On the Cross-Layer Network Planning for Flexible Ethernet over Elastic Optical Networks

Hui Liang, Nelson L. S. da Fonseca, and Zuqing Zhu, *Senior Member, IEEE*

***Abstract*—This paper investigates the cross-layer network planning for flexible Ethernet (FlexE) over elastic optical networks (EONs) (FlexE-over-EONs). Our investigation focuses on the most challenging setting, *i.e.*, the FlexE-over-EONs based on the FlexE-aware architecture, and considers both single-hop and multi-hop scenarios. For the single-hop scenario, we assume that all the client flows are routed over end-to-end lightpaths in the EON. We formulate a mixed integer linear programming (MILP) model for this problem, transform it into a class constrained bin packing problem (CCBP), and leverage the primal-dual interior-point (PDIP) method to propose a polynomial-time approximation algorithm. For the multi-hop scenario, we assume that each client flow can be routed over multiple lightpaths in the EON. We show that after solving the virtual topology design, the cross-layer planning can be transformed into a single-hop scenario. An integer linear programming (ILP) model is formulated to tackle the virtual topology design, and we design a polynomial-time approximation algorithm for it by modifying the well-known branch-and-bond method. To evaluate the performance of our two-step method for the multi-hop scenario, we also propose a heuristic algorithm. Results derived from extensive simulations show that our approximation algorithms are significantly more time-efficient than the ILP/MILP models, and the solutions produced by them have bounded gaps to the optimal one much smaller than those derived by the heuristic.**

***Index Terms*—Flexible Ethernet (FlexE), Elastic optical networks (EONs), Approximation algorithms, Traffic grooming.**

I. Introduction

RECENTLY, the fast development of datacenter (DC) and metro networks happen all over the world, to cope with the raising of 5G, cloud computing, and Big Data analytics [1]–[5]. This has imposed intensive pressure on networking technologies, especially for Ethernet and optical transport network (OTN). We have witnessed promising advances in these two areas over the past decade, to address these challenges. For instance, Flex Ethernet (FlexE) [6] has been published by the Optical Internetworking Forum (OIF), defining new Ethernet connection types to allow DC operators to utilize OTN bandwidth in more flexible manners, as well as to provide interfaces for realizing service isolation and network sharding.

The major advantage of FlexE is that it leverages time-division multiplexing (TDM) to support a variety of media access control (MAC) rates that support the physical channel (PHY) rates of Ethernet [7]. As the most recent implementation agreement, FlexE 2.0 [7] promises to carry the collections of 100 GbE, 200 GbE, and 400 GbE PHYs. The upcoming FlexE 2.1 will add the support for 50 GbE PHYs. With these PHYs, FlexE can support various MAC rates with the operation principle shown in Fig. 1. Specifically, FlexE inserts a shim layer between the MAC and physical layers, which allows the division of the bandwidth resources of a group of PHYs into a series of calendar slots (CS'), as well as the mapping of the data streams with different data-rates generated by FlexE clients into the CS'. In other words, the shim layer schedules data from MAC interfaces with different rates into the CS' (the transmission opportunities in the PHYs based on TDM).

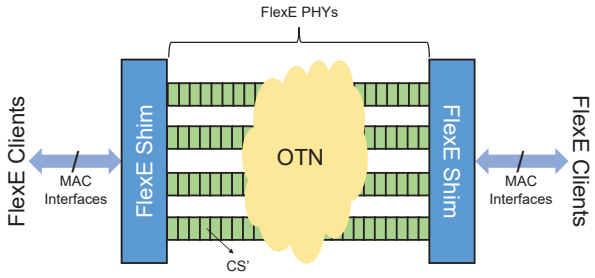


Fig. 1. Generic operation principle of FlexE.

To cope with the diversity of data-rates of MAC interfaces, FlexE utilizes three mechanisms [7]: 1) bonding, *e.g.*, supporting a 200 Gbps MAC interface with two bonded 100 GbE PHYs, 2) sub-rating, *e.g.*, transmitting the data from a 50 Gbps MAC over a 100 GbE PHY, and 3) channelization, *e.g*, supporting a 150 Gbps and two 25 Gbps MAC interfaces with two bonded 100 GbE PHYs. Hence, even though FlexE allocates bandwidth in terms of exclusive time slots, it still has the advantage of high efficiency due to statistical multiplexing.

Meanwhile, for long distance transmissions, the data carried by FlexE PHYs needs to be further fed into the transport boxes (T-Boxes) in an OTN [8] . Therefore, network planning for FlexE-over-OTN is an interesting and challenging problem to investigate, because it actually involves sophisticated cross-layer mapping, *i.e.*, MAC interfaces to FlexE PHYs, and FlexE PHYs to T-Boxes. Depending on the compatibility of the T-Boxes with FlexE, three architectures can be defined for FlexE-over-EONs: FlexE-unaware, FlexE-partially-aware, and FlexE-aware [7]. Only the FlexE-aware architecture uses T-Boxes fully compatible with FlexE. If, on one hand, network planning for the FlexE-aware architecture has the most flexible cross-layer mapping, allowing the highest resource utilization and cost-effectiveness; on the other hand, it is the the most

H. Liang and Z. Zhu are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, P. R. China (email: zqzhu@ieee.org).

N. Fonseca is with the Institute of Computing, State University of Campinas, Campinas, SP 13083-852, Brazil.

Manuscript received on May 5, 2020.

challenging for planning due to its flexibility.

Previously, Eira *et al.* [9] have performed a thoughtful comparative analysis on architecting FlexE-over-OTNs using fixed-grid wavelength-division multiplexing (WDM) networks (*i.e.*, FlexE-over-WDMs) with the three architectures mentioned above. However, they did not consider the flexible-grid elastic optical networks (EONs) [10]–[14]. EONs leverage bandwidth-variable transponders (BV-Ts) and bandwidth-variable wavelength-selective switches (BV-WSS') to realize OTNs with fine spectrum allocation granularities at 12.5 GHz or even narrower. Meanwhile, with sliceable BV-Ts [15]–[17], one can utilize the split-spectrum scheme [18]–[20] to change the data-rates of a T-Box at will. To this end, we expect that an EON-based OTN would be much more friendly toward FlexE.

In [21], we confirmed the benefits of an architecture employing both FlexE and EON. We evaluated the performance of the FlexE-unaware, FlexE-partially-aware, and FlexE-aware architectures. However, the problem of cross-layer network planning for FlexE-over-EONs was not fully explored. The work in [21] was based on the assumption that all traffic flows from MAC interfaces are routed over end-to-end lightpaths in EON; such assumption makes the formulated mixed integer linear programming (MILP) models intractable for large-scale scenarios. Indeed, the formulation for the cross-layer planning problems for FlexE-over-WDMs and FlexE-over-EONs are not fundamentally different. It was shown in [21] that by restricting the values of certain variables to different ranges, the same MILP model can be employed to the modeling of FlexE-over-WDMs and FlexE-over-EONs. Actually, in a FlexE-over-EON, the BV-Ts can take much more line-rates than those in a FlexE-over-WDM. Hence, the solution space of the cross-layer planning for a FlexE-over-EON is much larger, which makes the problem computationally hard.

In this paper, we tackle the problem of cross-layer network planning for FlexE-over-EONs, and focus our problem-solving on the most challenging setting, *i.e.*, the FlexE-over-EONs based on the FlexE-aware architecture. We first consider a simple "single-hop" scenario, where all the traffic flows from MAC interfaces are assumed to be routed over end-to-end lightpaths in the EON. We prove that the cross-layer planning for this single-hop scenario can be transformed into a class constrained bin packing problem (CCBP) [22], and leverage the primal-dual interior-point (PDIP) method [23] to design a polynomial-time approximation algorithm for solving it.

Next, we expand our study to consider a more realistic multi-hop scenario, in which each traffic flow originated from a MAC interface can be routed over multiple lightpaths in the EON [24], [25]. We first formulate an integer linear programming (ILP) model to tackle the virtual topology design, and then we propose a polynomial-time approximation algorithm by modifying the well-known branch-and-bond method [26]. After solving the virtual topology design in the multi-hop scenario, we obtain the hop-by-hop lightpath routing of each traffic flow, then transform the cross-layer planning to that of the single-hop scenario. To assess the performance of our two-step method for the multi-hop scenario, we also propose a heuristic algorithm. Finally, we run extensive simulations to evaluate our proposals. Results confirm that for large-scale cross-layer planning, our approximation algorithms outperform the ILP/MILP models significantly in terms of running time, and the solutions provided by the approximation algorithms are closer to the optimal solution that those given by the proposed heuristic.

The rest of the paper is organized as follows. Section II provides a brief survey on related work. Section III describes the network model. IV and V provides, respectively, the design of the algorithms for the single-hop and multi-hop scenarios. Section VI presents simulation results. Finally, Section VII summarizes the paper.

II. Related Work

Ethernet is a successful technology that has proven its efficiency for interconnecting network elements and carrying IP traffic in access and metro networks. However, when it comes to cover relatively large geographical areas, the joint employment of Ethernet and OTN technologies helps to cope with physical-layer impairments [27] that would be not possible by the single employment of Ethernet. As a consequence of such joint use, Ethernet-over-OTN has become a common practice in metro and core networks. On the other hand, traditional Ethernet interfaces were not developed taking into account the standard data-rates in OTN standard. The introduction of FlexE resolves this mismatch [6], allowing FlexE-over-OTN to deliver improved efficiency and flexibility. Moreover, to accommodate the dynamic traffic from Ethernet, OTN should be able to allocate bandwidth in a sub-wavelength granularity [28]. It is expected that the deployment of FlexE-over-EONs in future Internet will help operators realize effective traffic grooming and scheduling optimization, and thus both the capital expenditure (CAPEX) and the operating expenses (OPEX) can be greatly reduced [9], [21]. Last but not least, the bandwidth allocation mechanism of FlexE makes it easier to slice virtual networks in a FlexE-over-EON [6], while virtual network slicing is an important technique to improve resource utilization and cost-effectiveness in today's core and metro networks [29]–[31].

