, and the arrival rate per source is doubled for the inter-ODCN topology. It is interesting to see that the performance of the ML-RAR scheme depends on how well ML predicts the traffic, i.e., BBR increases with RMSE. Moreover, when the error is in the order of mean HTI, i.e., with 100% RMSE, which is the case for class-1 (mean HTI ~10 seconds) in Fig. 18, then the BBR could be worse. In contrast, when the error is relatively low (in Fig. 19), then the effect of RMSE is not noticeable. Thus, in an unpredictable scenario, when ML would not able to closely predict future event characteristics, an ML-based prediction might lead to inefficient management of optical resources.

VII. Conclusion

In this paper, we proposed a machine-leaning-based resource allocation and reallocation (ML-RAR) scheme, which utilizes the predictive and statistical knowledge of

mean holding times of connections. While we showed that traffic prediction is beneficial in resource (re)allocation, data center traffic is often characterized by heavy-tailed distributions, e.g., lognormal. We utilized the statistical properties of lognormal holding-time distribution and applied them for estimating the mean residual lives of active traffic flows, which eventually proved to be an important parameter for scheduling of time-dependent arrivals. Additionally, we showed that, with an accurate traffic prediction and delaying the latency-insensitive flows, resources can be allocated effectively and intelligently to reduce a number of resources consumed (e.g., transponders, spectrum) by lowering the connection blocking, which can be utilized for future connection requests. Last but not least, prediction-based resource allocation could lead to inefficient management of optical resources in unexpected scenarios for which ML-based models have not been prepared or trained.

References

[1] Cisco, "VNI global fixed and mobile internet traffic forecasts," 2017 [Online]. Available: https://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html.

[2] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," in *10th ACM SIGCOMM Conf. on Internet Measurement*, ACM, 2010, pp. 267–280.

[3] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of data center traffic: measurements & analysis," in *9th ACM SIGCOMM Conf. on Internet Measurement Conf.*, ACM, 2009, pp. 202–208.

[4] A. Roy, H. Zeng, J. Bagga, G. Porter, and A. C. Snoeren, "Inside the social network's (datacenter) network," *Comput. Commun. Rev.*, vol. 45, no. 4, pp. 123–137, 2015.

[5] Y. Chen, S. Jain, V. K. Adhikari, Z.-L. Zhang, and K. Xu, "A first look at inter-data center traffic characteristics via yahoo! datasets," in *IEEE INFOCOM*, IEEE, 2011, pp. 1620–1628.

[6] Y. Li, H. Liu, W. Yang, D. Hu, X. Wang, and W. Xu, "Predicting inter-data center network traffic using elephant flow and sublink information," *IEEE Trans. Netw. Serv. Manage.*, vol. 13, no. 4, pp. 782–792, 2016.

[7] J. Mata, I. de Miguel, R. J. Durán, N. Merayo, S. K. Singh, A. Jukan, and M. Chamania, "Artificial intelligence (AI) methods in optical networks: a comprehensive survey," *Opt. Switching Netw.*, vol. 28, pp. 43–57, 2018.

[8] S. K. Singh and A. Jukan, "Holding-time information (HTI): when to use it?" in *Optical Fiber Communication Conf.*, Optical Society of America, 2017.

[9] M. Klinkowski, M. Ruiz, L. Velasco, D. Careglio, V. Lopez, and J. Comellas, "Elastic spectrum allocation for time-varying traffic in flexgrid optical networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 1, pp. 26–38, 2013.

[10] M. Zhang, C. You, H. Jiang, and Z. Zhu, "Dynamic and adaptive bandwidth defragmentation in spectrum-sliced elastic optical networks with time-varying traffic," *J. Lightwave Technol.*, vol. 32, no. 5, pp. 1014–1023, 2014.

[11] P. S. Khodashenas, J. Comellas, S. Spadaro, J. Perelló, and G. Junyent, "Using spectrum fragmentation to better allocate time-varying connections in elastic optical networks," *J. Opt. Commun. Netw.*, vol. 6, no. 5, pp. 433–440, 2014.

[12] S. Shakya, X. Cao, Z. Ye, and C. Qiao, "Spectrum allocation in spectrum-sliced elastic optical path networks using traffic prediction," *Photon. Netw. Commun.*, vol. 30, no. 1, pp. 131–142, Aug. 2015.

[13] Y. Zong, C. Yu, Y. Liu, Q. Zhang, Y. Sun, and L. Guo, "Time-dependent load-balancing service degradation in optical data center networks," *Photon. Netw. Commun.*, vol. 34, pp. 411–421, 2017.

[14] H. Yang, J. Zhang, Y. Zhao, Y. Ji, J. Han, Y. Lin, S. Qiu, and Y. Lee, "Experimental demonstration of time-aware software defined networking for openflow-based intra-datacenter optical interconnection networks," *Opt. Fiber Technol.*, vol. 20, no. 3, pp. 169–176, 2014.

[15] Y. Zhao, J. Zhang, T. Zhou, H. Yang, W. Gu, Y. Lin, J. Han, G. Li, and H. Xu, "Time-aware software defined networking (TA-SDN) for flexi-grid optical networks supporting data center application," in *IEEE GLOBECOM Workshops (GC Wkshps)*, IEEE, 2013, pp. 1221–1226.

[16] H. Chen, Y. Zhao, J. Zhang, R. He, W. Wang, J. Wu, Y. Wang, Y. Ji, H. Zheng, Y. Lin, and B. Hou, "Time-spectrum consecutiveness based scheduling with advance reservation in elastic optical networks," *IEEE Commun. Lett.*, vol. 19, no. 1, pp. 70–73, 2015.

