Power-Aware Lightpath Management for SDN-Based Elastic Optical Networks

Yu Xiong[*†], Jin Shi[*], Yi Lv[*], George N. Rouskas[†]
[*]Key Laboratory of Optical Communication and Networks,
Chongqing University of Posts and Telecommunications, Chongqing, China
[†]Department of Computer Science, North Carolina State University, Raleigh, NC, USA

***Abstract*—Elastic optical networks (EONs) are considered as the most promising technology for interconnecting data centers. With the rapid growth of inter-datacenter traffic, power consumption of EONs becomes a significant challenge. In this work, we present a lightpath management algorithm that uses traffic prediction techniques to eliminate unnecessary lightpath termination and re-establishment so as to decrease switching power and enhance the energy efficiency of the network. Our algorithm builds upon the centralized control and capabilities of software defined networking (SDN) technology. Numerical results show 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.**

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 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 is 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 particle swarm optimization (PSO). 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.

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 Figure 1. The architecture includes OpenFlow-enabled IP Routers and BV-OXCs, which we refer to as OF-R and OF-OXC, respectively. 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. Specifically, the SDN controller communicates with EON and DC elements via the OpenFlow protocol, and hence, it may quickly and accurately converge the network state information, as well as respond to lightpath provisioning requests in a timely manner according to the obtained network resource information.

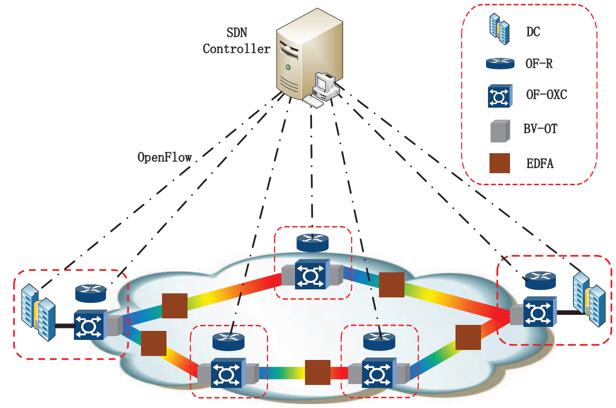

Fig. 1. Software defined EONs for DC application

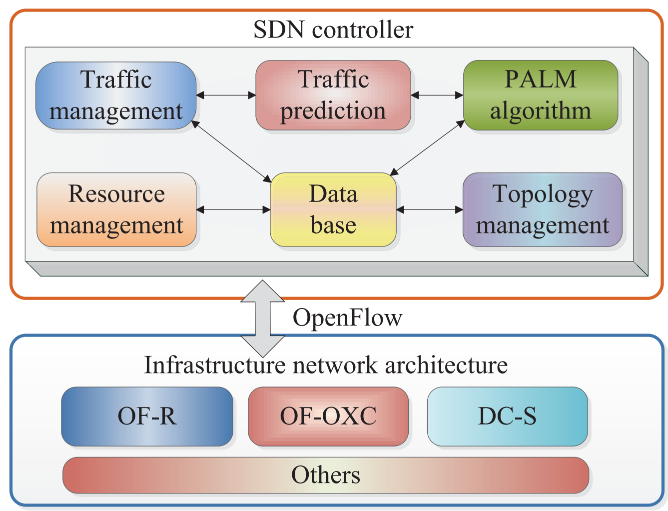

Fig. 2. The functional modules of controller in software defined EONs for DC application

A. Functional Models of SDN Controller

In order to achieve our power saving strategy for the whole network, the functionality of the SDN controller must be extended as shown in Figure 2. Specifically, the SDN controller consists of six modules, namely, resource management, database, topology management, traffic management, traffic prediction and our proposed power-aware lightpath management (PALM) algorithm. The basic responsibilities and interactions among the functional modules are as follows. The resource management module can interact with the OF-R module and the DC-S module to collect network resource information of the underlying EONs interconnecting the DCs and then abstract them into a unified resource. The database contains real-time information on the resources of the underlying network, including optical nodes resources, lightpath resources and application resources of DC-S, which can obtained from the resource management module. Topology management is mainly used for the generation of network topology, which needs to learn the entire network resource information from the database module. The traffic management module is concerned with collecting and storing traffic information of optical nodes and lightpaths maintained in the database module, and with monitoring the traffic status of the whole network in real time. Historical traffic information stored in the traffic management module is delivered to the traffic prediction module to estimate demand for lightpaths in the near future. The PALM algorithm module is executed based on the result of the traffic prediction and current resource information stored in the traffic prediction and database modules, and generates lightpath provisioning decisions.