FlexE-over-EON is essentially a packet-over-EONs network consisting of both packet and optical layers. Therefore, network planning and service provisioning need to consider such multilayer scenario. Specifically, operators need to address at least two tasks: the virtual topology design and traffic grooming. The virtual topology design helps establishing lightpaths on the optical layer to layout the virtual links for carrying the traffic specified in the traffic matrix from the packet layer. For that, it is necessary to solve a routing and spectrum assignment (RSA) problem [32]–[37]. Traffic grooming mechanisms groom and route packet flows into virtual links (*i.e.*, the lightpaths) [38], [39]. Previously, considering different network environment and optimization objective, the studies in [38]–[42] have formulated various ILP/MILP models and designed numerous heuristic algorithms to address network planning and service provisioning in packet-over-EONs. However, none of these studies have considered FlexE-over-EONs, and because they did not take the special features of FlexE into account, their approaches can hardly be leveraged to solve the

cross-layer network planning for FlexE-over-EONs, especially for the consideration of traffic grooming.

To accomplish traffic grooming in packet-over-EONs, the most well-known approach is based on the auxiliary graphs (AGs) [43]–[46]. Zhang *et al.* [43] proposed a three-layer AG model to address the mixed-electrical-optical grooming in packet-over-EONs under a dynamic traffic scenario. Also considering dynamic service provisioning, the study in [44] addressed the mixed channel traffic grooming in a shared backup path protected packet-over-EON and designed an AG-based heuristic to solve the problem. The authors of [45] formulated an ILP model to fully explore the adaptivity of packet-over-EONs for multilayer restoration. They also proposed an AG-based heuristic to reduce the time complexity of network planning. In [46], energy-efficient traffic grooming has been tackled in consideration of different kinds of BV-Ts and traffic scenarios. Nevertheless, all these studies did not address FlexE-over-EONs either, and they relied on ILP/MILP models and heuristics to solve the problem of traffic grooming, which either becomes intractable for large-scale problems or cannot provide approximated solutions with bounded distance (gaps) to the optimal ones.

The architectural advantages of FlexE-over-OTNs have been analyzed in [47], [48]. However, the authors only performed conceptual analysis and did not address the actual problem of cross-layer network planning. The cross-layer planning of FlexE-over-OTNs was first considered in [49], but the authors relied on the assumption that T-Boxes do not have FlexE-awareness, *i.e.*, the FlexE-partially-aware and FlexE-aware architectures were not addressed. The study in [50] was the first one that comprehensively accounted for the FlexE-unaware, FlexE-partially-aware, and FlexE-aware architectures as well as investigated the cross-layer planning in these architecture. Later on, the authors extended their work in [9] by formulating ILP models and designing greedy-based heuristics to solve the cross-layer planning. Nevertheless, since the heuristics are not approximation algorithms, they cannot get the solutions with bounded gaps to the optimal one. Moreover, the studies in [9], [50] did not consider FlexE-over-EONs. In [21], we formulated MILP models to optimize the cross-layer planning in FlexE-over-EONs and utilized them to demonstrate the benefits of FlexE-over-EONs in comparison to FlexE-over-WDMs. However, the formulation to the problem neither considered realistic assumptions nor approximation algorithms. Therefore, to the best of our knowledge, this is the first study that tackles the cross-layer planning in FlexE-over-EONs with in-depth theoretical analysis and approximation algorithm designs.

III. Problem Description

In this section, we first explain the operation principle of FlexE-over-EONs in the FlexE-aware architecture, and then describe the network model for cross-layer planning.

A. Operation Principle

Similar to the FlexE-over-WDMs discussed in [9], FlexE-over-EONs can also be realized based on the FlexE-unaware,

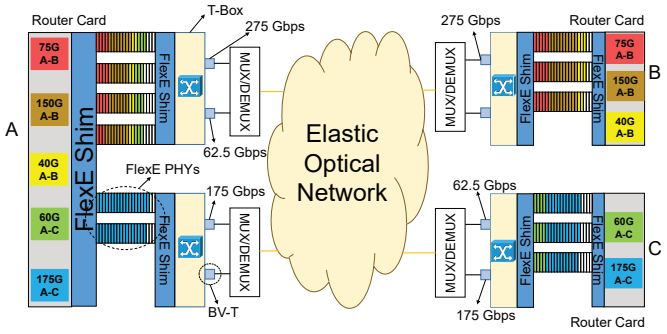


Fig. 2. Example of FlexE-over-EON based on the FlexE-aware architecture.

FlexE-partially-aware, and FlexE-aware architectures. In the FlexE-unaware architecture, the connections between PHYs and BV-Ts are preset and the BV-Ts have to use a fixed data-rate, while in the FlexE-partially-aware architecture, the BV-Ts can adjust their data-rates according to the usage of the CS'. In these two architectures, only the router cards possess FlexE shims, and this imposes limited flexibility. On the other hand, the T-Boxes in the FlexE-aware architecture are equipped with FlexE shims, which can recognize data from each MAC interface in the PHYs, allowing flow-level routing through the T-Boxes [7], [9]. In other words, the FlexE-aware architecture places FlexE shims not only between the MAC interfaces and PHYs but also between the PHYs and T-Boxes. Hence, instead of directly mapping PHYs to the BV-Ts in each T-Box, the architecture can sort out the data in each PHY and map them to the BV-Ts accordingly. Our analysis in [21] evinced that the FlexE-aware architecture is the most flexible and promising one for realizing FlexE-over-EONs. Therefore, this work considers only the FlexE-aware architecture for the cross-layer planning of FlexE-over-EONs.

Fig. 2 shows an example of the FlexE-over-EON in the FlexE-aware architecture. The architectures of FlexE-over-EONs and FlexE-over-WDMs are very similar, except for the FlexE-over-EONs being equipped with BV-Ts. Hence, Fig. 2 is adapted from the FlexE-over-WDM in the FlexE-aware architecture in [9]. There are three nodes (*i.e.*, *Nodes* A-C) in the FlexE-over-EON, and the colored boxes in the router card of each node represent the flows from/to the MAC interfaces of its FlexE clients. For the FlexE-over-EON, the cross-layer planning assigns flows from MAC clients to one or more T-Boxes through the PHYs connected to them, and then the flows are transmitted in the CS' corresponding to these PHYs, which is realized by leveraging the bonding, sub-rating and channelization mechanisms of FlexE [7]. Next, since each T-Box can identify the flows in the PHYs connected to it, the cross-layer planning allows serving the flows with its BV-Ts according to the flows' destinations.

For instance, if we assume that the total capacity of a T-Box is 400 Gbps and the capacity of each BV-T in a T-Box can be adjusted with a granularity of 12.5 Gpbs, the flows from *Node* A to *Nodes* B and C can be planned as illustrated in Fig. 2. Specifically, the cross-layer planning is conducted as follows. We first map the top four flows in *Node* A to the PHYs that connects to its first T-Box. Among the four flows, the first

three are all for A-B and their total capacity is 265 Gbps. This means that they can be sent out through the first BV-T with the rate of 275 Gbps (*i.e.*, corresponding to 22 frequency slots (FS') in the EON, if an FS provides a capacity of 12.5 Gbps). Then, the remaining capacity in the first T-box is 125 Gbps, which can only accommodate the fourth flow (*i.e.*, A-C at 60 Gbps). This is the reason why we only allocate the top four flows to the first T-Box in *Node* A. The second BV-T in the first T-Box uses a capacity of 62.5 Gbps to send the fourth flow to *Node* C, while the last flow is transmitted through the first BV-T in the second T-Box.

B. Network Model

With the aforementioned example, we can see that regarding the cross-layer network planning for FlexE-over-EON in the FlexE-aware architecture, two mappings need to be solved, *i.e.*, the mapping of flows from the MAC interfaces of FlexE clients to FlexE PHYs, and the mapping of FlexE PHYs to T-Boxes. These two mappings are correlated, and are restricted by the working principle of FlexE-over-EONs, making the cross-layer planning a complex problem to solve.

As shown in Fig. 2, each node in a FlexE-over-EON consists of router cards (on the FlexE side) and T-Boxes (on the EON side). We assume that each node is equipped with a fixed number of router cards, and each router card can send data through P PHYs and use them to connect to T T-Boxes. Due to the restrictions from hardware and cost, a T-Box usually possesses only a small number of BV-Ts [46], each can only set up one lightpath to a destination. However, the flows to a router card can choose arbitrary nodes in the network as destinations. Hence, if we want to ensure flexibility to FlexE-over-EONs, it is reasonable to assume that the flows to one router card will be served by multiple T-Boxes, *i.e.*, the capacity of a router card is larger than that of a T-Box, and this is also the case in practical implementations [7], [9]. Each router card needs to serve a few client flows from its MAC interfaces, while the client flows can have different bandwidth requirements and destination nodes. Each T-Box consists of B BV-Ts. We assume that the BV-Ts in each T-Box are sliceable BV-Ts [15], [17], which means that the capacity of a BV-T can be adjusted with a granularity of 12.5 Gbps and the total capacity of all the BV-Ts in a T-Box is fixed.

Regarding the network planning in the EON, we consider two scenarios, *i.e.*, the single-hop and multi-hop ones. The single-hop scenario assumes that the client flows are transmitted end-to-end an all-optically in the EON. In other words, if a client flow is mapped to a BV-T in one T-Box, the lightpath from the BV-T will be ended at the flow's destination node without any optical-to-electrical-to-optical (O/E/O) conversions in between. The single-hop scenario over-simplifies the network planning, and thus we also consider the multi-hop scenario in which each client flow can be routed over multiple lightpaths with O/E/O conversions and de-/re-grooming in intermediate nodes. Here, the O/E/O conversions are introduced to move the client flows between the EON and FlexE layers, but they are not for bypassing the spectrum continuity constraint on lightpaths. Hence, whether they cause wavelength conversions or not is irrelevant to our problem solving. Actually, the key problem in the multi-hop scenario is the design of the virtual topology, *i.e.*, how to plan the lightpaths to carry all the client flows with multi-hop routing. After getting the virtual topology, we transform the network planning into that of a single-hop scenario.