[17] W.-B. Jia, Z.-Q. Xu, Z. Ding, and K. Wang, "An efficient routing and spectrum assignment algorithm using prediction for elastic optical networks," in *Int. Conf. on Information System and Artificial Intelligence (ISAI)*, IEEE, 2016, pp. 89–93.

[18] M. Tornatore, A. Baruffaldi, H. Zhu, B. Mukherjee, and A. Pattavina, "Holding-time-aware dynamic traffic grooming," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 3, pp. 28–35, 2008.

[19] Y. Xiong, J. Shi, Y. Yang, Y. Lv, and G. Rouskas, "Lightpath management in SDN-based elastic optical networks with power consumption considerations," *J. Lightwave Technol.*, vol. 36, pp. 1650–1660, 2017.

[20] F. Guess and F. Proschan, "12 mean residual life: theory and applications," in *Handbook of Statistics*, vol. 7. Elsevier, 1988, pp. 215–224.

[21] K. Govil and K. Aggarwal, "Mean residual life function for normal, gamma and lognormal densities," *Reliab. Eng.*, vol. 5, no. 1, pp. 47–51, 1983.

[22] Y. Qi and S. Ishak, "A hidden Markov model for short term prediction of traffic conditions on freeways," *Transp. Res. Part C*, vol. 43, pp. 95–111, 2014.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] Y. Yin, K. Wen, D. J. Geisler, R. Liu, and S. Yoo, "Dynamic on-demand defragmentation in flexible bandwidth elastic optical networks," *Opt. Express*, vol. 20, no. 2, pp. 1798–1804, 2012.

[25] S. K. Singh, W. Bziuk, and A. Jukan, "Analytical performance modeling of spectrum defragmentation in elastic optical link networks," *Opt. Switching Netw.*, vol. 24, pp. 25–38, 2017.

[26] S. Ba, B. C. Chatterjee, S. Okamoto, N. Yamanaka, A. Fumagalli, and E. Oki, "Route partitioning scheme for elastic optical networks with hitless defragmentation," *J. Opt. Commun. Netw.*, vol. 8, no. 6, pp. 356–370, 2016.

[27] S. K. Singh and A. Jukan, "Efficient spectrum defragmentation with holding-time awareness in elastic optical networks," *J. Opt. Commun. Netw.*, vol. 9, no. 3, pp. B78–B89, 2017.

[28] Y. Takita, K. Tajima, T. Hashiguchi, and T. Katagiri, "Wavelength defragmentation for seamless service migration," *J. Opt. Commun. Netw.*, vol. 9, no. 2, pp. A154–A161, 2017.

[29] A. Sharieh, W. Al Rawagepfeh, M. Mahafzah, and A. Al Dahamsheh, "An algorithm for finding maximum independent set in a graph," *Eur. J. Sci. Res.*, vol. 23, no. 4, pp. 586–596, 2008.

[30] R. M. Karp and A. Wigderson, "A fast parallel algorithm for the maximal independent set problem," *J. ACM*, vol. 32, no. 4, pp. 762–773, 1985.

Sandeep Kumar Singh received a B.Tech in electronics and communication engineering from the SASTRA University, and M.S. in electrical engineering from the Indian Institute of Technology Madras, Chennai, India in 2010 and 2013, respectively. He is currently working toward a Ph.D. in communication networks with the Technische Universität Carolo-Wilhelmina zu Braunschweig, Braunschweig, Germany. He is recipient of a German Academic Exchange Service (DAAD) scholarship at TU Berlin (2011–2012). His research interests include optical networks, traffic engineering in data centers, stochastic analysis, and machine learning.

Admela Jukan is Chair Professor of Communication Networks at the Technische Universität Carolo-Wilhelmina zu Braunschweig (Brunswick) in Germany. Prior to that post, she was research and visiting faculty at the Institut National de la Recherche Scientifique (INRS), University of Illinois at Urbana Champaign (UIUC) and Georgia Tech (GaTech). In 1999 and 2000, she was a visiting scientist at Bell Labs, Holmdel, NJ, and in 2015 at MIT. From 2002–2004, she served as Program Director in Computer and Networks System Research at the National Science Foundation (NSF) in Arlington, VA. Admela Jukan received her Dr. Tech. degree (cum laude) in Electrical and Computer Engineering from the Technische Universität Wien, a M.Sc. in Information Technologies from the Politecnico di Milano, Italy, and her Dipl. -Ing. from the Fakultet Elektrotehnike i Racunarstva (FER), in Zagreb, Croatia.

Dr. Jukan has chaired and co-chaired several international conferences, including IFIP ONDM, IEEE ANTS, IEEE ICC and IEEE GLOBECOM. She serves as Optical Series Editor for IEEE Communications Magazine and as Senior Editor in the IEEE Journal of Selected Areas in Communications. She is a co-Editor in Chief of the Elsevier Journal on Optical Switching and Networking (OSN). She was an elected Chair of the IEEE Optical Network Technical Committee, ONTC (2014–2015). She is recipient of an Award of Excellence for the BMBF/CELTIC project "100Gb Ethernet" and was also awarded the IBM Innovation Award (2009). She was the Coordinating Principal Investigator of the FP7 EU-Projects PACE and ONE, focusing on innovation in next-generation network management systems. Prof. Jukan's research group is active in research areas related to theory and practice of network architecture and protocols, and network software platforms, as well as theory and performance.