Lightpath Management in SDN-Based Elastic Optical Networks With Power Consumption Considerations

Yu Xiong, Jin Shi, Yaya Yang, Yi Lv, and George N. Rouskas

***Abstract*—Elastic optical networks (EONs) are increasingly used to interconnect data centers. As interdatacenter traffic is expected to continue to grow at high rates for the foreseeable future, it becomes imperative to address the respective growth in power consumption within EONs. In this paper, we take the first steps toward enhancing the energy efficiency of interdatacenter EONs. Specifically, we use traffic prediction techniques for centralized lightpath management that leverages the capabilities of software defined networking (SDN) technology. Our objective is to eliminate unnecessary lightpath termination and re-establishment operations, and in turn decrease switching power consumption within the network. Our evaluation study confirms that the proposed algorithm is effective in achieving substantial savings in power consumption while maintaining a bandwidth blocking ratio at levels comparable to those of earlier algorithms.**

***Index Terms*—Elastic optical networks, energy efficiency, lightpath management, software defined networking, switching power, traffic prediction.**

I. Introduction

With the proliferation of cloud computing, data center (DC) technology and facilities that provide information storage and processing, have emerged as the key infrastructure for supporting essential Internet functionality and services. The rapid growth in Internet users, the explosive increase of traffic demands, and the continuous evolution of service models necessitate an increase in the scale of DC facilities, which in turn leads to higher requirements in terms of inter-DC communication to support data backup, data synchronization, and collaboration between different DCs. The demands placed on inter-DC communication call for appropriate network technologies to interconnect DCs effectively and efficiently.

Elastic optical networks (EONs) [1], [2] are widely regarded as the most promising technology for interconnecting DCs, and have been studied extensively. EONs utilize bandwidth variable optical transponders (BV-OT) and bandwidth variable optical cross-connects (BV-OXC) that operate on a set of spectrally-contiguous frequency slots to set up lightpaths. Since these frequency slots occupy a much narrower bandwidth than the conventional wavelength channels, EONs can provision bandwidth adaptively according to actual traffic demands [3], and hence may meet the requirements of DC traffic. At the same time, the technological heterogeneity and resource diversity between DC and EONs presents a challenge. In order to control the heterogeneous resources uniformly and implement a common overall network management and control strategy, software defined networking (SDN) enabled by the OpenFlow protocol has been introduced into the optical network [4]. The SDN is a virtualization technology that abstracts heterogeneous resources via a unified interface and applies centralized control. At the same time, SDN supports programmability of network functionalities and protocols, which provide a high degree of flexibility for the functions and services with a global view. Therefore, operators consider the application of SDN techniques to control globally network and application resources in DCs and EONs interconnecting them [5].

The accelerating growth of DC traffic means that the power consumption of the inter- and intra-DC networks becomes more prominent and a significant fraction of the power consumed by servers [6]. In fact, how to effectively reduce the power consumption of EONs is a topic that has received significant attention within the research community recently [7]. For instance, a set of power management primitives for network elements were introduced in [8]; these primitives are used to monitor traffic load conditions and turn off network elements (e.g., transponders, etc.) during idle periods. In this manner, power consumption due to elements that are active unnecessarily may be avoided. On the other hand, when a network element in the off state receives new work, a long wake-up time may be incurred before it returns to fully operational status, introducing an undesirable delay in responding to new traffic demands. Therefore, it may be unwise to keep the entire element in sleep status. To address

Manuscript received August 6, 2017; revised October 12, 2017 and November 20, 2017; accepted December 12, 2017. Date of publication December 20, 2017; date of current version February 27, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61401052, in part by the Project of China Scholarship Council under Grant 201608500030, in part by the Science and Technology Project of Chongqing Municipal Education Commission under Grant KJ1400418 and Grant KJ1500445, in part by the Starting Foundation for Doctors of Chongqing University of Posts and Telecommunications under Grant A2015-09, and in part by the Program for Innovation Team Building at Institutions of Higher Education in Chongqing under Grant CXTDX201601020. *(Corresponding author: Yu Xiong.)*

Y. Xiong is with the Key Laboratory of Optical Communication and Networks, Chongqing University of Posts and Telecommunications, Chongqing 400065, China and also with the Department of Computer Science, North Carolina State University, Raleigh, NC 27606 USA (e-mail: cqxiongyu@foxmail.com).

J. Shi, Y. Yang, and Y. Lv are with the Key Laboratory of Optical Communication and Networks, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: shijin730@163.com; 18381333953@163.com; luyi@cqupt.edu.cn).

G. N. Rouskas is with the Department of Computer Science, North Carolina State University, Raleigh, NC 27606 USA, and also with the Department of Computer Science, King Abdulaziz University, Jeddah 21589 3112, Saudi Arabia (e-mail: rouskas@ncsu.edu).

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

Digital Object Identifier 10.1109/JLT.2017.2785410

0733-8724 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

this issue, a fine-grained energy-efficient consolidation strategy was presented in [9] that applies a sleeping scheme at the level of an element's components. Specifically, with this strategy, some components of an element (e.g., transponder) are kept in a working state while others are kept in sleep mode, so that the transponder with the active part can respond to newly arriving requests in a timely manner without incurring a wake-up delay. However, since the capacity of traditional transponders is higher than the bandwidth requested by a typical connection, the result is lower lightpath utilization. The work in [10] introduced a sliceable-transponder that can divide a physical transponder into multiple sub-transponders, each of which can transmit or receive an independent elastic lightpath. The flexibility of transponder provisioning can be achieved in this way so that it can save much power. However, it should be noted that the transponders which can be sliced are generally more expensive than those that do not have this functionality and using more of them will increase the overall network cost.

At the traditional transponder level, the authors of [11] proposed an algorithm named energy-efficient manycast (EEM) to minimize the power consumption by jointly considering the network elements, including BV-OT, BV-OXC, erbium doped fiber amplifier (EDFA) and IP Router. In this study, it was shown that the power savings can be significant when all the network elements contributing to power consumption are taken into consideration. The authors of [12] proposed an algorithm named dynamic scheduling and distance adaptive transmission (DS+DAT) to exploit the over-provisioning capacity in EONs under SDN architecture. The DS+DAT scheme provisions just enough transponders and grooms estimated future traffic to these transponders; the future traffic is estimated by applying an auto-regressive integrated moving average method. As a result, both the amount of lightpaths and the number of transponders are reduced, saving power.

It has been estimated [13] that the amount of power consumed due to frequent establishment of lightpaths contributes to up to 15% of the average power consumption in realistic scenarios. Existing power management strategies for EONs, including the ones discussed above, do not address this aspect: lightpaths are torn down as soon as they become idle, even if future traffic demands might make use of these lightpaths. The premise of our work is that delaying the tearing down of lightpaths in anticipation of future traffic demands will lead to less frequent establishment of lightpaths in the network, cutting down on power consumption. Thus, our work makes three contributions. First, we develop a power consumption model that takes into consideration power consuming resources in both the EON and DCs, and also accounts for the power consumed in setting up lightpaths. Second, we develop a model for predicting future traffic demands by combining concepts from back propagation (BP), neural networks and improved particle swarm optimization (IPSO). Finally, we propose a new, parameterized, power-aware lightpath management (PALM) algorithm that extends the lifetime of idle lightpaths with the goal of serving future demands and avoiding the establishment of new lightpaths. Numerical results to be presented indicate that our approach is successful in achieving significant power savings compared to existing approaches.