For the two scenarios, the cross-layer planning tries to minimize the number of T-Boxes used to carry client flows. Since previous studies have already addressed the RSA in EONs intensively, we would not explicitly solve it in our cross-layer planning. Specifically, after the lightpaths have been planned, their RSA schemes can be obtained by leveraging an existing algorithm (*e.g.*, the fragmentation-aware approaches in [12], [34], [51]). Note that, the problem considered in this paper is for static network planning, which means that it needs to be solved before a network operator actually builds its FlexE-over-EON. Similarly to other studies on EON planning (*e.g.*, in [32]), we assume that the optical spectra in the EON will always be sufficient to support all the lightpaths, and that all the client flows will be served. To ensure this assumption is practical, we can analyze the capacity of fiber links in the EON and limit the maximum number of T-Boxes per node accordingly. This makes our network model considers the fiber capacity constraints implicitly. In our future work, we will address the cases in which request blocking occurs due to insufficient fiber capacity.

IV. Single-Hop Scenario

In this section, we present the design algorithms to solve the cross-layer network planning in the single-hop scenario. We first formulate an MILP model [21] to describe the optimization, leverage it to transform the cross-layer planning to the class constrained bin packing problem (CCBP) [22], and then propose a polynomial-time approximation algorithm for it. We also design a greedy-based heuristic, which can be used as the benchmark for the performance comparisons related to large-scale problems.

A. MILP Model

The following MILP model describe the cross-layer planning for a FlexE-over-EON in the FlexE-aware architecture.

Notations:

- $G(V, E)$: the FlexE-over-EON's physical topology, where V and E are the sets of nodes and fiber links, respectively.
- R: the set of client flows, where r_i is the i-th client flow, which has a bandwidth demand of w_i in Gbps and s_i-d_i as source-destination pair .
- P: the number of PHYs that each router card can use.
- T: the number of T-Boxes that each router card can use.
- B: the number of BV-Ts in each T-Box.
- K_v: the set of router cards in node v, where $k \in K_v$ refers to such a router card.
- B_v: the set of BV-Ts in node v, where $b \in B_v$ refers to such a BV-T (*i.e.*, its T-Box and router card are denoted as $t_{v,b}$ and $k_{v,b}$, respectively).
- $B_{v,t}$: the set of BV-Ts in T-Box t of node v.

- $T_{v,k}$: the set of T-Boxes that the router card k in node v can use, where $t \in T_{v,k}$ refers to such a T-Box.
- C_p: the capacity of a PHY in Gbps ($C_p = 100$ Gbps).
- C_g: the capacity granularity of a BV-T ($C_g = 12.5$ Gbps).

Variables:

- $\alpha_{i,b}$: a boolean variable that is equal to 1 if client flow $r_i \in R$ is transmitted via BV-T b in node s_i, and 0 otherwise.
- $\beta_{v,k}$: a boolean variable that is equal to 1 if router card k in node v is used, and 0 otherwise.
- $\beta_{v,b}$: a boolean variable that is equal to 1 if BV-T b in node v is used, and 0 otherwise.
- $\beta_{v,t}$: a boolean variable that is equal to 1 if T-Box t in node v is used, and 0 otherwise.
- $p_{v,b}$: a nonnegative variable that indicates the used capacity of BV-T b in node v (*i.e.*, in terms of C_g).
- a_v: a nonnegative integer variable that indicates the number of used router cards in node v.
- b_v: a nonnegative integer variable that indicates the number of used T-Boxes in node v.
- c_v: a nonnegative integer variable that indicates the number of used BV-Ts in node v.

Objective:

The optimization objective is to minimize the total number of T-Boxes used in the cross-layer network planning.

$$\text{Minimize} \quad \sum_{v \in V} b_v. \tag{1}$$

Constraints:

$$\sum_{b \in B_{s_i}} \alpha_{i,b} = 1, \quad \forall r_i \in R. \tag{2}$$

Eq. (2) ensures that each flow r_i is transmitted via one and only one BV-T in its source node s_i.

$$\begin{aligned} &\alpha_{i,b} + \alpha_{j,b} \le \beta_{v,b}, \quad \forall v \in V,\ b \in B_v, \\ &\{i,j : r_i, r_j \in R,\ s_i = s_j = v,\ d_i \ne d_j\}. \end{aligned} \tag{3}$$

Eq. (3) ensures that a BV-T b in node v can carry only the client flows whose destination nodes are the same.

$$\sum_{\{i: r_i \in R,\ s_i = v\}} \alpha_{i,b} \cdot w_i \le p_{v,b} \cdot C_g, \quad \forall v \in V,\ b \in B_v. \tag{4}$$

Eq. (4) ensures that the total bandwidth of all the client flows assigned to each BV-T b does not exceed the BV-T's capacity, where w_i is the bandwidth demand of the i-th client flow.

$$\begin{aligned} \sum_{b \in B_{v,t}} \sum_{\{i: r_i \in R,\ s_i = v\}} \alpha_{i,b} \cdot w_i \le \frac{C_p \cdot P}{T}, \\ \forall v \in V,\ k \in K_v,\ t \in T_{v,k}. \end{aligned} \tag{5}$$

Eq. (5) ensures that the total bandwidth of all the client flows assigned to each T-Box t does not exceed the T-Box's capacity.

$$\beta_{v,b} \le \beta_{v,k_{v,b}}, \quad \forall v \in V,\ b \in B_v. \tag{6}$$

Eq. (6) ensures that when a BV-T b in node v is used, the corresponding router card is also marked as used.

$$\beta_{v,b} \le \beta_{v,t_{v,b}}, \quad \forall v \in V,\ b \in B_v. \tag{7}$$

Eq. (7) ensures that when a BV-T b in node v is used, the corresponding T-Box is also marked as used.

$$\sum_{k \in K_v} \beta_{v,k} \le a_v, \quad \forall v \in V, \tag{8}$$

$$\sum_{k \in K_v} \sum_{t \in T_t} \beta_{v,t} \le b_v, \quad \forall v \in V, \tag{9}$$

$$\sum_{b \in B_v} \beta_{v,b} \le c_v, \quad \forall v \in V. \tag{10}$$

Eqs. (8)-(10) ensure that the values of a_v, b_v, and c_v are correctly set, respectively.

Lemma 1. *The cross-layer network planning modeled with the MILP described above can be transformed into a general case of CCBP, and thus it is an $\mathcal{NP}$-hard problem.*

Proof. First of all, we can easily verify that in the single-hop scenario, minimizing the total number of used T-Boxes is equivalent to minimizing the number of T-Boxes used in each node. Therefore, we decompose the MILP into $|V|$ independent subproblems. For the subproblem about node $v \in V$, we treat all the client flows that originate from node v as items, each of which has a size of w_i (*i.e.*, the bandwidth demand) and a color class d_i (*i.e.*, the destination node). Each T-Box in node v is treated as a bin with capacity $\frac{C_p \cdot P}{T}$ that can accommodate items with B (*i.e.*, the number of BV-Ts in each T-Box) color classes at most. Then, we transform the subproblem into a general case of CCBP [22]. Hence, the cross-layer network planning can be transformed into a general case of CCBP. As CCBP is an $\mathcal{NP}$-hard problem, we also prove its $\mathcal{NP}$-hardness. □

B. Approximation Algorithm Design

Since the cross-layer planning for FlexE-over-EONs is $\mathcal{NP}$-hard, we will not try to design exact algorithms for it but decide to propose a polynomial-time approximation algorithm. The main idea of the approximation algorithm is to classify flows into different types based on their bandwidth demands and process similarly the flows of the same type. Thus, the time complexity can be significantly reduced compared to solving the MILP. Since the target problem can be transformed into a CCBP, we can leverage the primary-dual interior point (PDIP) method [23] to develop the approximation algorithm. *Algorithm* 1 shows the overall procedure. Here, we need to obtain the capacity of each T-Box as an input, *i.e.*,

$$C_{\max} = \frac{C_p \cdot P}{T}. \tag{11}$$

The for-loop checks each node $v \in V$ and minimizes the number of used T-Boxes in it in each iteration (*Lines* 1-8). Here, *Lines* 2-3 correspond to the initialization of the algorithm. We define the size of each flow by its normalized bandwidth demand[1]. Then, *Line* 4 uses *Algorithm* 2 to classify flows in R_v into small, medium, and the largest ones according to the flows' sizes and the preset tolerance ε. Next, we

[1]Note that, the normalization in *Line* 3 is just for the convenience of choosing the value of ε and classifying the flows accordingly, but it is not mandatory, *i.e.*, *Algorithm* 1 can operate without it.

construct a linear programming (LP) to serve the medium flows and utilize *Algorithm* 3 to solve it with the PDIP method (*Line* 5). Finally, we serve the small and largest flows with *Algorithm* 4 and obtain the number of used T-Boxes in node v (*Lines* 6-7). In *Line* 9, after checking all the nodes in V, the total number of used T-Boxes is returned.

Algorithm 1: Overall Procedure of PDIP-based Approximation Algorithm

Input: Physical topology $G(V, E)$, set of client flows R, capacity of a T-Box $C_{\max}$, and tolerance ε.

Output: Total number of used T-Boxes.