B. Power Consumption Model

Returning to Figure 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 [14], turning on network elements 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 Figure 1 may be calculated as:

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

Typical power consumption values for the network elements and DC-S are listed in Table II, and have been taken from the studies in [15], [16]. For simplicity, in this work we assume that the conventional IP Router and BV-OXC consume the same power as the OF-R and OF-OXC, respectively.

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_T^d	Total dynamic power consumption of BV-OTs
P_R^d	Total dynamic power consumption of OF-Rs
W_{ij}	Number of EDFAs on lightpath ij
O_{ij}	Number of BV-OT pairs on lightpath ij
Q_{ij}	Number of OF-Rs on lightpath ij
Ω_n	Binary, equal to 1 if traffic load arrive at node n
K_s	Binary, equal to 1 if DC-S stays working
M_n	Set of add/drop ports of OF-OXC in node n
D_n	Set of the degree of node n
R_{sd}	Traffic demand (Gb/s) of node-pair sd
X_{ij}^{sd}	Binary, equals 1 if traffic between node-pair sd uses lightpath ij
Y_{ab}^{ij}	Binary, equals 1 if lightpath ij uses BV-OTs a, b at nodes i, j
$p_{e,ab}^{ij}$	Power consumption for BV-OT ab to set up lightpath ij
p_f^{ij}	Holding power of lightpath ij
p_d^{ij}	Transmission power for traffic load transverse lightpath ij

Now note that the infrastructure network consists mainly of EDFA, BV-OT, OF-R, OF-OXC and DC-S. According to Table II, all these elements have fixed power consumption during network operation. Therefore, the fixed component P_f may be obtained as follows:

$$\begin{aligned} P_f &= P_E^f + P_T^f + P_R^f + P_O^f + P_S^f \\ &= \sum_{i \in N} \sum_{j \in N: j \neq i} 110 \times W_{ij} + \sum_{i \in N} \sum_{j \in N: j \neq i} 120 \times O_{ij} \\ &+ \sum_{i \in N} \sum_{j \in N: j \neq i} 1329 \times Q_{ij} \\ &+ \sum_{n \in N} \left(150 + \sum_{d \in D_n} 85 \times d + \sum_{g \in M_n} 50 \times g \right) \times \Omega_n \\ &+ \sum_{s \in S} 180 \times K_s \end{aligned} \tag{2}$$

On the other hand, per Table II, the EDFA, OF-OXC and DC-S have no dynamic power consumption. Thus, the dynamic power of the whole network is:

$$\begin{aligned} P_d &= P_T^d + P_R^d \\ &= \sum_{i \in N} \sum_{j \in N: j \neq i} \left(0.18 \times \sum_{sd \in r} R_{sd} \times X_{ij}^{sd} \right) \\ &+ \sum_{i \in N} \sum_{j \in N: j \neq i} \sum_{a \in Z} \sum_{b \in Z} \left(0.47 \times \sum_{sd \in r} R_{sd} \times Y_{ab}^{ij} \right) \end{aligned} \tag{3}$$

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

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 and OF-R 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:

$$p_f^{ij} = 110 \times W_{ij} + 120 \times O_{ij} + 1329 \times Q_{ij} \tag{6}$$

$$p_d^{ij} = 0.18 \times \sum_{sd \in r} R_{sd} X_{ij}^{sd} + \sum_{a \in Z} \sum_{b \in Z} \left(0.47 \times \sum_{sd \in r} R_{sd} Y_{ab}^{ij} \right) \tag{7}$$

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:

$$\begin{aligned} p_{e,ab}^{ij} &= 4p^{ij} = 4(p_f^{ij} + p_d^{ij}) \\ &= 4\,(110 \times W_{ij} + 120 \times O_{ij} + 1329 \times Q_{ij} \\ &+ 0.18 \times \sum_{sd \in r} R_{sd} X_{ij}^{sd} \\ &+ \sum_{a \in Z} \sum_{b \in Z} \left(0.47 \times \sum_{sd \in r} R_{sd} Y_{ab}^{ij} \right) \Bigg) \end{aligned} \tag{8}$$

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.