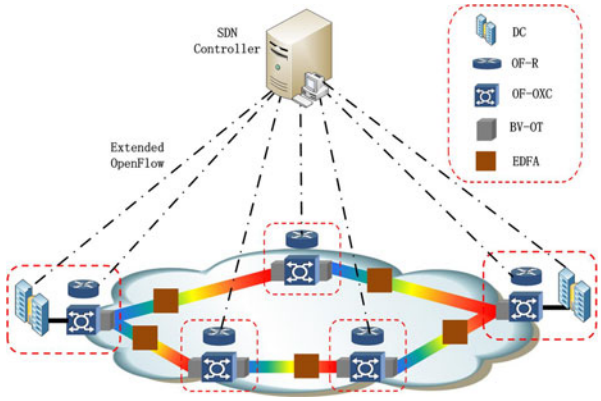

Fig. 1. Software defined EONs for DC application.

The rest of this paper is organized as follows. Section II describes the network architecture and presents the corresponding power consumption model. The traffic prediction model and PALM algorithm are discussed in Section III. We present numerical results in Section IV, and we conclude the paper in Section V.

II. Network Architecture and Power Consumption Model

We consider the software-defined EON architecture for interconnecting DCs as illustrated in Fig. 1. The architecture includes extended OpenFlow-enabled IP Routers and BV-OXCs, which we refer to as OF-R and OF-OXC, respectively. The extended OpenFlow protocol is detailed depicted in [14]. Other elements in the architecture, including EDFAs, BV-OTs and the DCs with DC servers (DC-S) are also shown in the figure. We assume that integration of the EON resources (i.e., network elements) and DC resources (i.e., servers) is realized via the SDN controller at the top of the figure. At the same time, the proposed PALM algorithm is extended to the SDN controller to manage the lightpaths in real time. Due to SDN, the network state information can be converged quickly and accurately, which is benefit to respond to the lightpath provisioning requests in a timely manner.

In Fig. 1, we can see that the underlying network includes several network elements (i.e., OF-R, OF-OXC, BV-OT, and EDFA) and application servers (i.e., DC-S). According to [7], the network elements are considered the main sources of power consumption. However, since our goal is to improve the energy efficiency of the network architecture as a whole, the power consumption of the DC-S cannot be ignored. Hence, we take into consideration both the network elements and application servers in our model.

As is common in studies of network element energy consumption, we also assume that the power of all network elements and application servers includes two parts: a fixed part that is independent of traffic served, and a dynamic part that is dependent on traffic. The fixed part contributes constant power while the element or server is in operation, while the dynamic part represents variable power consumption that is proportional to the traffic that the element or server handles. In addition, our model accounts for the energy consumed for establishing lightpaths in the network. As explained in [15], turning on network elements

TABLE I
NOTATIONS USED IN OUR MODEL

Notation	Meaning
P	Total power consumption in network
N	Set of nodes
Z	Set of BV-OTs of a physical node
S	Set of DC-Ss
R	Set of traffic node pairs
P_f	Total fixed power consumption
P_d	Total dynamic power consumption
P_e	Total switching power consumption
P_E^f	Total fixed power consumption of EDFAs
P_T^f	Total fixed power consumption of BV-OTs
P_R^f	Total fixed power consumption of OF-Rs
P_O^f	Total fixed power consumption of OF-OXCs
P_S^f	Total fixed power consumption of DC-Ss
P_E^d	Total dynamic power consumption of EDFAs
P_T^d	Total dynamic power consumption of BV-OTs
P_R^d	Total dynamic power consumption of OF-Rs
P_O^d	Total dynamic power consumption of OF-OXCs
P_S^d	Total dynamic power consumption of DC-Ss
p_e^f	The fixed power of an EDFA
p_t^f	The fixed power of a BV-OT
p_r^f	The fixed power of an OF-R
p_o^f	The fixed power of an OF- OXC
p_s^f	The fixed power of a DC-S
p_e^d	The dynamic power of an EDFA
p_t^d	The dynamic power of a BV-OT
p_r^d	The dynamic power of an OF-R
p_o^d	The dynamic power of an OF- OXC
p_s^d	The dynamic power of a DC-S

for setting up lightpath consumes a considerable amount of energy which increases linearly with the bandwidth of the served traffic. Importantly, as the study in [13] has shown, energy consumption due to powering on switching elements for setting up a lightpath, is about four times that consumed by the same elements for switching a traffic serving lightpath. Accordingly, our study considers all three components: fixed and dynamic power consumption of network elements and application servers, and switching power consumption for establishing lightpaths.

In deriving the power model, we will use the notation listed in Table I. Let us denote the three components above as P_f, P_d, and P_e. Then, following our discussion, the overall power consumption of the network architecture depicted in Fig. 1 may be calculated as:

$$P = P_f + P_d + P_e \tag{1}$$

Now note that the infrastructure network consists mainly of EDFA, BV-OT, OF-R, OF-OXC and DC-S. Therefore, the fixed component P_f may be obtained as follows:

$$\begin{aligned} P_f &= P_E^f + P_T^f + P_R^f + P_O^f + P_S^f \\ &= \sum_{i\in N}\sum_{j\in N:j\neq i} p_e^f \times W_{ij} + \sum_{i\in N}\sum_{j\in N:j\neq i} p_t^f \times O_{ij} \\ &+ \sum_{i\in N}\sum_{j\in N:j\neq i} p_r^f \times Q_{ij} + \sum_{i\in N}\sum_{j\in N:j\neq i} p_o^f \times K_{ij} \\ &+ \sum_{s\in S} p_s^f \times F_s \end{aligned} \tag{2}$$

where W_{ij} is the number of EDFAs on lightpath ij, O_{ij} is the number of BV-OT pairs on lightpath ij, Q_{ij} is the number of OF-Rs on lightpath ij, K_{ij} is the number of OF-OXCs on lightpath ij, F_s is a binary which is 1 if the server s stays working.

Similarly, the dynamic component P_d can be calculated as follows:

$$\begin{aligned} P_d &= P_E^d + P_T^d + P_R^d + P_O^d + P_S^d \\ &= \sum_{i\in N}\sum_{j\in N:j\neq i}\left(p_e^d \times \sum_{sd\in R} R_{sd} \times X_{ij}^{sd}\right) \\ &+ \sum_{i\in N}\sum_{j\in N:j\neq i}\left(p_t^d \times \sum_{sd\in R} R_{sd} \times X_{ij}^{sd}\right) \\ &+ \sum_{i\in N}\sum_{j\in N:j\neq i}\sum_{a\in Z}\sum_{b\in Z}\left(p_r^d \times \sum_{sd\in R} R_{sd} \times Y_{ab}^{ij}\right) \\ &+ \sum_{n\in N}\left(p_o^d \times \sum_{sd\in R} R_{sd}\times K_n\right) + \sum_{s\in S}\left(p_s^d \times \sum_{sd\in R} \Upsilon_{sd}\times K_s\right) \end{aligned} \tag{3}$$

where R_{sd} is traffic demand (Gb/s) of node-pair sd and Υ_{sd} is the traffic demand (Gb/s) of DC-S s. The X_{ij}^{sd} is a binary which is 1 if traffic between node-pair sd uses lightpath ij, the Y_{ab}^{ij} is a binary which is 1 if lightpath ij uses BV-OTs ab and nodes ij, the K_n is a binary which is 1 if OF-OXC n stays working and the K_s is a binary which is 1 if DC-S stays active.