1 **for** *each node* $v \in V$ **do**
2 store all client flows originating from v in set R_v;
3 normalize bandwidth demands of flows in R_v as $\hat{w}_i = \frac{w_i}{C_{\max}}$, and use it as the size of each flow;
4 use *Algorithm* 2 to classify flows in R_v as small, medium, and largest ones based on their sizes;
5 build an LP to serve medium flows and use *Algorithm* 3 to solve it;
6 serve small and largest flows with *Algorithm* 4;
7 calculate number of used T-Boxes to store in b_v;
8 **end**
9 **return**$(\sum_{v \in V} b_v)$;

1) Flow Classification: To serve the flows originating from each node $v \in V$, we first classify them into a few subsets and label them as small, medium and the largest ones by using *Algorithm* 2 [22]. *Lines* 1-2 correspond to the initialization of the algorithm. In *Lines* 3-11, we first check each flow $r_i \in R_v$, and define its color as d_i (*i.e.*, flows to the same destination node have the same color). Then, if the flow's size $\hat{w}_i$ is not larger than the preset tolerance ε, we mark it as a small flow and insert it in set R_v^S (*Line* 6). Otherwise, the flow is inserted in set $R_v^{L,u}$ according to its color (*Lines* 8-9). Next, the for-loop that covers *Lines* 12-22 further divides $R_v^{L,u}$ into $K = \frac{1}{\varepsilon^2}$ subsets. Specifically, if there are at least K flows in $R_v^{L,u}$, we divide it into subsets $\{R_{v,1}^{L,u}, \cdots, R_{v,K}^{L,u}\}$ (*Line* 16), where the size of each subset satisfies

$$\left\lceil |R_v^{L,u}| \cdot \varepsilon^2 \right\rceil = |R_{v,1}^{L,u}| \geq \cdots |R_{v,k}^{L,u}| \geq \cdots \geq |R_{v,K}^{L,u}| = \left\lfloor |R_v^{L,u}| \cdot \varepsilon^2 \right\rfloor. \tag{12}$$

Otherwise, we set $R_{v,1}^{L,u}$ as an empty set, and divide $R_v^{L,u}$ into $K-1$ subsets with sizes satisfying Eq. (12) (*Lines* 18-19). Here, for each $u \in V'$, the flows in $R_{v,1}^{L,u}$ have the largest sizes, and thus we mark them as the largest ones, while the remaining flows in $\{R_{v,2}^{L,u}, \cdots, R_{v,K}^{L,u}\}$ are labeled as medium ones (*Line* 21). Finally, we set the size of each medium flow as the largest size in its subset in *Lines* 23-27.

2) Serving Medium Flows: Next, we try to serve all the medium flows at first, which can be done by formulating a linear program (LP) and solving it with the PDIP method [23]. For each node $v \in V$, we denote its set of medium flows as

$$R_v^M = \bigcup_{u \in V'} \left(\bigcup_{k=2}^{K} R_{v,k}^{L,u} \right). \tag{13}$$

Before formulating the LP to serve all the flows in R_v^M, we need to clarify the definitions of "flow type" and "allocation mode" since they are the key concepts for understanding it.

Definition 1. *Since all the flows in* $R_{v,k}^{L,u}$ *have the same size after* Algorithm *2, we denote their size as* $\tilde{w}_{v,k}^u$*. Hence, we define* ***the type of each flow*** *in* $R_{v,k}^{L,u}$ *as the tuple* $(\tilde{w}_{v,k}^u, u)$*. We denote the set of flow types as* F_T*.*

Definition 2. *An* ***allocation mode*** m *is a possible assignment of certain flows in* R_v^M *to a T-Box, which includes* $|F_T| + 1$

Algorithm 2: Flow Classification

Input: Set of client flows from node v (R_v) with normalized bandwidths $\{\hat{w}_i\}$, and tolerance ε.

Output: Sets of classified client flows R_v^S and $\{R_{v,k}^{L,u},\ \forall k \in [1, \frac{1}{\varepsilon^2}],\ \forall u \in V'\}$.

1 denote destination set of flows in R_v as $V' = V \setminus v$;
2 $R_v^S = \emptyset$, $\{R_v^{L,u} = \emptyset,\ \forall u \in V'\}$;
3 **for** *each flow* $r_i \in R_v$ **do**
4 define the color of r_i as its destination $d_i \in V'$;
5 **if** $\hat{w}_i \leq \varepsilon$ **then**
6 mark r_i as a small flow and insert it in R_v^S;
7 **else**
8 obtain the color of r_i as $u = d_i$;
9 insert flow r_i in $R_v^{L,u}$;
10 **end**
11 **end**
12 **for** *each node* $u \in V'$ **do**
13 $K = \frac{1}{\varepsilon^2}$;
14 sort flows in $R_v^{L,u}$ in descending order of sizes;
15 **if** $|R_v^{L,u}| \geq K$ **then**
16 partition $R_v^{L,u}$ into $\{R_{v,1}^{L,u}, \cdots, R_{v,K}^{L,u}\}$ in sorted order to satisfy Eq. (12), where $R_{v,1}^{L,u}$ contains the flows with the largest sizes;
17 **else**
18 set $R_{v,1}^{L,u} = \emptyset$;
19 divide $R_v^{L,u}$ into $\{R_{v,2}^{L,u}, \cdots, R_{v,K}^{L,u}\}$ in sorted order to satisfy Eq. (12);
20 **end**
21 mark each $r_i \in R_{v,1}^{L,u}$ as a largest flow, and label the remaining flows in $R_v^{L,u}$ as medium ones;
22 **end**
23 **for** *each node* $u \in V'$ **do**
24 **for** *each* $k \in [2, K]$ **do**
25 set sizes of all flows in $R_{v,k}^{L,u}$ as $\max_{r_i \in R_{v,k}^{L,u}} (\hat{w}_i)$;
26 **end**
27 **end**
28 **return**(R_v^S and $\{R_{v,k}^{L,u},\ \forall k \in [1, \frac{1}{\varepsilon^2}],\ \forall u \in V'\}$);

components. In the first $|F_T|$ components, the j-th one represents the number of type-j flows ($j \in [1, |F_T|]$) that are allocated to the T-Box. As each T-Box consists of T BV-Ts, the flows allocated to the T-Box cannot have more than T colors. Hence, the last component of m represents the set of the colors of the flows that are assigned to the T-Box, and we denote the component as $Dest(m)$.

Since R_v^M is known, we can obtain all the feasible allocation modes to assign certain medium flows to a T-Box and store them in set $\mathcal{M}$, based on which the LP is formulated as

Notations:

- $\mathcal{M}$: the set of allocation modes, where each $m \in \mathcal{M}$ represents a feasible allocation mode to assign certain medium flows to a T-Box.
- F_T: the set of flow types.
- $k_{j,m}$: the number of type-j flows ($j \in [1, |F_T|]$), which are assigned in allocation mode $m \in \mathcal{M}$.
- k_j: the total number of type-j flows in R_v^M.

Variables:

- γ_m: the nonnegative integer variable that indicates the number of T-Boxes that serve the flows according to allocation mode m, in the final network planning for R_v^M.
- $\delta_{u,\hat{V}}$: the nonnegative real variable that indicates the bandwidth reserved for small flows with color u in a T-Box, which uses an allocation mode m satisfying $Dest(m) = \hat{V}$, where $\hat{V}$ is a subset of V' that includes T colors, and u is a color in $\hat{V}$.

Objective:

The optimization objective of the LP is to minimize the total number of used T-Boxes.

$$\text{Minimize} \quad \sum_{m \in \mathcal{M}} \gamma_m. \tag{14}$$

Constraints:

$$\sum_{m \in \mathcal{M}} k_{j,m} \cdot \gamma_m \geq k_j, \quad \forall j \in [1, |F_T|]. \tag{15}$$

Eq. (15) ensures that all the flows in R_v^M are served.

$$\sum_{\{m: Dest(m) = \hat{V}\}} \left(1 - \sum_{j=1}^{|F_T|} k_{j,m} \cdot \tilde{w}_j\right) \cdot \gamma_m \geq \sum_{u \in \hat{V}} \delta_{u,\hat{V}}, \quad \{\hat{V} : \hat{V} \subseteq V',\ |\hat{V}| = T\}, \tag{16}$$

$$\sum_{\{\hat{V}: \hat{V} \subseteq V',\ |\hat{V}| = T\}} \delta_{u,\hat{V}} \geq \sum_{\{r_i: r_i \in R_v^S, d_i = u\}} \hat{w}_i, \quad \forall u \in V'. \tag{17}$$

Eqs. (16)-(17) ensure that the bandwidths reserved in all the T-Boxes are sufficient for serving the small flows in R_v^S, where the $\tilde{w}_j$ in Eq. (16) denotes the size of type-j medium flows.

$$\gamma_m \geq 0, \quad \forall m \in \mathcal{M}, \tag{18}$$

$$\delta_{u,\hat{V}} \geq 0, \quad \forall u \in \hat{V},\ \{\hat{V} : \hat{V} \subseteq V',\ |\hat{V}| = T\}. \tag{19}$$

Eqs. (18)-(19) ensure that the variables are nonnegative.

As the LP above is formulated based on allocation modes, the number of variables in it takes a polynomial form. Hence, the LP can be solved in polynomial-time with the PDIP method [23]. *Algorithm* 3 shows the detailed procedure.

Algorithm 3: Solving the LP to Serve Medium Flows

```
transform the LP into the standard form [52], with
  total number of variables denoted by N;
get the dual problem of the LP;
obtain the Karush-Kuhn-Tucker (KKT) condition
  [53] for the primal-dual problem;
define a "duality measure" μ to check the gap to the
  optimal solution;
while μ ≥ ε/N do
    solve the nonlinear optimization constructed
      with the Jacobian matrix of the KKT condition
      for the primary-dual problem;
    use the obtained solution X to update μ;
end
convert X to the solution of the LP: {γ*_m, δ_{u,V̂}};
γ_m = ⌈γ*_m⌉;
return({γ_m, δ_{u,V̂}});
```

3) Serving Small and the Largest Flows: Finally, we design *Algorithm* 4 to serve the small and the largest flows based on the network planning for the medium ones. Here, for node $v \in V$, the set of largest flows can be obtained as

$$R_v^L = \bigcup_{u \in V'} R_{v,1}^{L,u}. \tag{20}$$

In *Lines* 1-3, we allocate a new T-Box to serve each largest flow in R_v^L. Next, the for-loop covering *Lines* 4-11 tries to serve small flows with the remaining bandwidths in the T-Boxes that have been allocated to carry medium flows by *Algorithm* 3. Finally, if there are still unserved small flows, we allocate new T-Boxes to serve them (*Lines* 12-14).