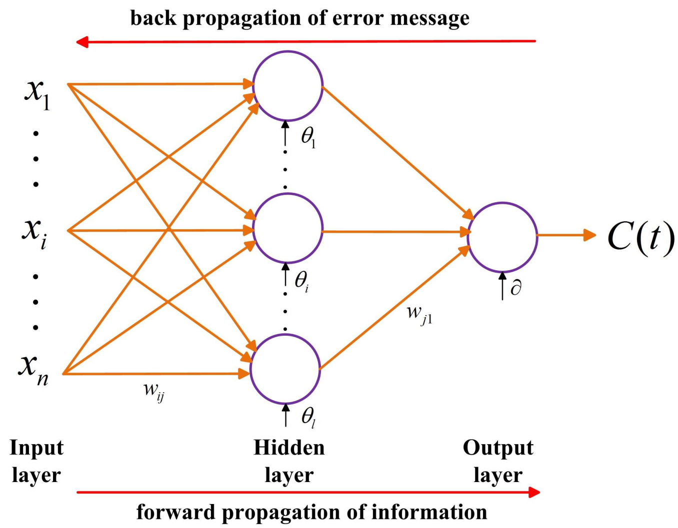

Fig. 3. Configuration of a 3-layer BP network model

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 $C(t)$ denote the traffic load of some lightpath 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 [17], on the other hand, have excellent non-linear 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 Figure 3. 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 [17]. In this paper, we use the three-layer BP network model depicted in Figure 3

The BP neural network of Figure 3 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:

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

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 outputs of network may be obtained as:

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

where ∂ is the threshold of the output layer node.

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

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

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

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

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

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

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 exploit a particle swarm optimization (PSO) algorithm in [18] to improve the BP neural network model. The excellent global optimization ability of the PSO algorithm make 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 [18]. As time passes, each particle updates its position and velocity to a new value according to expressions (14) and (15).

$$\begin{aligned} v_i(t+1) &= \omega\ v_i(t) + c_1\ rand(0,1)(p_i - x_i(t)) \\ &+ c_2\ rand(0,1)(p_g - x_i(t)) \end{aligned} \tag{14}$$

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

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

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{T_{\max}}\ k \tag{16}$$

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.

We use the PSO algorithm 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.

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. More specifically, once the SDN controller detects an idle lightpath (the lightpath to be removed) from the database module, it uses the prediction algorithm based on PSO-BP that we discussed in the previous subsection, to determine a *holding time* t_h 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 [19], the authors have estimated that the control plane latency for setting up a lightpath is around 23ms, 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_h of an idle lightpath. To this end, we use the prediction algorithm based on PSO-BP, 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 $t_h, t_h \leq t_{max}$, if the prediction results indicate one of two possibilities:

- *Case 1:* A traffic demand that will request the idle lightpath is expected to arrive at time t_h 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_h. In this case, we reroute the new traffic request to the idle lightpath 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 l_i results in power consumption no greater than transmission on its original lightpath L_{nx}: $P_{l_i}(C(t)) \leq P_{L_{nx}}(C(t))$. Finally, we note that the lightpath spectrum must satisfy the spectrum continuity and spectrum contiguity constraints when the routing is carried out [1], [20].