Finally, the total switching power for setting up lightpaths in the network may be derived as:

$$P_e = \sum_{i\in N}\sum_{j\in N:j\neq i}\sum_{a\in Z}\sum_{b\in Z} p_{e,ab}^{ij} \tag{4}$$

Let p^{ij} denote the power consumption of an active lightpath ij. p^{ij} is composed of holding power (necessary to keep the lightpath on) and transmission power (necessary to serve traffic). Hence, the power consumption p^{ij} for an active lightpath ij may be written as:

$$p^{ij} = p_f^{ij} + p_d^{ij} \tag{5}$$

In the above expression, p_f^{ij} denotes the holding power (a fixed component) and p_d^{ij} denotes the transmission power (a dynamic component). The EDFA, BV-OT, OF-R and OF-OXC contribute to the fixed component p_f^{ij} to keep the lightpath on, while the dynamic component p_d^{ij} is proportional to the amount of traffic carried by the lightpath. Therefore, we may express the two components of power consumption for active lightpaths as:

$$p_f^{ij} = p_e^f \times W_{ij} + p_t^f \times O_{ij} + p_r^f \times Q_{ij} + p_o^f \times K_{ij} \tag{6}$$

$$\begin{aligned} p_d^{ij} &= p_e^d \times \sum_{sd\in R} R_{sd} \times X_{ij}^{sd} + p_t^d \times \sum_{sd\in R} R_{sd} \times X_{ij}^{sd} \\ &+ p_r^d \times \sum_{sd\in R} R_{sd} \times Y_{ab}^{ij} + p_o^d \times \sum_{sd\in R} R_{sd}\times K_n \end{aligned} \tag{7}$$

Finally, according to [13], the switching power of setting up a lightpath is about four times that of the same lightpath in active mode. Therefore, following the above analysis, the switching

power involved in setting up a lightpath is as follows:

$$p_{e,ab}^{ij} = 4p^{ij} = 4(p_f^{ij} + p_d^{ij})$$
$$= 4\left(p_e^f \times W_{ij} + p_t^f \times O_{ij} + p_r^f \times Q_{ij} + p_o^f \times K_{ij} + p_e^d \times \sum_{sd\in R} R_{sd} \times X_{ij}^{sd} + p_t^d \times \sum_{sd\in R} R_{sd} \times X_{ij}^{sd} + p_r^d \times \sum_{sd\in R} R_{sd} \times Y_{ab}^{ij} + p_o^d \times \sum_{sd\in R} R_{sd} \times K_n\right) \quad (8)$$

Due to the high-bandwidth characteristics of inter-DC traffic, the transmission power of lightpaths to serve this traffic, as shown in expression (7), is substantial. Consequently, the switching power of setting up a lightpath, as expressed in (8)) is also considerable and represents a significant fraction of the overall power in the network. While there is not much one can do with respect to transmission power (after all, a sufficient number of lightpaths must be on to carry the traffic), avoiding unnecessary lightpath teardown and setup operations may have a significant impact in power consumption across the network. Our premise is that power-aware management of lightpaths in such a network environment may provide significant power savings, and in the following we present our approach to reducing the frequency of power-inefficient lightpath setup operations.

III. Algorithm Design

A. Traffic Prediction

Consider a lightpath that is about to be terminated. If we knew, or could predict, that the same lightpath will be needed again a short time later, we could extend its lifetime until new traffic is available to use it, hence avoiding power-consuming operations to tear down and re-establish the same lightpath. Let $C(l_e, t)$ denote the traffic load of lightpath l_e at time t. Our goal is to use appropriate prediction methods to obtain an estimate of the lightpath's traffic load in the near future. Note that DC traffic characteristics are closely related to user behavior, which in turn is affected by both subjective and objective factors. Therefore, network traffic in a DC scenario is nonlinear and exhibits self-similarity and long-term correlation, making traditional linear prediction models unsuitable for estimating accurately future traffic loads. BP neural networks [16], on the other hand, have excellent nonlinear and strong self-learning characteristics, and represent promising prediction models for complex DC traffic.

The learning process of the BP neural network is composed of forward propagation of information and back propagation of error messages, as shown in Fig. 2. In forward propagation, the input signal is transferred from the input layer to the output layer through one or more hidden layers. If the output layer does not obtain the desired result, then the error message is returned via back propagation, and the thresholds and weights are adjusted by constant training until the error is reduced below a specified threshold [16]. In this paper, we use the three-layer BP network model depicted in Fig. 2.

The BP neural network of Fig. 2 has n input layer nodes, m hidden layer nodes, and one output layer node. Let $w_{ij}, i = 1, \ldots, n, j = 1, \ldots, m$, denote the weight of the link between input node i and hidden node j, and w_{jo} be the weight of the link between hidden node j and output node o. Then the output of the hidden layer is expressed as:

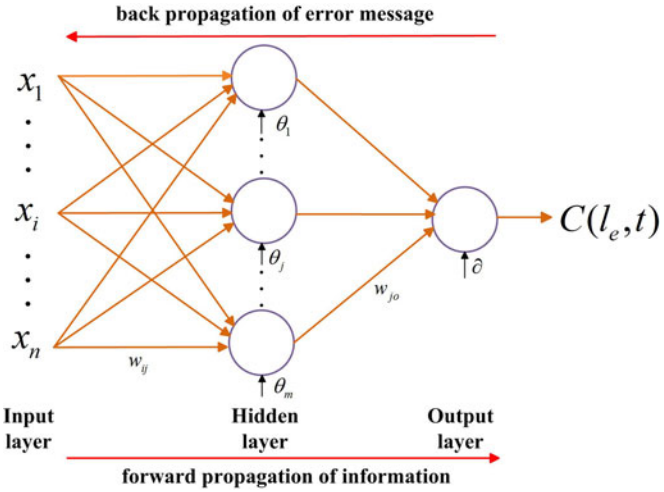

Fig. 2. Configuration of a 3-layer BP network model.

$$h_j = f\left(\sum_{i=1}^{n} w_{ij}x_i - \theta_j\right), j = 1, 2, \ldots, m \quad (9)$$

where θ_j is the threshold of the hidden layer nodes, and $f(\cdot)$ is a nonlinear transfer function. In this paper, we use the sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}}$$

as the transfer function. Thus, the output of network may be obtained as:

$$C(l_e, t) = \sum_{j=1}^{m} h_j w_{jo} - \partial \quad (10)$$

where ∂ is the threshold of the output layer node.

Given the desired actual output $\hat{C}(l_e, t)$ and neural network output $C(l_e, t)$, the network error e is calculated as:

$$e = \frac{1}{2}\left(C(l_e, t) - \hat{C}(l_e, t)\right)^2 \quad (11)$$

In the training process, the weights and thresholds are adjusted to minimize the error e. Specifically, the weights are updated as follows:

$$w_{ij} = w_{ij} + \eta h_j(1 - h_j)x_i \sum_{j=1}^{m} w_{jo}\, e \quad , i = 1, 2, \ldots, n$$
$$w_{jo} = w_{jo} + \eta h_j e\,, j = 1, 2, \ldots, m \quad (12)$$

where η is the learning rate, a positive constant less than 1. The threshold is also updated as:

$$\theta_j = \theta_j + \eta h_j(1 - h_j) \sum_{j=1}^{m} w_{jo}\, e \quad , j = 1, 2, \ldots, m$$
$$\partial = \partial + e \quad (13)$$

The BP neural network model is the most widely used prediction model, as it has the advantages of simple structure, strong plasticity, and excellent ability of approximating nonlinear mapping. But there are two obvious disadvantages in this model: first, the model may often get trapped in a local minimum value, and second, it converges slowly. To overcome these drawbacks of BP network, in this work we apply a particle swarm optimization (PSO) algorithm in [16] to improve the BP neural network model. The excellent global optimization ability of the PSO algorithm makes it a natural candidate for training the BP neural network model so as to optimize its weights for short-term network traffic forecasting.