4) Complexity Analysis and Approximation Ratio: It is easy to verify that *Algorithms* 2 and 4 have polynomial-time complexity. The time complexity of the overall procedure in *Algorithm* 1 is dominated by that of the PDIP method in *Algorithm* 3. We know that the PDIP method can be accomplished in polynomial-time [23]. Therefore, *Algorithm* 1 is a polynomial-time algorithm to solve the cross-layer planning for FlexE-over-EON in the single-hop scenario.

Algorithm 2 ensures that the number of flow types for the medium flows in R_v^M is at most $|V'| \cdot (\frac{1}{\varepsilon^2} - 1)$, and for any flow in $R_{v,k}^{L,u}$ ($k \geq 2$), its new size $\tilde{w}_{v,k}^u$ will not be greater than the original size of any flow in $R_{v,k-1}^{L,u}$. Hence, if we define the optimal solution of flow set R as $OPT(R)$, we have

$$OPT(R_v^M) \leq OPT(R_v). \tag{21}$$

Next, the performance of the PDIP method in *Algorithm* 3 guarantees that its solution $\{\gamma_m,\ m \in \mathcal{M}\}$ satisfies [23]

$$\sum_{m \in \mathcal{M}} \gamma_m \leq (1 + \varepsilon) \cdot OPT(R_v^M) + (|F_T| + |V'| + |V'|^T), \tag{22}$$

where ε is the preset tolerance and $(|F_T| + |V'| + |V'|^T)$ is a constant. We then consider the largest flows in R_v^L. Note that the operation principle of the cross-layer planning ensures

$$\sum_{r_i \in R_v^M} \hat{w}_i \leq OPT(R_v^M). \tag{23}$$

Algorithm 4: Serving the Small and Largest Flows

Input: Set of largest flows R_v^L, set of small flows R_v^S, and the solution of *Algorithm* 3: $\{\gamma_m, \delta_{u,\hat{V}}\}$.
Output: Number of additional T-Boxes.

1 **for** *each largest flow* $r_i \in R_v^L$ **do**
2 assign r_i to a new T-Box (dedicated to it only);
3 **end**
4 **for** *each used allocation mode* m *based on* $\{\gamma_m\}$ **do**
5 $\hat{V} = Dest(m)$;
6 calculate remaining bandwidth in the T-Box that uses m as σ_m;
7 **for** *each color* $u \in \hat{V}$ **do**
8 calculate the remaining bandwidth for small flows with color u as $\sigma_{m,u}$;
9 assign unserved small flows with color u to the current T-Box greedily in descending order of their sizes until $\sigma_{m,u}$ is used up;
10 **end**
11 **end**
12 **for** *each color* $u \in \hat{V}$ **do**
13 allocate new T-Boxes and assign unserved small flows with color u (if there are any) to them;
14 **end**
15 **return**(Number of additional T-Boxes);

Meanwhile, for each $u \in V'$, we have

$$|R_{v,1}^{L,u}| \leq 3\varepsilon^2 \cdot |R_v^{L,u} \setminus R_{v,1}^{L,u}| \leq 3\varepsilon \cdot \sum_{r_i \in (R_v^{L,u} \setminus R_{v,1}^{L,u})} \hat{w}_i, \quad (24)$$

as long as we have $\varepsilon \leq \frac{1}{3}$. Therefore, by adding up the two sides of the inequality in Eq. (24), we have

$$|R_v^L| \leq 3\varepsilon \cdot \sum_{r_i \in R_v^M} \hat{w}_i \leq 3\varepsilon \cdot OPT(R_v^M). \quad (25)$$

Hence, based on *Lines* 1-3 in *Algorithm* 4, we can conclude that the number of the additional T-Boxes demanded by the largest flows in R_v^L is upper-bounded by $3\varepsilon \cdot OPT(R_v^M)$. Finally, we consider the small flows in R_v^S. The LP solved by *Algorithm* 3 ensures that the bandwidth reserved for small flows with color u in a T-Box, which uses an allocation mode m satisfying $Dest(m) = \hat{V}$ and $u \in \hat{V}$, is at least $\delta_{u,\hat{V}}$. This suggests that for each $m \in \mathcal{M}$ and $u \in Dest(m)$, there is at most one small flow that has not been served after *Line* 11 of *Algorithm* 4. Consequently, with the condition that the size of any small flow will not be greater than ε, we can conclude that the additional bandwidths for these small flows are at most

$$T \cdot \varepsilon \cdot \sum_{m \in \mathcal{M}} \gamma_m^* \leq T \cdot \varepsilon \cdot (1+\varepsilon) \cdot OPT(R_v^M) \leq 2T \cdot \varepsilon \cdot OPT(R_v^M), \quad (26)$$

where γ_m^* denotes the exact solution to the LP. Hence, the number of additional T-Boxes for these small flows is at most

$$\frac{2T \cdot \varepsilon \cdot OPT(R_v^M)}{1-\varepsilon} + |V'| \leq 3T \cdot \varepsilon \cdot OPT(R_v^M). \quad (27)$$

By summarizing the right sides of the inequalities in Eqs. (22), (25) and (27), we obtain that the total number of T-Boxes is upper-bounded by

$$\varpi = [1 + \varepsilon \cdot (3T+4)] \cdot OPT(R_v^M) + (|F_T| + |V'| + |V'|^T), \quad (28)$$

which leads to an approximation ratio of

$$\frac{\varpi}{OPT(R_v)} \leq \frac{\varpi}{OPT(R_v^M)} = [1 + \varepsilon \cdot (3T+4)] + \left(\frac{|F_T| + |V'| + |V'|^T}{OPT(R_v^M)} \right), \quad (29)$$

according to Eq. (21). To this end, we verify that *Algorithm* 1 is a polynomial-time approximation algorithm.

C. Heuristic Algorithm for Single-Hop Scenario

For the performance comparisons in Section VI, we still need a heuristic for the single-hop scenario. As the cross-layer planning in FlexE-over-EONs has not been studied in the literature, we cannot simply adapt an existing heuristic. Therefore, we leverage the idea in [9] to design a greedy-based heuristic, as shown in *Algorithm* 5. Specifically, the heuristic serves all the client flows in R sequentially in a greedy manner, using the outer for-loop (*Lines* 1-15). For each flow $r_k \in R$, we first try to leverage a used T-Box in its source s_k to transmit it to d_k (*Lines* 3-11). If this fails, we allocate a new T-Box in s_k to serve r_k (*Line* 12-14). We can easily verify that *Algorithm* 5 is also a polynomial-time algorithm. Nevertheless, it can only provide feasible solutions, but cannot guarantee bounded performance gaps to the optimal solutions.

Algorithm 5: Heuristic for Single-Hop Scenario

Input: Physical topology $G(V, E)$, set of client flows R, and capacity of a T-Box $C_{\max}$.
Output: Total number of used T-Boxes.

1 **for** *each flow* $r_k \in R$ **do**
2 $flag = 0$;
3 **for** *each used T-Box* t *in source node* s_k **do**
4 **if** $flag = 0$ **then**
5 **if** *T-Box* t *has enough capacity to support* w_k *and one of its BV-Ts goes to* d_k **then**
6 $flag = 1$;
7 assign r_k to t and update its capacity;
8 **break**;
9 **end**
10 **end**
11 **end**
12 **if** $flag = 0$ **then**
13 allocate a new T-Box in s_k to serve r_k and connect a BV-T in it to d_k;
14 **end**
15 **end**
16 **return**(Total number of used T-Boxes);

V. Multi-Hop Scenario

In this section, we consider the multi-hop scenario in which each client flow can be routed over multiple lightpaths with O/E/O conversions and de-/re-grooming in intermediate nodes. We tackle the cross-layer planning for the multi-hop scenario with a two-step approach. Specifically, we first solve the virtual topology design to plan the smallest number of lightpaths for carrying all the client flows with multi-hop routing, and then map the client flows in each node to T-Boxes with *Algorithm* 1. Therefore, the focus of this section is the virtual topology design, which can be modeled with the common flow-based ILP in Appendix A. We propose a polynomial-time approximation algorithm to solve the ILP, and for performance evaluations, we also design a greedy-based heuristic for handling the multi-hop scenario with one algorithm.

Different from the single-hop scenario, the two-step approach does not tackle the cross-layer planning with only one optimization problem. More precisely, since we divide the original problem into two optimization problems, a lower bound cannot be computed in the procedure, and thus we cannot obtain the approximation ratio analytically. However, as we formulate the optimization problems in two steps to work coordinately and we propose approximation algorithms for the optimizations in the two steps, the performance of the overall cross-layer planning can be maintained at acceptable levels, which will be illustrated in Section VI.

A. Approximation Algorithm for Virtual Topology

Previous studies have already verified that the virtual topology design in network planning is an $\mathcal{NP}$-hard problem [38], [42]. Therefore, we also restore to design a polynomial-time approximation algorithm for it. Specifically, we design the approximation algorithm by leveraging the well-known branch-and-bound method [26], and fix the number of iterations to get a modified branch-and-bound (MBB) approach. By doing so, the proposed algorithm can obtain a qualified solution in polynomial-time. Note that, the MBB approach might not be generalized to all ILP models, and it is applicable to the virtual topology design due to the characteristics of our ILP model.