A pseudocode description of the PALM algorithm is provided as Algorithm 1. The traffic prediction module firstly run the prediction algorithm based on the PSO-BP neural network to predict the traffic load for every lightpath within the next $t_{\max}$ time units, and the value $C(t)$ 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, the t_h for the l_i is initialized to zero. And then the traffic management module searches all other lightpaths L_n between (s, d) (Lines 4-6). Then the PALM module checks every lightpath's traffic load at time $t(C(t))$ stored in the traffic prediction module. If it finds a traffic request will arrive to the idle lightpath l_i, i.e., there is a traffic load $C_{l_i}(t)$ to be carried on lightpath l_i, the t will be this idle lightpath's holding time t_h (Lines 7-8).Then it break the loop to returns the idle lightpath's holding time t_h and the power consumption of whole network based on expression (1) (Line 23). Otherwise, if the newly arriving traffic $C_{L_{nx}}(t)$ to one of the lightpaths $L_n(i.e., L_{nx})$ at time t leads to a total traffic load about L_{nx} beyond the pre-established threshold value, M, and the power consumption of the traffic $C_{L_{nx}}(t)$ on the idle lightpath l_i is lower or equal than the power of the traffic $C_{L_{nx}}(t)$ on its original lightpath L_{nx}, then $C_{L_{nx}}(t)$ is rerouted to the idle lightpath l_i, the holding time t_h is changed to t, and the load on l_i as well as the network topology G_p are updated (Lines 11-16). Note that if the sum of bandwidth of the rerouted traffic achieves the highest capacity of the idle lightpath, the rerouting process is terminated. The idle lightpath l_i will be torn down if the power consumption of the traffic $C_{L_{nx}}(t)$ on the idle lightpath l_i is higher than the power of the traffic $C_{L_{nx}}(t)$ on its original lightpath L_{nx} at time t, after which G_p is updated (Lines 17-20). Then the algorithm returns the holding time and the power consumption of whole network based on expression (1) (Line 23).

Algorithm 1 Power-Aware Lightpath Management (PALM)

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

1: **for** $\forall C$ **do**
2: $\quad C(t) \leftarrow PSO \cdot BP(s, t \leq t_{\max}, d)$;
3: **end for**
4: **for** $\forall\, l_i$ **do**
5: $\quad t_h \leftarrow 0$;
6: $\quad L_n \leftarrow (s, d)$;
7: $\quad$ **if** $C_{l_i}(t) > 0$ **then**
8: $\quad\quad t_h \leftarrow t$;
9: $\quad\quad break$;
10: $\quad$ **end if**
11: $\quad$ **if** $C_{L_{nx}} + C_{L_{nx}}(t) \geq M\%$ **then**
12: $\quad\quad$ **if** $P_{l_i}(C_{L_{nx}}(t)) \leq P_{L_{nx}}(C_{L_{nx}}(t))$ **then**
13: $\quad\quad\quad l_i \leftarrow C_{L_{nx}}(t)$;
14: $\quad\quad\quad t_h \leftarrow t$;
15: $\quad\quad\quad C_{l_i} \leftarrow C_{l_i} + C_{L_{nx}}(t)$;
16: $\quad\quad\quad$ Update G_p;
17: $\quad\quad$ **else**
18: $\quad\quad\quad G_p - l_i$;
19: $\quad\quad\quad$ Update G_p;
20: $\quad\quad$ **end if**
21: $\quad$ **end if**
22: **end for**
23: $P = P_f + P_d + P_e,\ t_h$;

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 24-node, 43-link USNET network topology shown in Figure 4, with DCs at nodes 6, 8, 9, 15, 18, and 22 [7]. We assume that

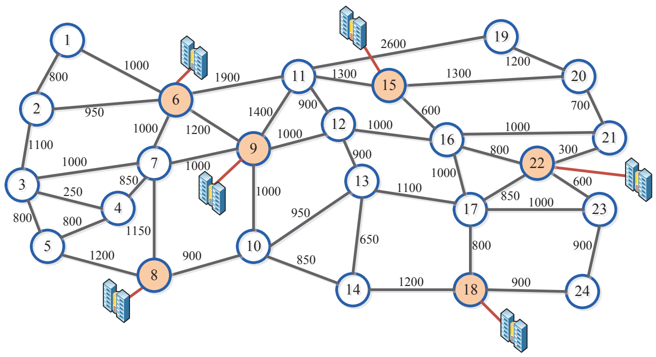

Fig. 4. USNET topology

there is one pair of bi-directional fiber on each link, and the available spectrum width of each fiber is set to 4000GHz with a slot width of 12.5GHz. In our simulation, we set the capacity of each transponder to 100Gb/s, the modulation format is 4-QAM, and the traffic demand is the aggregate monthly traffic from [12]. The power consumption values for each element and application server in the network are those listed in Table II. We further assume that flow requests to DC nodes arrive following a Poisson process, and their bandwidth is randomly and uniformly distributed between 12.5 Gbps to 100 Gbps. The traffic holding time of each request is exponentially distributed with unit (normalized) mean [21].