Similar to the Genetic algorithm, PSO is a population-based algorithm with each individual or candidate solution being called a "particle". The basic PSO model consists of a swarm of particles moving in a d-dimensional search space where a certain quality measure, the fitness, can be calculated. Each particle has a position, represented by a vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$, and a velocity, represented by vector $v_i = (v_{i1}, v_{i2}, \ldots, v_{id})$, where i is the index of the particle. While exploring the search space for an optimal solution, each particle remembers two variables: the best position this particle has found so far, denoted by p_i, and the best position found by any particle in the swarm, denoted by p_g [17]. As time passes, each particle updates its position and velocity to a new value according to expressions (14) and (15).

$$v_i(t+1) = \omega v_i(t) + c_1\, rand(0,1)(p_i - x_i(t)) + c_2\, rand(0,1)(p_g - x_i(t)) \quad (14)$$

$$x_i(t+1) = x_i(t) + v_i(t+1), i = 1, 2, \ldots, n \quad (15)$$

In the above expressions, ω is called the inertial factor and is described by the following equation:

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{T_{\max}} k \quad (16)$$

where $T_{\max}$ is the number of iterative generations, k is the present iterative generation, c_1 and c_2 are positive constants referred to as acceleration constants, and $rand(0,1)$ is a random number uniformly distributed in the range $[0,1]$. In general, the value of each component in v_i may be clamped to the range $[-v_{\max}, +v_{\max}]$ to control excessive roaming of particles outside the search space. The particle moves towards a new position according to expressions (14) and (15). This process is repeated until a user-defined stopping criterion is reached.

But, the species scale G of traditional PSO algorithm is fixed which is based on user experience. Actually, a fixed species scale usually deteriorates algorithm's performance, such as the quality of the optimal solution and the running time. For this reason, we design a dynamic species strategy that the species scale G changes adaptively as the algorithm performs. Specifically, the average distance from each particle to all the other particles is calculated in each generation of evolution. Then, the algorithm compares the average distance calculated by the global optimum particle with the average distance calculated by other particles in the current generation. If the former is smaller, it shows that the species is likely to fall into the local optimum. At this point, to maintain species diversity and to make the algorithm jump out of the local optimum, a particle is added to the original species scale. Otherwise, a particle will be reduced to the species scale.

Algorithm 1: Improved Particle Swarm Optimization for BP Neural Network (IPSO-BPNN).

Input: BP neural network structure, particle dimension d, inertia factor ω, species scale G, the maximum number of iterations $T_{\max}$, the particle position x_i and velocity v_i, the neural network weights (w_{ij} and w_{jo}) and thresholds (θ_j and ∂);

Output: The prediction result of traffic demand for lightpaths.

for $\forall k \in T_{\max}$ **do**
  Calculate the mean square error of each particle;
  if the optimal position of the particle is better than the target value **then**
    The optimal position of the particle is selected as the optimal position of the individual;
  end if
  if p_i is superior to p_g **then**
    $p_g = p_i$;
  end if
  if the average distance calculated by the global optimum particle is less than the average distance calculated by the other particles **then**
    $G = G + 1$;
  else
    $G = G - 1$;
  end if
  Update x_i and v_i by formula (14) (15);
end for
The optimal output values (w_{ij}, w_{jo}, w_{jo} and ∂) are substituted into the BP neural network for learning and training;
Run BP neural network algorithm;
Return the prediction result of traffic demand for lightpaths;

We use the improved PSO (later expressed as IPSO) algorithm described above to train the initial weights (w_{ij} and w_{jo}) and thresholds (θ_j and ∂) of the BP neural network. The optimized weights and thresholds are then used to carry out the BP algorithm only if the error trends to a certain stable value. The improved algorithm has the advantage of faster convergence and more accurate prediction. A pseudocode description of the IPSO for BP neural network is provided as Algorithm 1.

The data used to serve the IPSO-BPNN is from the inter-DC links at X company, one of largest Internet companies in China. The links serve as X's inter-DC backbone, connecting multiple data centers with tens of thousands of servers. These data centers host hundreds of large scale applications, both interactive and batch. Fig. 3 depicts the variation of traffic load in the inter-DC links at X for a period of one week. And the actual traffic in the first six days is used as the historical data for the learning of BP neural network. Then the prediction result of the last day is obtained by IPSO-BPNN which is shown in the picture. Due to

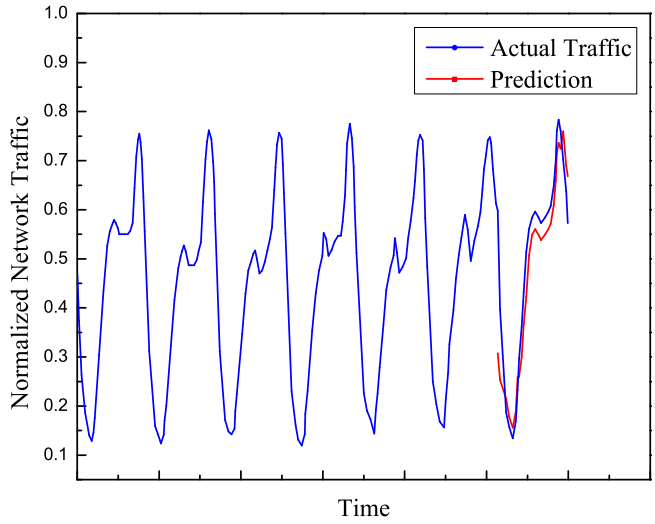

Fig. 3. The actually traffic and prediction using IPSO-BPNN.

confidentiality concerns, we normalize the Y-axis of all figures so we do not reveal the actual amount of data transfers. The normalization does not affect the results of this paper. It is the evident that the IPSO-BPNN captures the behavior of the past traffic profile and the mean square error can be kept below 3%.

B. Proposed Algorithm

The objective of the proposed power-aware lightpath management (PALM) algorithm is to avoid unnecessarily tearing down a lightpath between a source-destination pair if there is a reasonable expectation that there will be a request for setting up a lightpath between the same pair of nodes in the near future. By keeping the lightpath about to be teared down active, the switching power involved in the establishment of a future lightpath can be avoided; as a result, the term P_e in expression (1) will be reduced. Although, in principle, PALM may be deployed in non-SDN architectures, e.g., a GMPLS-based control plane, in this work we only study its benefits in the context of SDN. More specifically, once the SDN controller detects an idle lightpath (the lightpath to be removed) from the database module, it uses the algorithm 1 that we discussed in the previous subsection, to determine a *holding time* t for the lightpath. The maximum holding time, $t_{\max}$, of the idle lightpath may be obtained from expression (17):

$$p_{e,ab}^{ij} \times T = p_f^{ij} \times t_{\max} \tag{17}$$

where T is the time it takes to set up a lightpath within the SDN architecture. In [18], the authors have estimated that the control plane latency for setting up a lightpath is around 23 ms, i.e., $T =$ 23 ms, and this is the figure we use in this work. Combining the above with expressions (5) and (17), we obtain $t_{\max}$ as:

$$t_{\max} = \frac{p_{e,ab}^{ij} \times T}{p_f^{ij}} = \frac{4(p_f^{ij} + p_d^{ij}) \times T}{p_f^{ij}} \tag{18}$$

We now focus on estimating accurately the holding time t of an idle lightpath. To this end, we use the algorithm 1, that we described above. Specifically, we decide to keep an idle lightpath on (i.e., *hold* it in active status) for a value of time equal to $t, t \leq t_{\max}$, if the prediction results indicate one of two possibilities:

- *Case 1:* A traffic demand that will request the idle lightpath (we call the traffic *Initiative Traffic* (IT)) is expected to arrive at time t in the future.
- *Case 2:* A traffic request for a lightpath between the same source-destination pair as that of the idle lightpath, but on a different physical path, is expected to arrive at time t. In this case, we reroute the new traffic request to the idle lightpath (the rerouting traffic is called *Passive Traffic* (PT)) only if (1) the new traffic would lead the utilization of its original lightpath above a certain threshold $M\%$, and (2) transmission on the idle lightpath results in power consumption no greater than transmission on its original lightpath. Finally, we note that the lightpath spectrum must satisfy the spectrum continuity and spectrum contiguity constraints when the routing is carried out [1], [19]. Note that, in *Case 2*, to improve the resource utilization of lightpath and reduce the bandwidth blocking ratio, we set the threshold $M\%$ dynamically. More concretely, when an idle lightpath appears in the network, a threshold is set for the adjacent lightpath of the idle lightpath, and the threshold is cancelled immediately when the idle lightpath is reused to serve traffic.