Algorithm 6 shows the procedure of the MBB-based approximation algorithm. *Lines* 1-3 correspond to the initialization of the algorithm. Specifically, we relax the ILP for virtual topology design obtaining an LP, which is solved by using the PDIP method whose procedure is similar to that of *Algorithm* 3. The solution $\{x_{i,j}, y_k^{i,j}\}$ is given by real numbers. Then, the for-loop in Lines 4-15 uses I iterations to optimize the solution of the LP. In each iteration, we select the $x_{i,j}$ whose value is the maximum, mark it as processed, and use its value to generate new constraints and get two new LPs $\mathcal{L}_1$ and $\mathcal{L}_2$ (*Lines* 5-8). Then, we solve the new LPs, and utilize the one that provides the smallest objective values to update the solution of the original LP (*Lines* 9-14). After the for-loop, *Line* 16 rounds up the real numbers in $\{x_{i,j}, y_k^{i,j}\}$ to get an integer solution. However, since the solution obtained in *Line* 16 is just an approximated one, it might set the value of $y_k^{i,j}$ larger than the correct one, *i.e.*, two routing paths might be assigned to a flow r_k. Hence, we recalculate the routing path of each flow by applying the Dijkstra's algorithm on the designed virtual topology (governed by $\{x_{i,j}\}$), and update $\{y_k^{i,j}\}$ accordingly (*Lines* 17-20).

Algorithm 6: MBB-based Virtual Topology Design

Input: Set of nodes V, set of client flows R, preset number of iterations I, and tolerance ε.
Output: Virtual topology design $\{x_{i,j}, y_k^{i,j}\}$.

1 relax the ILP of Eqs. (31)-(39) to get an LP;
2 solve the LP with PDIP method (similar procedure of *Algorithm* 3) to get a solution $\{x_{i,j}, y_k^{i,j}\}$;
3 mark variables $\{x_{i,j},\ \forall i,j\}$ as unprocessed and store them in set X;
4 **for** $n = 1$ *to* I **do**
5 $\quad \hat{x} = \max_{x_{i,j} \in X}(x_{i,j})$, $(i^*, j^*) = \operatorname{argmax}_{x_{i,j} \in X}(x_{i,j})$;
6 $\quad$ mark x_{i^*,j^*} as processed and remove it from X;
7 $\quad$ add a new constraint to the LP: $x_{i^*,j^*} \geq \lfloor \hat{x} \rfloor + 1$, to get a new LP $\mathcal{L}_1$;
8 $\quad$ add a new constraint to the LP: $x_{i^*,j^*} \leq \lfloor \hat{x} \rfloor$, to get another new LP $\mathcal{L}_2$;
9 $\quad$ solve LPs $\mathcal{L}_1$ and $\mathcal{L}_2$ with PDIP method;
10 $\quad$ select LP with smaller objective from $\mathcal{L}_1$ and $\mathcal{L}_2$;
11 $\quad$ denote solution of the chosen LP as $\{\tilde{x}_{i,j}, \tilde{y}_k^{i,j}\}$;
12 $\quad$ **if** *all variables* $\{\lceil \tilde{x}_{i,j} \rceil\}$ *satisfy Eq.* (32) **then**
13 $\quad\quad$ $x_{i,j} = \tilde{x}_{i,j}$, $y_k^{i,j} = \tilde{y}_k^{i,j}$, $\forall i,j,k$;
14 $\quad$ **end**
15 **end**
16 $x_{i,j} = \lceil x_{i,j} \rceil$, $y_k^{i,j} = \lceil y_k^{i,j} \rceil$, $\forall i,j,k$;
17 **for** *each* $r_k \in R$ **do**
18 $\quad$ run the Dijkstra's algorithm in the designed virtual topology for r_k to finalize its routing path;
19 $\quad$ update $\{y_k^{i,j}\}$ accordingly;
20 **end**
21 **return**($\{x_{i,j}, y_k^{i,j}\}$);

The time complexity of *Algorithm* 6 is dominated by that of the PDIP method, and thus it is a polynomial-time algorithm. Based on the principle of the MBB-based approach, we can get the approximation ratio of *Algorithm* 6 as [26]

$$(1+\varepsilon) + \left[\frac{|V| \cdot (|V|-1)}{OPT} \right], \tag{30}$$

where OPT is the objective of the optimal solution. With the virtual topology designed in *Algorithm* 6, we can use the values of $\{y_k^{i,j}\}$ to easily transform the multi-hop scenario to the single-hop one. Then, the cross-layer planning can be solved by using the algorithms developed in Section IV.

B. Heuristic Algorithm for Multi-Hop Scenario

Similar to the case of the single-hop scenario, we also design a greedy-based heuristic for the multi-hop scenario, shown in *Algorithm* 7. For each flow $r_k \in R$, we first try to transmit it directly to d_k with an end-to-end lightpath that

Algorithm 7: Heuristic for Multi-Hop Scenario

Input: Set of nodes V, set of client flows R, and capacity of a T-Box $C_{\max}$.
Output: Total number of used T-Boxes.

1 **for** *each flow* $r_k \in R$ **do**
2 $flag = 0$;
3 **for** *each used T-Box* t *in source node* s_k **do**
4 **if** $flag = 0$ **then**
5 **if** *T-Box* t *has enough capacity to support* w_k *and one of its BV-Ts goes to* d_k **then**
6 $flag = 1$;
7 assign r_k to t and update its capacity;
8 **break**;
9 **end**
10 **end**
11 **end**
12 **if** $flag = 0$ **then**
13 find a routing path based on used T-Boxes in the FlexE-over-EON to serve r_k;
14 **if** *the path can be found* **then**
15 serve r_k with the used T-Boxes on the path and update their capacities;
16 **else**
17 allocate a new T-Box in s_k to serve r_k and connect a BV-T in it to d_k;
18 **end**
19 **end**
20 **end**
21 **return**(Total number of used T-Boxes);

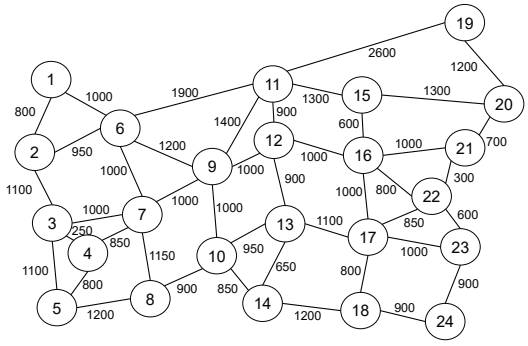


Fig. 3. US Backbone topology.

originates from a used T-Box in its source s_k (*Lines* 3-11). If this fails, we then try to calculate a multi-hop path with the used T-Boxes in the FlexE-over-EON to route the flow (*Lines* 13-15). But if the path still cannot be found, we allocate a new T-Box in s_k to transmit r_k to d_k with an end-to-end lightpath (*Line* 17). *Algorithm* 7 is also a polynomial-time algorithm with no performance guarantee.

VI. Simulation Results

In this section, we perform simulations to evaluate the performance of our algorithms for cross-layer network planning.

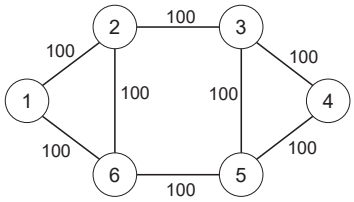


Fig. 4. Six-node topology.

A. Simulation Setup

The simulations consider both the single-hop and multi-hop scenarios. For the single-hop scenario, we compare the results of the MILP model in Section IV-A to those of *Algorithm* 1. Since the client flows originating from different nodes can be handled independently in the single-hop scenario, we use the 24-node US Backbone (USB) topology in Fig. 3 as the physical topology. Regarding the multi-hop scenario, we first compare the performance of the ILP in Appendix A with *Algorithm* 6 for the virtual topology design. Since solving the ILP is intractable for a relatively large topology, we only simulate them with the six-node topology shown in Fig. 4. Then, we evaluate the combination of approximation algorithms (*i.e.*, *Algorithms* 1 and 6) and the heuristic (*Algorithm* 7) to assess their overall performance on the cross-layer planning in the multi-hop scenario, using the USB topology in Fig. 3.

We select the bandwidth demands of client flows from $\{10, 40, 25 \cdot \lambda\}$ Gbps, where λ is the bit-rate update multiplier of MAC interfaces [7], [9] and its value is normally within $[1, 8]$. To study the performance of cross-layer planning for different traffic distributions, the simulations consider three scenarios

- *Random Traffic*: bandwidth demands are randomly selected from $\{10, 40, 25 \cdot \lambda\}$ Gbps, where we have $\lambda \in [1, 8]$.
- *Light Traffic*: bandwidth demands are randomly selected from $\{10, 40, 25 \cdot \lambda\}$ Gbps, where we have $\lambda \in [1, 4]$.
- *Heavy Traffic*: bandwidth demands are randomly selected from $\{10, 40, 25 \cdot \lambda\}$ Gbps, where we have $\lambda \in [5, 8]$.

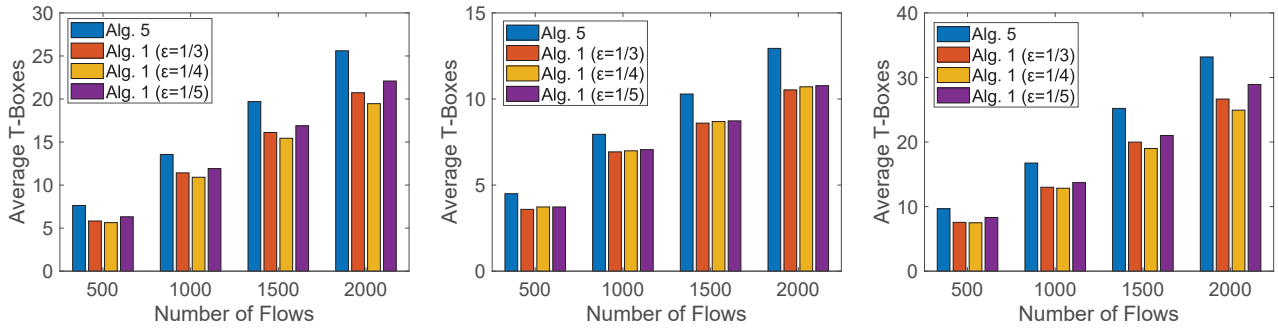
The source-destination pair of each flow is randomly selected. We assume that each T-Box includes $B = 2$ BV-Ts, the capacity of a PHY is $C_p = 100$ Gbps, the maximum capacity of a T-Box is $C_{\max} = 400$ Gbps, and the capacity granularity of each BV-T is $C_g = 12.5$ Gbps. In order to ensure the statistical accuracy of simulation results, we run each simulation with 10 independent sets of client flows, and average the results to obtain each data point. All the simulations have been conducted on a computer with 1.6 GHz Inter Core i5-8250 CPU and 8 GB memory, and the simulation environment is MATLAB 2019a with Gurobi optimization toolbox.