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 PSO-BP prediction algorithm are taken from the web site http://noc.net.internet2.edu/. Also, the parameters of the PSO-BP model 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 PSO algorithm parameters are set to [22]: the 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.

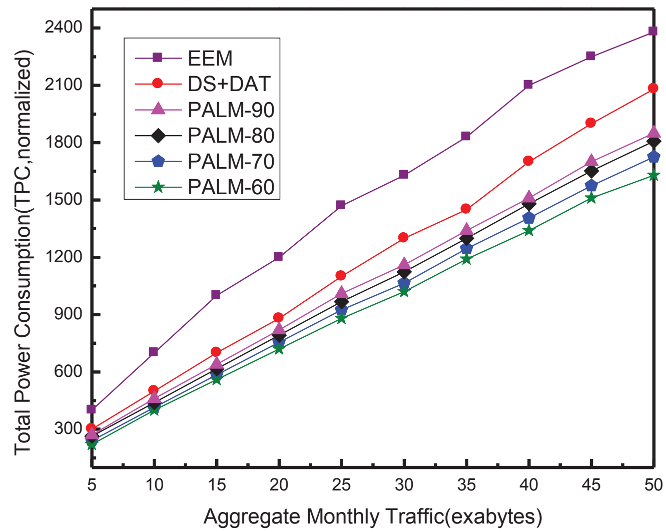

Fig. 5. Algorithm comparison with respect to TPC (normalized)

B. Simulation Results

Let us first consider the total power consumption (TPC) metric. Figure 5 plots the average TPC achieved by the various algorithms listed above, as a function of the monthly aggregate traffic load. As expected, power consumption increases with the amount of traffic carried by the network. Importantly, we observe that all variants of the new PALM algorithm achieve lower consumption compared to the EEM and DS+DAT algorithms for the same monthly aggregate traffic volume. Although the PALM variants keep 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. This 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.

Figure 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 $M = 80\%$), relative to the EEM and DS+DAT algorithms.