Algorithm 2: Power-Aware Lightpath Management (PALM).

Input: An idle lightpath l_i to be torn down, initial network topology G_p;
Output: The holding time t for idle lightpath, total power consumption P;

for $\forall l_e$ **do**
  $C(l_e, t) \leftarrow IPSO - BPNN(l_e, t \leq t_{\max})$;
end for
for $\forall l_i$ **do**
  Search all lightpaths L_n that have the same source destination pair as l_i;
  Set the threshold ($M\%$) for L_n;
  if $(C(l_i, t) \in IT) > 0$ **then**
    Cancel the thresholds ($M\%$) of L_n;
    $HT(l_i) \leftarrow t$;
    $break$;
  end if
  if $(C(l_i, t) \in PT) > 0$ **then**
    $HT(l_i) \leftarrow t$;
    $break$;
  end if
  if $(C(l_i, t) \in IT) \leq 0 \&\& (C(l_i, t) \in PT) \leq 0$ **then**
    Tear down the idle lightpath l_i;
    Cancel the thresholds ($M\%$) of L_n;
  end if
end for
Calculate the total power consumption based on expression (1);

A pseudocode description of the PALM algorithm is provided as Algorithm 2. The traffic prediction module firstly runs the algorithm 1 (IPSO-BPNN) to predict the traffic load for every lightpath l_e within the next $t_{\max}$ time units, and the value $C(l_e, t)$ of traffic for every lightpath at time t ($t \leq t_{\max}$) is estimated and stored in the module (Lines 1-3).

For each idle lightpath l_i between a source destination pair (s, d) to be torn down, All other lightpaths L_n between (s, d) are searched by the traffic management module firstly. And then the thresholds are set for all the lightpath L_n (Lines 4-6). Then the PALM module checks every lightpath's traffic load at time t ($C(l_e, t)$) stored in the traffic prediction module. If it finds a IT request will arrive to the idle lightpath l_i, the thresholds of L_n will be cancelled and the t will be set for this idle lightpath's holding time (Lines 7-9). Then it will break the loop to return the power consumption of whole network based on expression (1) (Line 19). Otherwise, if the PALM module finds a PT request will arrive to the idle lightpath l_i, the t will be also set for this idle lightpath's holding time (Lines 12-13), and then break the loop (Line 14). The idle lightpath l_i will be torn down if there is no traffic (include IT and PT) request within $t_{\max}$ and the thresholds of L_n will be cancelled (Lines 16-18). Then the algorithm returns the power consumption of whole network based on expression (1) (Line 19).

The PALM algorithms assumes that there is no information about future lightpath arrivals, and invokes the IPSO-BPNN module to make decisions based on predictions on future traffic demands. Under an alternative model of advance reservations [20], [21], information about future demands (e.g., the start time, deadline, and/or holding time of lightpaths) may be known to the network operator. In this case, it would be possible to modify and refine the PALM algorithm to take this information into account in making decisions on whether and how long to maintain an idle lightpath. However, we note that extending PALM for advance reservations is outside the scope of this work, but is a possible direction of future research.

IV. Numerical Results

We now present a set of simulation results to evaluate the performance of the proposed PALM algorithm.

A. Simulation Setup

To assess the benefits of the proposed algorithm, we leverage the Mininet+Floodlight and Python simulation tool building test platform. For the experiments, we use the 14-node, 21-link NSFNET network topology with DCs at nodes 2, 5, 6, 10 and 12, and the 24-node, 43-link USNET network topology with DCs at nodes 1, 2, 6, 9, 12, 19, and 22 [22] shown in Fig. 4. We assume that there is one pair of bi-directional fiber on each link, and the available spectrum width of each fiber is set to 4000 GHz. To guarantee the quality of transmission (QoT), in our simulation, we adopt the adaptive modulation-level selection in [23]. Specifically, with B_i as the lightpath's bandwidth requirement in Gb/s, the number of FS' we need to assign is:

$$n = \frac{B_i}{m \cdot B_{grid}^{BPSK}}$$

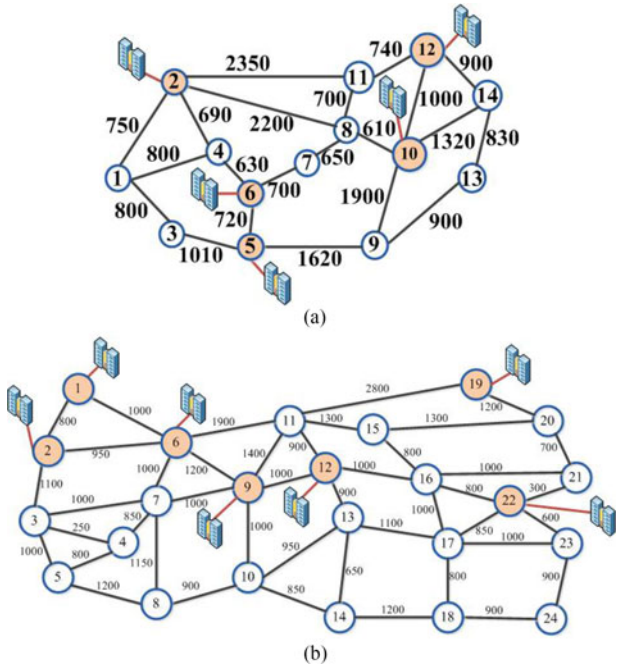

Fig. 4. Network topologies. (a) NSFNET. (b) USNET.

Where B_{grid}^{BPSK} represents the transmission capacity that an FS with a bandwidth of f_{grid} can provide with BPSK, and m is the modulation-level, which is 1, 2, 3, and 4, standing for BPSK, QPSK, 8-QAM and 16-QAM, respectively. In the experiments, we set $f_{grid} = 12.5$ GHZ and $B_{grid}^{BPSK} = 12.5$ Gb/s. The maximum transmission reaches of BPSK, QPSK, 8-QAM and 16-QAM signals are 5000, 2500, 1250, and 625 km, respectively. The power consumption values for each element and application server in the network are those listed in Table II [24, 25]. The dynamic requests' arrivals follow the Poisson process with the average arrival rate as λ, and the holding time of each request follows the negative exponential distribution with the mean value $1/\mu$. Hence, we can quantify the traffic load as λ/μ in Erlangs [26], which varies within [50–500] Erlang by setting $\lambda = 10$ and changing $1/\mu$ from 5 up to 50.

In the experiments, we compare three algorithms:

1) The energy-efficient manycast (EEM) algorithm of [11]. The EEM algorithm considers all the network elements contributing significantly to power consumption and turns them off during idle periods.
2) The dynamic scheduling and distance-adaptive transmission (DS+DAT) algorithm proposed in [12]. The DS+DAT algorithm employs a power saving strategy that dynamically adjusts the transponder capacity and the distance adaptive transmission.
3) Our proposed PALM algorithm described in the previous section. We use four different values for the threshold M that denotes the utilization of the lightpath, $M = 60, 70, 80, 90\%$, hence we refer to the various variants of PALM as PALM-60, PALM-70, PALM-80, and PALM-90, respectively. Note that the smaller the value of M, the greater the number of attempts of rerouting the lightpath. All the data for the IPSO-BPNN prediction algorithm are taken from the inter-DC network traffic data of X

TABLE II
TYPICAL POWER CONSUMPTION OF NETWORK ELEMENTS AND APPLICATION SERVERS

Network element	Fixed power consumption (W)	Traffic-dependent power consumption (W/Gb)
EDFA	110	0
BV-OT	120	0.18
OF-R	1329	0.47
OF-OXC	$150 + 85d + 50g$ (d: node degree, g: number of add/drop capable ports)	0
DC-S	180	0

company in China. Also, the parameters of the IPSO-BPNN are set as follows. The BP neural network is a three-layer structure of $n = 6$ input layer nodes, $m = 13$ hidden layer nodes, and one output node; the training time is 100; the training target is 0.00001; and the learning rate is 0.01. The IPSO algorithm parameters are set to [27]: the initialized species scale is 30; the evolution algebra is 100 times; the accelerating constants are $c_1 = c_2 = 1.49445$; and the particle position and velocity value intervals are $[-5, 5]$ and $[-1, 1]$, respectively.