B. Single-Hop Scenario

For the single-hop scenario, we first generate $|R| \in \{100, 200, 300, 400\}$ client flows in the 24-node USB, and use the MILP and *Algorithm* 1 to solve the cross-layer planning. Here, we set $\varepsilon = \frac{1}{4}$ in *Algorithm* 1, and select the number of T-Boxes in each node (T) according to $|R|$. For instance, we

TABLE I
PERFORMANCE OF MILP AND *Algorithm* 1 FOR THE SINGLE-HOP SCENARIO

$\|R\|$	Random Traffic				Light Traffic				Heavy Traffic			
	MILP		*Algorithm* 1		MILP		*Algorithm* 1		MILP		*Algorithm* 1	
	Average T-Boxes	Running Time (s)	Average T-Boxes	Running Time (s)	Average T-Boxes	Running Time (s)	Average T-Boxes	Running Time (s)	Average T-Boxes	Running Time (s)	Average T-Boxes	Running Time (s)
100	1.52	6.50	1.81	0.08	1.28	6.26	1.97	0.06	1.69	6.74	2.13	0.09
200	2.49	9.33	2.97	0.16	1.91	7.11	2.57	0.11	2.98	9.85	3.68	0.18
300	3.60	16.19	4.06	0.24	2.32	10.97	3.15	0.17	4.14	59.46	4.99	0.24
400	4.44	32.61	4.97	0.25	2.73	18.40	3.69	0.19	5.35	27685.35	5.90	0.36



(a) Random Traffic (b) Light Traffic (c) Heavy Traffic

Fig. 5. Results for large-scale simulation results employing *Algorithm* 5 and *Algorithm* 1 for the single-hop scenario.

set $T = 6$ for $|R| = 400$. Table I shows the simulation results for the three traffic scenarios, where "Average T-Boxes" refers to the average number of used T-Boxes per node. We can see that the numbers of used T-Boxes from the MILP are always smaller than those demanded by the use of *Algorithm* 1, while the gaps between the results from the MILP and *Algorithm* 1 always satisfy the approximation ratio in Eq. (29). In the meantime, the results in Table I clearly indicate the advantage of our approximation algorithm in terms of time complexity. For $|R| = 400$ client flows, the running time of the MILP under heavy traffic is more than 7 hours, but that of *Algorithm* 1 is less than 0.4 second.

Then, we increase the number of client flows to consider $|R| \in \{500, 1000, 1500, 2000\}$ and further evaluate the performance of *Algorithm* 1. Note that, the MILP is intractable for these cases (*i.e.*, it cannot provide the solution within 24 hours). Hence, we only simulate *Algorithm* 1 with ε from $\{\frac{1}{3}, \frac{1}{4}, \frac{1}{5}\}$ and *Algorithm* 5. We still select T based on $|R|$ (*e.g.*, $T = 40$ for $|R| = 2000$), and still consider the three traffic scenarios. Fig. 5 shows results that indicate that *Algorithm* 1 with $\varepsilon = \frac{1}{4}$ outperforms those with $\varepsilon \in \{\frac{1}{3}, \frac{1}{5}\}$ under the random and heavy traffic scenarios, while in the light traffic scenario, the one with $\varepsilon = \frac{1}{3}$ uses the lowest number of T-Boxes. This suggests that for *Algorithm* 1, the selection of ε should be empirical according to the actual traffic distribution.

We observe that for all the simulation scenarios, *Algorithm* 1 (the approximation algorithm) outperforms *Algorithm* 5 (the heuristic) in terms of the average number of T-Boxes required in the cross-layer planning. This confirms that our approximation algorithm plans FlexE-over-EONs in a more cost-efficiently way than does the greedy-based heuristic. The running time of the two algorithms for the problems, with largest simulation size (*i.e.*, $|R| = 2000$), is listed in Table II. We can see that the running time of *Algorithm* 1 decreases with ε. This is because with a smaller ε, the iterations in *Algorithm* 3 take longer time. Since *Algorithm* 5 is just a greedy-based heuristic that does not use iterative optimization, it runs much faster than *Algorithm* 1. Therefore, the approximation algorithm sacrifices running time for cost-efficiency of cross-layer planning. Note that the problem addressed in this paper is for static network planning, which should be solved in an offline manner before the FlexE-over-EON is actually built. Hence, the running time of network planning algorithms will not be a serious issue. In other words, the network operator is willing to spend more time on network planning, as long as the used algorithm is not intractable and can achieve significant savings on CAPEX.

TABLE II
RUNNING TIME OF ALGORITHMS FOR THE SINGLE-HOP SCENARIO

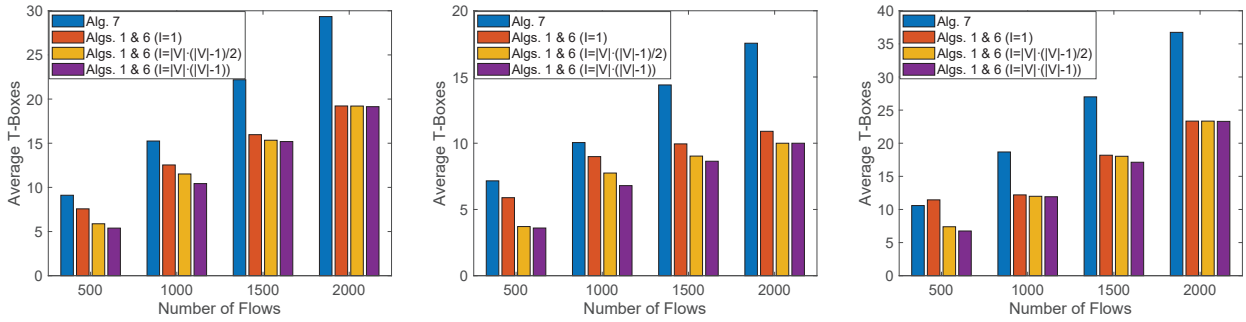
Running Time (s)				
$\|R\| = 2000$	*Algorithm* 1	$\varepsilon = \frac{1}{3}$	$\varepsilon = \frac{1}{4}$	$\varepsilon = \frac{1}{5}$
		1.94	9.71	230.87
	Algorithm 5	0.01		

C. Multi-Hop Scenario

In the multi-hop scenario, the cross-layer planning needs to solve two problems. The first is the virtual topology design of the underlying EON, and the second is the network planning for the single-hop scenario. Therefore, we tackle the cross-layer planning with a two-step approach. For the first problem

TABLE III
PERFORMANCE OF ILP AND *Algorithm* 6 FOR THE VIRTUAL TOPOLOGY DESIGN

$\|R\|$	Random Traffic				Light Traffic				Heavy Traffic			
	ILP		*Algorithm* 6		ILP		*Algorithm* 6		ILP		*Algorithm* 6	
	Average VLs	Running Time (s)	Average VLs	Running Time (s)	Average VLs	Running Time (s)	Average VLs	Running Time (s)	Average VLs	Running Time (s)	Average VLs	Running Time (s)
50	18.80	5.80	20.50	5.74	12.40	20.42	14.20	5.08	19.00	12.87	22.00	5.88
100	29.20	72.88	34.60	9.94	17.00	62.42	21.60	9.21	35.40	361.08	37.80	10.59
150	39.80	145.93	46.00	20.26	23.20	74.46	25.60	20.55	49.40	532.49	57.00	24.87
200	53.20	442.06	60.60	32.65	29.20	371.81	34.40	31.32	63.00	2165.31	70.60	35.98



(a) Random Traffic (b) Light Traffic (c) Heavy Traffic

Fig. 6. Large-scale simulation results of *Algorithms* 1 & 6 ($\varepsilon = \frac{1}{4}$) and *Algorithm* 7 for multi-hop scenario.

(*i.e.*, the virtual topology design), we design both an ILP model and a polynomial-time approximation algorithm (*Algorithm* 6). Hence, the simulations first compare the ILP with the *Algorithm* 6, and due to the time complexity of the ILP, only the six-node topology in Fig. 4 is considered. Then, in order to evaluate the overall performance of our two-step approach in the multi-hop scenario, we run simulations with larger topologies (*i.e.*, the 24-node USB in Fig. 3) to compare the combination of approximation algorithms (*Algorithms* 1 and 6) with the heuristic (*Algorithm* 7).

In the simulations that compare the performance of the ILP with that of *Algorithm* 6, we generate $|R| \in \{50, 100, 150, 200\}$ client flows, and select the iteration number as $I = |V| \cdot (|V| - 1)$ in *Algorithm* 6. The other parameters are the same as those in the simulations for the single-hop scenario. Table III illustrates the results for the three traffic scenarios, where "Average VLs" refers to the average number of planned lightpaths (*i.e.*, virtual links (VLs)) in the virtual topology. We observed that the numbers of planned lightpaths by the ILP are always smaller than those estimated by the *Algorithm* 6, while the gaps between the results from the MILP and *Algorithm* 1 always satisfy the approximation ratio in Eq. (30). Moreover, Table IV shows the advantage of our approximation algorithm in terms of time complexity. For $|R| = 200$ client flows, the running time of the ILP under heavy traffic is more than half an hour, but that of *Algorithm* 6 is around 35 seconds.