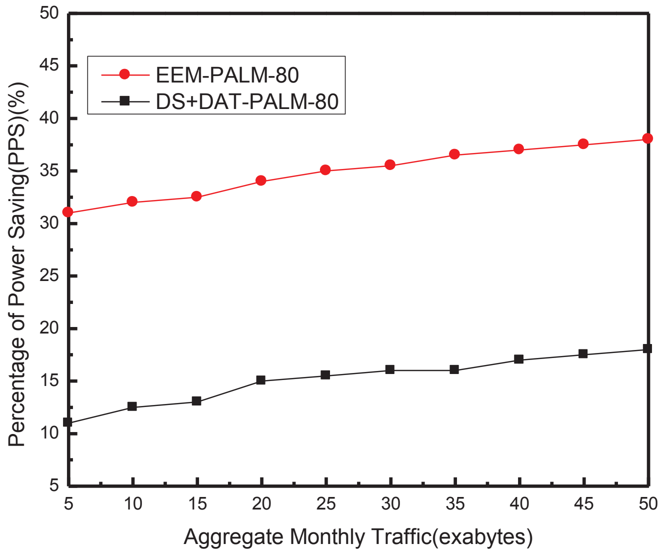

Fig. 6. Algorithm comparison with respect to PPS

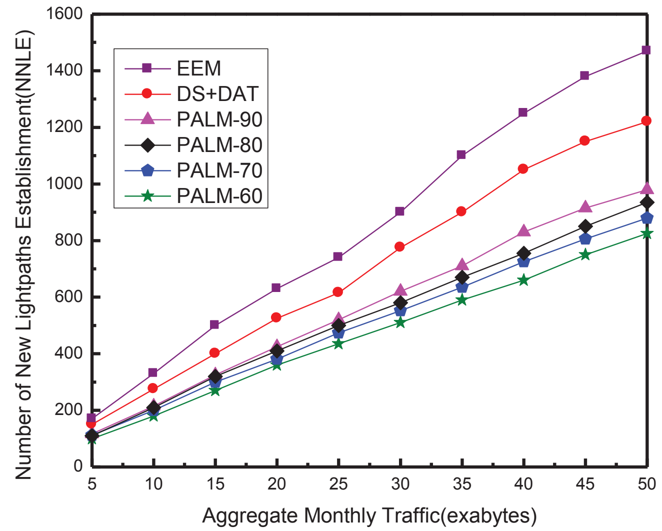

Fig. 7. Algorithm comparison with respect to NNLE

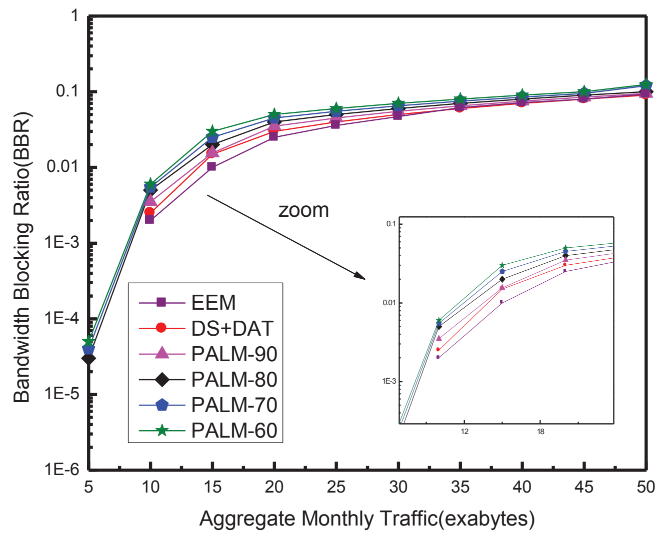

Fig. 8. Algorithm comparison with respect to BBR

As we can see, PALM-80 reduces power consumption between 31-36% compared to EEM, and between 11-16% compared to DS+DAT. Importantly, 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 Figure 5.

Figure 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 all PALM variants lead to a reduction in the NNLE at the same traffic load, 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 two figures. The relative performance among the PALM variants is similar to the one observed in Figure 5, and may explained using similar arguments.

Finally, Figure 8 plots the bandwidth blocking ration (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 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 2% and 4%, respectively. Considering that the power savings are 36% and 16% respectively, we believe that this is a cost-effective 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 a PSO-BP neutral network 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.

Acknowledgment

This work was made possible with funding from: the National Natural Science Foundation of China (61401052); the Science and Technology Project of Chongqing Municipal Education Commission (KJ1400418, KJ1500445); the Starting Foundation for Doctors of Chongqing University of Posts and Telecommunications (A2015-09); and the Program for Innovation Team Building at Institutions of Higher Education in Chongqing (CXTDX201601020).

References

[1] Sahar Talebi, Furqan Alam, I. Katib, M. Khamis, R. Khalifah, George N. Rouskas, "Spectrum Management Techniques for Elastic Optical Networks: A Survey." *Optical Switching and Networking*, vol. 13, pp. 34-48, July 2014.

[2] Matteo Dallaglio, Alessio Giorgetti, Nicola Sambo, Filippo Cugini, Piero Castoldi, "Impact of Slice-ability on Dynamic Restoration in GMPLS-based Flexible Optical Networks," in *Proc. Optical Fiber Communication Conference*, 2014.

[3] Yawei Yin, Huan Zhang, Mingyang Zhang, Ming Xia, Zuqing Zhu, Stefan Dahlfort, S. J. B. Yoo, "Spectral and spatial 2D fragmentation C aware routing and spectrum assignment algorithms in elastic optical networks [Invited]," *J. Opt. Commun. Networking*, vol.5, no.10, pp.A100-A106, 2013.

[4] Mayur Channegowda, Reza Nejabati, Dimitra Simeonidou, "Software-defined optical networks technology and infrastructure: Enabling software-defined optical network operations [Invited]," *J. Optical Communications and Networking*, vol.5, no.10, pp. A274-A282, 2013.

[5] Jie Zhang, Yongli Zhao, Hui Yang, Yuefeng Ji, Hui Li, Yi Lin, Gang Li, Jianrui Han, Young Lee, Teng Ma, "First demonstration of enhanced software defined networking (eSDN) over elastic grid (eGrid) optical networks for data center service migration," in *Proc. National Fiber Optic Engineers Conference*, 2013.