We compare the various algorithms with respect to four metrics:

- Total power consumption (TPC) across the EON and DCs.
- The number of new lightpaths established (NNLE).
- The percentage of power saving (PPS) of the PALM variants relative to EEM and DS+DAT.
- The bandwidth blocking ratio (BBR), i.e., the percentage of the amount of blocked traffic in relation to the total bandwidth requested.
- The Resource Utilization Rate (RUR), i.e., the resources being used in relation to the total allocated resource in network.

B. Simulation Results

Let us first consider the total power consumption (TPC) metric. Fig. 5 plots the average TPC achieved by the various algorithms listed above, as a function of the traffic load. As expected, power consumption increases with the amount of traffic carried by the network. Importantly, we observe that the PALM algorithm achieves lower consumption compared to the DS+DAT and EEM algorithms for the same traffic volume in both NSFNET and USNET. Although PALM keeps idle lightpaths on (i.e., even when they do not carry traffic), the power savings from not having to tear lightpaths down only to activate them again a short time later when a new traffic request arrives, more than compensates for the power needed to maintain the idle lightpaths. The result indicates that a reduction in the frequency of setting up lightpaths indeed leads to lower TPC for the whole network. We also note that the network expends more power under PALM variants with higher threshold M which denotes the utilization of the lightpath. Recall that smaller values of M imply higher probability of rerouting, and hence a higher probability of keeping an idle lightpath on, which in turn decreases the switching power of the network.

Fig. 6 provides a different perspective of the power savings possible by deploying our PALM algorithm. Specifically, the figure plots the average percent power savings (PPS) achieved by the PALM-80 algorithm (i.e., PALM with threshold $M = 80\%$),

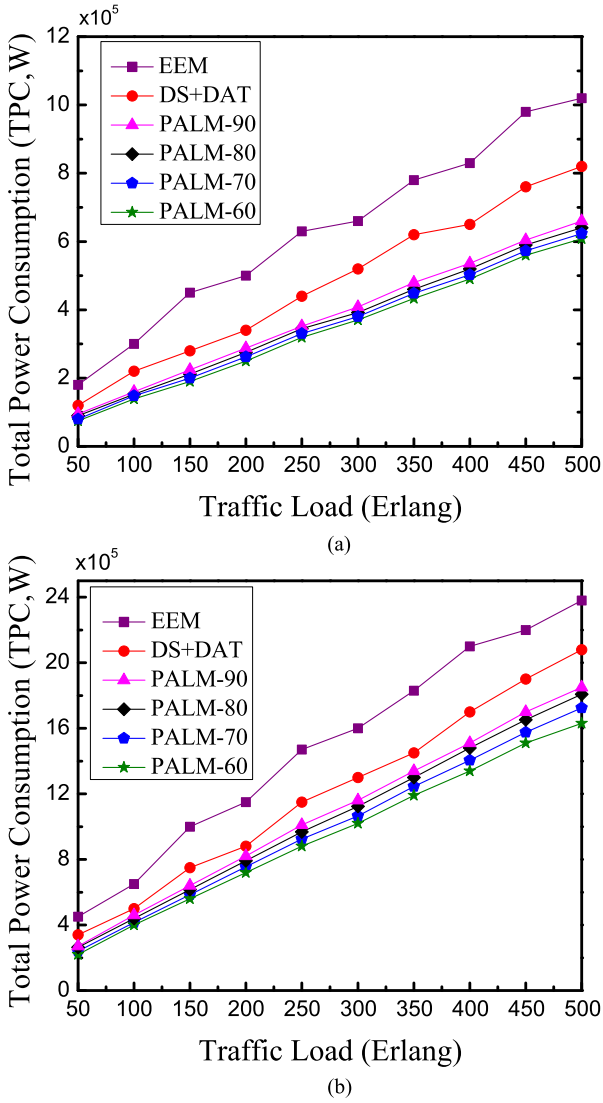

Fig. 5. Results on Total Power Consumption (TPC, normalized). (a) NSFNET. (b) USNET.

relative to the EEM and DS+DAT algorithms. As we can see, in NSFNET, PALM reduces power consumption between 40–47% compared to EEM, and between 16–24% compared to DS+DAT. This benefit can be 30–37% and 13–19% in USNET. We conclude that the combination of PALM and SDN contributes significantly to power savings. We discussed the benefits of PALM above. We also note that, as pointed out in [28], the average lightpath setup time in SDN (with OpenFlow control) is lower than in traditional distributed control (e.g., GMPLS). The main reason is that in SDN, the controllers are able to carry out path

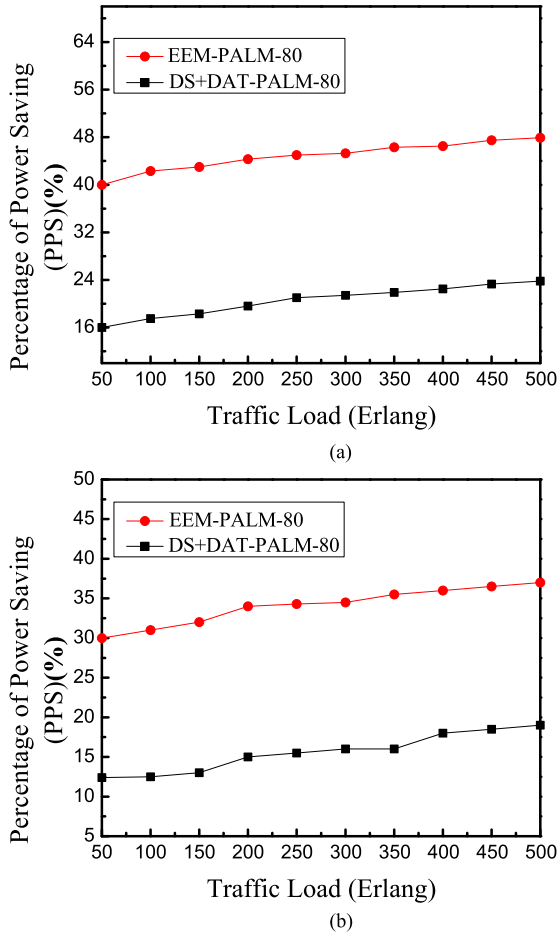

Fig. 6. Results on Percentage of Power Saving (PPS). (a) NSFNET. (b) USNET.

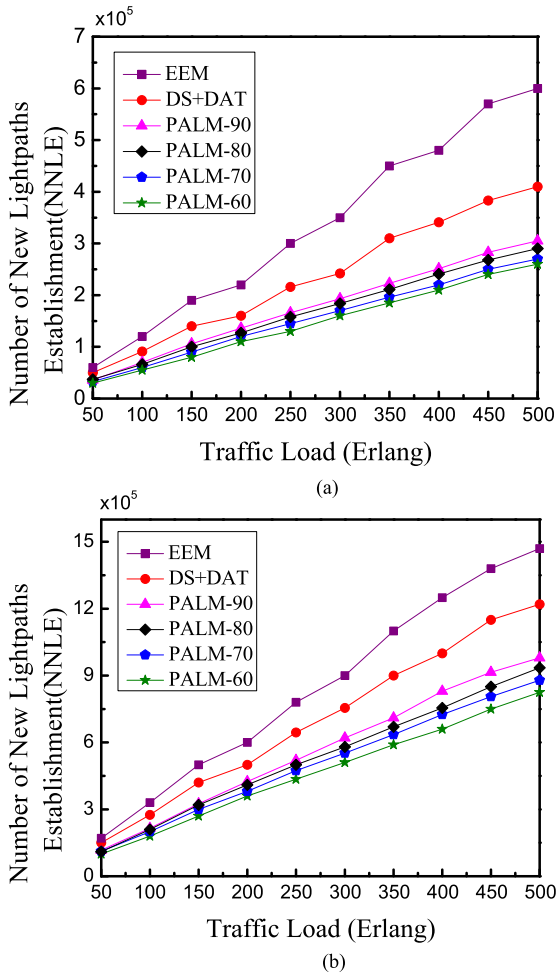

Fig. 7. Results on Number of New Lightpaths Establishment (NNLE). (a) NSFNET. (b) USNET.

computation, resource assignment, and reservation in parallel [28]. Since DS+DAT uses an SDN control plane, the reduction shown in Fig. 6 is solely due to PALM. But since EEM uses GMPLS, the corresponding reduction is due to both PALM and SDN. In other words, PALM leads to a considerable power savings, and SDN makes this benefit more prominent. Moreover, PPS increases with traffic load: as expected, the higher the traffic load, the more opportunities to hold an idle lightpath on, as it is likely that a new traffic request may arrive soon. Nevertheless, the PALM algorithm is successful in providing meaningful power savings across the traffic loads that were considered in our experiments. The PPS values for other PALM variants are similar, as can be deduced from Fig. 5.

Fig. 7 plots the average number of new lightpaths established (NNLE metric) for the various algorithms considered in our study. Since the PALM algorithm is explicitly designed to avoid the tearing down (and, hence, later activation) of lightpaths, it is no surprise that the PALM variants lead to a reduction in the NNLE at the same traffic load in both NSFNET and USNET, compared to the EEM and DS+DAT algorithms. Establishing a lightpath contributes directly to power consumption in the network, mainly through the activation of BV-OTs. Therefore, this decrease in NNLE is a major factor that the PALM variants achieve the power savings illustrated in the previous figures. The relative performance among the PALM variants is similar to the one observed in Fig. 5, and may explained using similar arguments.

Fig. 8 plots the bandwidth blocking ratio (BBR) for the algorithms we considered in this study. It can be seen that the BBR of the proposed PALM algorithm is slightly larger than that of the EEM and DS+DAT algorithms under the same traffic load. For the EEM and DS+DAT, the rerouting or blocking are performed in case the bandwidth of a lightpath is fully used, while PALM takes these actions when the bandwidth of a lightpath exceeds the predefined threshold M. Therefore, PALM may slightly increase the probability of bandwidth blocking, and consequently, the BBR decreases with the value of M, as shown in the figure. For instance, when $M = 80$, the increase in BBR compared to EEM and DS+DAT is about 5% in NSFNET, and this value is about 3% in USNET. Considering the significant power savings, we believe that this is a cost-effective tradeoff.

Finally, Fig. 9 plots the resource utilization rate (RUR) for the various algorithms considered in our study. From the definition of RUR, we can see that if all the resources in the network are allocated on demand and immediately (a symbol of high flexibility), the RUR must be high, that is, it will lead to a high

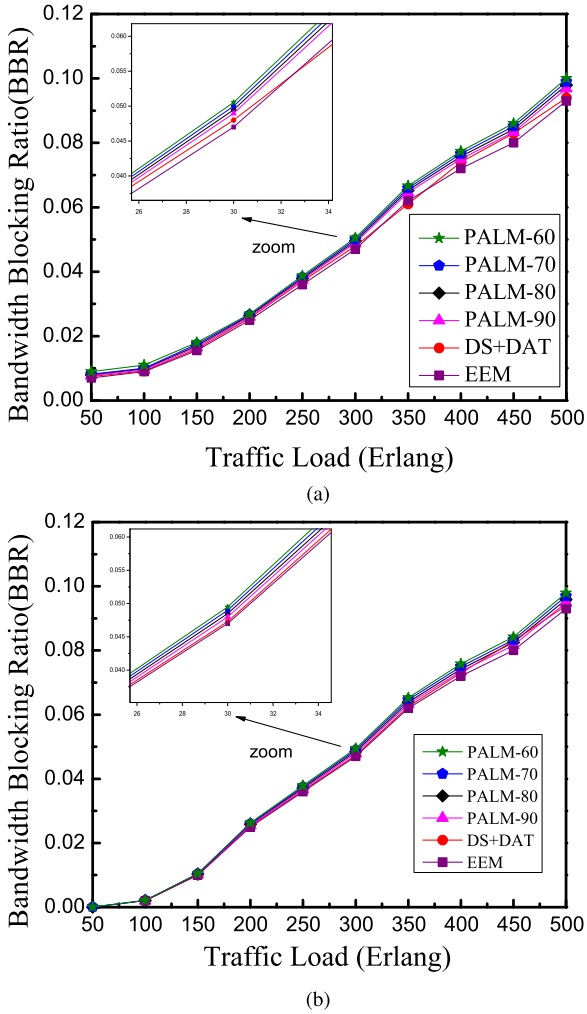

Fig. 8. Results on Bandwidth Blocking Ratio (BBR). (a) NSFNET. (b) USNET.

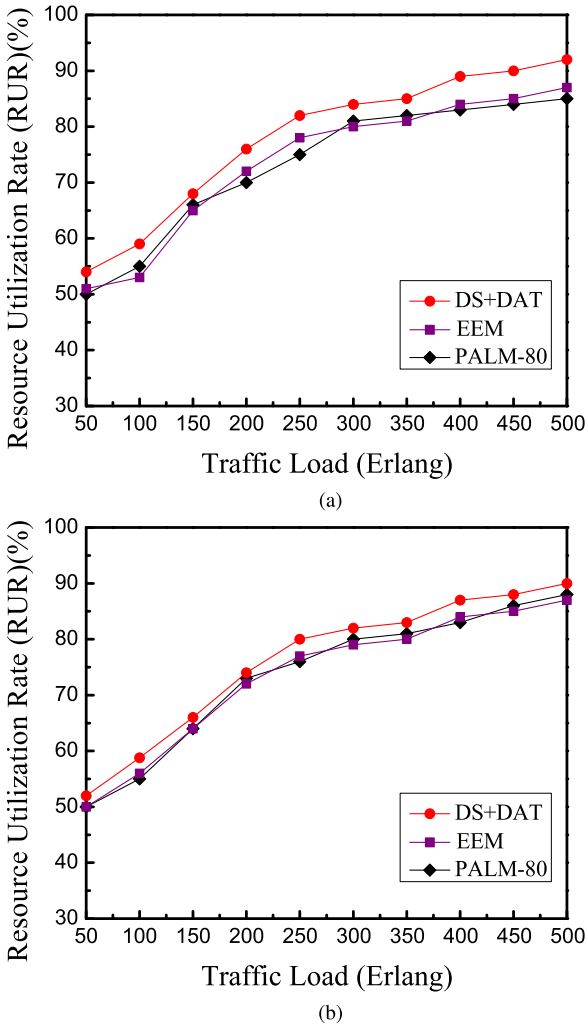

Fig. 9. Results on Resource Utilization Rate (RUR). (a) NSFNET. (b) USNET.

network flexibility. The RUR of DS+DAT is highest among all algorithms which we can see from Fig. 9. This is because it intelligently applied SDN technology that the resources are allocated on demand and immediately. The RUR of the proposed algorithm (PALM) is lower, because the purpose of PALM is to hold the lightpaths that will be reused in the near future, therefore, there are some idle lightpaths in network, that is, there are some allocated but unused resources in the network. On the other hand, recall that DS+DAT employs 4QAM, a fixed and lower order modulation format, whereas PALM employs an adaptive modulation strategy. Since 4QAM uses more spectrum resources than the adaptive strategy, the higher RUR values of DS+DAT do not necessarily reflect more effective use of resources but also account for the higher resources that DS+DAT allocates to lightpaths. Moreover, the RUR of PALM which applied the SDN technology is comparable to the EEM that does not employ SDN. From the two topologies, we can see that the RUR of proposed PALM is 5%−7% lower than that of DS+DAT. Considering the substantial power savings and the impact of 4QAM in the higher RUR values of DS+DAT, we believe that PALM represents a worthy tradeoff.