[6] Dennis Abts, Mike Marty, Philip Wells, Peter Klausler, Hong Liu, "Energy proportional datacenter networks," in *Proc. ACM SIGARCH Computer Architecture News*, 2010.

[7] Ping Lu, Liang Zhang, Xiahe Liu, Jingjing Yao, Zuqing Zhu, "Highly efficient data migration and backup for big data applications in elastic optical inter-data-center networks," *J. IEEE Network*, vol.29, no.5, pp. 36-42, 2015.

[8] Bolla Raffaele, Bruschi Roberto, Davoli Franco, Di Gregorio Lorenzo, Donadio Pasquale, "The Green Abstraction Layer: A Standard Power-Management Interface for Next-Generation Network Devices," *J. IEEE Internet Computing*, vol.17, no.2, pp.82-86, 2013.

[9] Raffaele Bolla, Roberto Bruschi, Franco Davoli, Chiara Lombardo, "Fine-Grained Energy-Efficient Consolidation in SDN Networks and Devices," *J. Network and Service Management*, vol.12, no.2, pp. 132-145, 2015.

[10] Jiawei Zhang, Yuefeng Ji, Mei Song, Yongli Zhao, Xiaosong Yu, Jie Zhang, Biswanath Mukherjee, "Dynamic traffic grooming in sliceable bandwidth-variable transponder-enabled elastic optical networks," *J. Lightwave Technology*, vol.33, no.1, pp. 183-191, 2015.

[11] Ahmad Fallahpour, Hamzeh Beyranvand, Jawad A. Salehi, "Energy-efficient manycast routing and spectrum assignment in elastic optical networks for cloud computing environment," *J. Lightwave Technology*, vol.33, no.19, pp. 4008-4018, 2015.

[12] Hamid Khodakarami, Bipin Sankar Gopalakrishna Pillai, William Shieh, "Quality of service provisioning and energy minimized scheduling in software defined flexible optical networks," *J. Optical Communications and Networking*, vol.8, no.2, pp. 118-128, 2016.

[13] Raffaele Bolla, Roberto Bruschi, Paolo Lago, "The hidden cost of network low power idle," in *Proc. IEEE International Conference on Communications (ICC)*, 2013.

[14] Emre Yetginer, George N. Rouskas, "Power Efficient Traffic Grooming in Optical WDM Networks," in *Proc. GLOBECOM*. 2009.

[15] Shuqiang Zhang, Biswanath Mukherjee, "Energy-efficient dynamic provisioning for spectrum elastic optical networks," in *Proc. IEEE International Conference on Communications (ICC)*, 2012.

[16] Albert Greenberg, James Hamilton, David A. Maltz, Parveen Patel, "The cost of a cloud: research problems in data center networks," *J. ACM SIGCOMM computer communication review*, vol.39, no.1, pp. 68-73, 2008.

[17] Chih-Fong Tsai, Jhen-Wei Wu, "Using neural network ensembles for bankruptcy prediction and credit scoring," *J. Expert Systems with Applications*, vol.34, no.4, pp. 2639-2649, 2008.

[18] Eberhart, Yuhui Shi, "Particle swarm optimization: developments, applications and resources," in *Proc. evolutionary computation*, 2001.

[19] Xiaoliang Chen, Massimo Tornatore, Shilin Zhu, Fan Ji, Wenshuang Zhou, Cen Chen, Daoyun Hu, Liu Jiang, Zuqing Zhu, "Flexible Availability-Aware Differentiated Protection in Software-Defined Elastic Optical Networks," *J. Lightwave Technology*, vol.33, no.18, pp. 3872-3882, 2015.

[20] Yu Xiong, Xue Fan, Shuming Liu, "Fairness enhanced dynamic routing and spectrum allocation in elastic optical networks," *J. IET Communications*, vol.10, no.9, pp. 1012-1020, 2016.

[21] Shuqiang Zhang, Charles Martel, Biswanath Mukherjee, "Dynamic traffic grooming in elastic optical networks," *J. IEEE Journal on selected areas in communications*, vol.31, no.1, pp. 4-12, 2013.

[22] Song Li, Liu Wang, Bo Liu, "Prediction of short-term traffic flow based on PSO-optimized chaotic BP neural network," in *Proc. Computer Sciences and Applications (CSA)*, 2013.