V. Concluding Remarks

With this study, we demonstrate the benefits of managing the lifetime of lightpaths in an EON interconnecting DCs, so as to improve the energy efficiency of the network. Specifically, we introduced a parameterized algorithm, referred to as PALM, which uses traffic prediction to avoid tearing down a lightpath that becomes idle with the goal of decreasing the switching power involved in setting up the lightpath again a short time later. We also introduced an IPSO-BPNN model to aid the PALM algorithm in accurately predicting future traffic demands. The PALM algorithm leads to lower frequency of new lightpath establishment in the network, which in turn reduces power consumption significantly compared to algorithms that do not manage the lightpath lifetimes.

References

[1] S. Talebi, F. Alam, I. Katib, M. Khamis, R. Khalifah, and G. N. Rouskas, "Spectrum management techniques for elastic optical networks: A survey," *Opt. Switching Netw.*, vol. 13, pp. 34–48, Jul. 2014.

[2] M. Dallaglio, A. Giorgetti, N. Sambo, F. Cugini, and P. Castoldi, "Impact of slice-ability on dynamic restoration in GMPLS-based flexible optical networks," in *Proc. Opt. Fiber Commun. Conf.*, 2014.

[3] Y. Yin *et al.*, "Spectral and spatial 2D fragmentation C aware routing and spectrum assignment algorithms in elastic optical networks [Invited]," *J. Opt. Commun. Netw.*, vol. 5, no. 10, pp. A100–A106, 2013.
[4] M. Channegowda, R. Nejabati, and D. Simeonidou, "Software-defined optical networks technology and infrastructure: Enabling software-defined optical network operations [Invited]," *J. Opt. Commun. Netw.*, vol. 5, no. 10, pp. A274–A282, 2013.
[5] J. Zhang *et al.*, "First demonstration of enhanced software defined networking (eSDN) over elastic grid (eGrid) optical networks for data center service migration," in *Proc. Nat. Fiber Optic Eng. Conf.*, 2013.
[6] D. Abts, M. Marty, P. Wells, P. Klausler, and H. Liu, "Energy proportional datacenter networks," in *Proc. ACM SIGARCH Comput. Arch. News*, 2010.
[7] P. Lu, L. Zhang, X. Liu, J. Yao, and Z. Zhu, "Highly efficient data migration and backup for big data applications in elastic optical inter-data-center networks," *IEEE Network*, vol. 29, no. 5, pp. 36–42, Sep./Oct. 2015.
[8] B. Raffaele, B. Roberto, D. Franco, D. G. Lorenzo, and D. Pasquale, "The green abstraction layer: A standard power-management interface for next-generation network devices," *IEEE Internet Comput.*, vol. 17, no. 2, pp. 82–86, Mar./Apr. 2013.
[9] R. Bolla, R. Bruschi, F. Davoli, and C. Lombardo, "Fine-Grained energy-efficient consolidation in SDN networks and devices," *J. Netw. Serv. Manage.*, vol. 12, no. 2, pp. 132–145, 2015.
[10] J. Zhang *et al.*, "Dynamic traffic grooming in sliceable bandwidth-variable transponder-enabled elastic optical networks," *J. Lightw. Technol.*, vol. 33, no. 1, pp. 183–191, Jan. 2015.
[11] A. Fallahpour, H. Beyranvand, and J. A. Salehi, "Energy-efficient manycast routing and spectrum assignment in elastic optical networks for cloud computing environment," *J. Lightw. Technol.*, vol. 33, no. 19, pp. 4008–4018, 2015.
[12] H. Khodakarami, B. Sankar G. Pillai, and W. Shieh, "Quality of service provisioning and energy minimized scheduling in software defined flexible optical networks," *J. Opt. Commun. Netw.*, vol. 8, no. 2, pp. 118–128, 2016.
[13] R. Bolla, R. Bruschi, and P. Lago, "The hidden cost of network low power idle," in *Proc. IEEE Int. Conf. Commun.*, 2013.
[14] M. Dallaglio, N. Sambo, J. Akhtar, F. Cugini, and P. Castoldi, "YANG model and NETCONF protocol for control and management of elastic optical networks," in *Proc. Opt. Fiber Commun. Conf. Exhibition*. 2016.
[15] E. Yetginer and G. N. Rouskas, "Power efficient traffic grooming in optical WDM networks," in *Proc. GLOBECOM*. 2009.
[16] C.-F. Tsai and J.-W. Wu, "Using neural network ensembles for bankruptcy prediction and credit scoring," *J. Expert Syst. Appl.*, vol. 34, no. 4, pp. 2639–2649, 2008.
[17] Eberhart and Y. Shi, "Particle swarm optimization: Developments, applications and resources," in *Proc. Evol. Comput.*, 2001, pp. 81–86.
[18] X. Chen *et al.*, "Flexible availability-aware differentiated protection in software-defined elastic optical networks," *J. Lightw. Technol.*, vol. 33, no. 18, pp. 3872–3882, Sep. 2015.
[19] Y. Xiong, X. Fan, and S. Liu, "Fairness enhanced dynamic routing and spectrum allocation in elastic optical networks," *J. IET Commun.*, vol. 10, no. 9, pp. 1012–1020, 2016.
[20] N. Charbonneau and V. M. Vokkarane, "A survey of advance reservation routing and wavelength assignment in wavelength-routed WDM networks," in *IEEE Commun. Surveys Tutorials*, vol. 14, no. 4, pp. 1037–1064, Oct.–Dec. 2012.
[21] W. Lu, Z. Zhu, and B. Mukherjee, "On hybrid IR and AR service provisioning in elastic optical networks," *J. Lightw. Technol.*, vol. 33, no. 22, pp. 4659–4670, Nov. 2015.
[22] Y. Wu, M. Tornatore, S. Ferdousi, and B. Mukherjee, "Green data center placement in optical cloud networks," *IEEE Trans. Green Commun. Netw.*, vol. 1, no. 3, pp. 347–357, Sep. 2017.
[23] C. Chen *et al.*, "Demonstrations of efficient online spectrum defragmentation in software-defined elastic optical networks," *J. Lightw. Technol.*, vol. 32, no. 24, pp. 4701–4711, Dec. 2014.
[24] S. Zhang and B. Mukherjee, "Energy-efficient dynamic provisioning for spectrum elastic optical networks," in *Proc. IEEE Int. Conf. Commun.*, 2012, pp. 3031–3035.
[25] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *J. ACMSIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, 2008.
[26] X. Liu, L. Zhang, M. Zhang, and Z. Zhu, "Joint defragmentation of spectrum and computing resources in inter-datacenter networks over elastic optical infrastructure," in *Proc. IEEE Int. Conf. Commun.*, 2014, pp. 3289–3294.
[27] S. Li, L. Wang, and B. Liu, "Prediction of short-term traffic flow based on PSO-optimized chaotic BP neural network," in *Proc. Comput. Sci. Appl.*, 2013, pp. 292–295.
[28] Y. Zhao, J. Zhang, H. Yang, and Y. Yu, "Which is more suitable for the control over large scale optical networks, GMPLS or OpenFlow?," in *Proc. Opt. Fiber Commun. Conf. Expo. Nat. Fiber Opt. Eng. Conf.*, 2013.

Authors' biographies not available at the time of publication.