Next, we evaluate the performance of our two-step approach for the overall cross-layer planning in the multi-hop scenario. We consider $|R| \in \{500, 1000, 1500, 2000\}$ for the 24-node USB. Note that both the MILP and ILP are intractable in these cases, and thus we only simulate the combination of approximation algorithms (*Algorithms* 1 and 6) and the heuristic (*Algorithm* 7). For the approximation algorithms, we set $\varepsilon = \frac{1}{4}$, and $I \in \{1, \frac{|V|}{2} \cdot (|V| - 1), |V| \cdot (|V| - 1)\}$. The simulations consider the three traffic scenarios. Fig. 6 shows the simulation results, which indicate that when the number of flows increases, the advantage of *Algorithms* 1 and 6 over *Algorithm* 7 becomes more and more obvious, and for the combination of approximation algorithms, the gaps between the results obtained with different values of I gradually decrease. The algorithms running time for the largest problems (*i.e.*, $|R| = 2000$) is listed in Table IV. We observe that *Algorithms* 1 and 6 with $I = 1$ take less than 12 minutes to accomplish the whole optimization, while if we increase I to $|V| \cdot (|V| - 1)$, it will take around 3 hours.

Considering the fact that for these problems, *Algorithms* 1 and 6 do not provide significant different results with $I = 1$ and $I = |V| \cdot (|V| - 1)$. We can conclude that using a small I value is sufficient for the combination of approximation algorithms to handle large-scale problems. The heuristic (*Algorithm* 7) takes shorter running time, but as shown in Fig. 6, its solutions are much less cost-efficient than those from *Algorithms* 1 and 6. Once again, as our problem is for static network planning, the operator will pay much more attention on the CAPEX of required equipment.

VII. CONCLUSION

In this paper, we studied the problem of cross-layer network planning for FlexE-over-EONs, and focused our problem-solving on the FlexE-over-EONs based on the FlexE-aware architecture. We first considered the single-hop scenario in

TABLE IV
RUNNING TIME OF ALGORITHMS FOR THE MULTI-HOP SCENARIO

Running Time (s)			
$\|R\| = 2000$	*Algorithms* 1 & 6	$\varepsilon = \frac{1}{4}$	
		$I = 1$	$I = \|V\| \cdot (\|V\| - 1)$
		697.39	10573.30
	Algorithm 7	0.26	

which all the client flows are assumed to be routed over end-to-end lightpaths in the EON. We proved that the cross-layer planning for this scenario can be transformed into CCBP. We proposed a polynomial-time approximation algorithm to solve it based on the PDIP method. Next, we expanded our study to address a more realistic multi-hop scenario, where each client flow can be routed over multiple lightpaths in the EON. We formulated the virtual topology design as an ILP model, and then designed a polynomial-time approximation algorithm based on MBB. With the virtual topology designed, we obtained the hop-by-hop lightpath routing of each client flow, and transformed the cross-layer planning to that of the single-hop scenario. To evaluate the performance of our two-step method for the multi-hop scenario, we also proposed a heuristic algorithm. Extensive simulations confirmed that regarding large-scale cross-layer planning for FlexE-over-EONs, our approximation algorithms significantly outperform the ILP/MILP models in terms of running time, and the results produced by these algorithms have smaller difference to the optimal solution (gap) than those produced by the heuristic.

APPENDIX A
ILP MODEL FOR VIRTUAL TOPOLOGY DESIGN

As the physical topology $G(V, E)$ is definitely a connected graph to ensure feasible cross-layer network planning, the virtual topology design does not need to care too much about the physical links. Hence, we number the nodes in V with indices in $[1, |V|]$, and refer to each node with its index.

Notations:

- V: the set of nodes in the physical topology, and each node is referred by the index $i \in [1, |V|]$.
- R: the set of client flows, where r_k is the k-th flow, with a bandwidth demand of w_k in Gbps and a source-destination pair as s_k-d_k.
- T: the number of T-Boxes in each node.
- $C_{\max}$: the maximum capacity of a lightpath (*i.e.*, $C_{\max}$ can be obtained by using Eq. (11)).

Variables:

- $x_{i,j}$: a nonnegative integer variable that indicates the number of directed lightpaths from node i to node j.
- $y_k^{i,j}$: a boolean variable that equals 1 if flow r_k uses a directed lightpath from node i to node j, and 0 otherwise.

Objective:

The optimization objective is to minimize the total number of lightpaths planned in the virtual topology.

$$\text{Minimize} \sum_{i,j \in [1,|V|]} x_{i,j}. \tag{31}$$

Constraints:

$$\sum_{j \in [1,|V|]} x_{i,j} \le T, \quad \forall i \in [1, |V|]. \tag{32}$$

Eq. (32) ensures that the number of planned lightpaths does not exceed the number of T-Boxes in each node.

$$y_k^{i,j} \le x_{i,j}, \quad \forall r_k \in R,\ i \in [1, |V|],\ j \in [1, |V|]. \tag{33}$$

Eq. (33) ensures that flows will not be assigned to nonexistent lightpaths.

$$\sum_{j \in [1,|V|]} y_k^{s_k,j} = 1, \quad \forall r_k \in R. \tag{34}$$

Eq. (34) ensures that each flow is assigned to one and only one lightpath that is from its source.

$$\sum_{i \in [1,|V|]} y_k^{i,d_k} = 1, \quad \forall r_k \in R. \tag{35}$$

Eq. (35) ensures that each flow is assigned to one and only one lightpath that is ended at its destination.

$$\sum_{i \in [1,|V|]} y_k^{i,l} = \sum_{j \in [1,|V|]} y_k^{l,j}, \quad \forall r_k \in R, \\ \{l : l \in [1, |V|], l \ne s_k, l \ne d_k\}. \tag{36}$$

Eq. (36) ensures that each flow is handled correctly at the intermediate nodes on its routing path.

$$\sum_{i \in [1,|V|]} y_k^{i,l} \le 1,\ \forall r_k \in R,\ \{l : l \in [1, |V|], l \ne s_k, l \ne d_k\}. \tag{37}$$

Eq. (37) ensures that for each node in the topology, there is at most one lightpath from it, which carries the flow r_k.

$$\sum_{j \in [1,|V|]} y_k^{l,j} \le 1,\ \forall r_k \in R,\ \{l : l \in [1, |V|], l \ne s_k, l \ne d_k\}. \tag{38}$$

Eq. (38) ensures that for each node in the topology, there is at most one lightpath to it, which carries the flow r_k. Note that, Eqs. (33)-(38) are the constraints to ensure flow conservation, *i.e.*, they guarantee that for any flow $r_k \in R$, the routing path from its source to its destination is unique and loopless.

$$\sum_{r_k \in R} y_k^{i,j} \cdot w_k \le x_{i,j} \cdot C_{\max}, \quad \forall r_k \in R,\ i, j \in [1, |V|]. \tag{39}$$

Eq. (39) ensures that the total bandwidth of the flows assigned to each lightpath does not exceed its capacity.

ACKNOWLEDGMENTS

This work was supported in part by the NSFC projects 61871357, 61771445 and 61701472, ZTE Research Fund PA-HQ-20190925001J-1, Zhejiang Lab Research Fund 2019LE0AB01, CAS Key Project (QYZDY-SSW-JSC003), and SPR Program of CAS (XDC02070300).

REFERENCES

[1] "Cisco global cloud index: Forecast and methodology, 2016-2021." [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/white-paper-c11-738085.html.

[2] A. Gupta and R. Jha, "A survey of 5G network: Architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, Aug. 2015.

[3] 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-datacenter networks," *IEEE Netw.*, vol. 29, pp. 36–42, Sept./Oct. 2015.

[4] N. Xue, X. Chen, S. Li, L. Gong, D. Hu, and Z. Zhu, "Demonstration of OpenFlow-controlled network orchestration for adaptive SVC video manycast," *IEEE Trans. Multimedia*, vol. 17, pp. 1617–1629, Sept. 2015.

[5] J. Liu, W. Lu, F. Zhou, P. Lu, and Z. Zhu, "On dynamic service function chain deployment and readjustment," *IEEE Trans. Netw. Serv. Manag.*, vol. 14, pp. 543–553, Sept. 2017.

[6] "FlexE." [Online]. Available: https://en.wikipedia.org/wiki/FlexE.

[7] "Flex Ethernet 2.0 implementation agreement," Optical Internetworking Forum, Jun. 2018. [Online]. Available: https://www.oiforum.com/wp-content/uploads/2019/01/OIF-FLEXE-02.0-1.pdf.

[8] H. M. N. S. Oliveira, I. Katib, N. L. S. da Fonseca, and D. Medhi, "Comparison of network protection in three-layer ip/mpls-over-otn-over-dwdm networks," in *In Proc. of 2015 IEEE Globecom*, 2015, pp. 1–6.

[9] A. Eira, A. Pereira, J. Pires, and J. Pedro, "On the efficiency of flexible ethernet client architectures in optical transport networks," *J. Opt. Commun. Netw.*, vol. 10, pp. A133–A143, Jan. 2018.

[10] O. Gerstel, M. Jinno, A. Lord, and B. Yoo, "Elastic optical networking: A new dawn for the optical layer?" *IEEE Commun. Mag.*, vol. 50, pp. 12–20, Feb. 2012.

[11] L. Gong, X. Zhou, X. Liu, W. Zhao, W. Lu, and Z. Zhu, "Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. 836–847, Aug. 2013.

[12] Y. Yin, H. Zhang, M. Zhang, M. Xia, Z. Zhu, S. Dahlfort, and S. Yoo, "Spectral and spatial 2D fragmentation-aware routing and spectrum assignment algorithms in elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. A100–A106, Oct. 2013.

[13] L. Gong and Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 450–460, Feb. 2014.

[14] L. Zhang and Z. Zhu, "Spectrum-efficient anycast in elastic optical inter-datacenter networks," *Opt. Switch. Netw.*, vol. 14, pp. 250–259, Aug. 2014.

[15] M. Jinno, H. Takara, Y. Sone, K. Yonenaga, and A. Hirano, "Multiflow optical transponder for efficient multilayer optical networking," *IEEE Commun. Mag.*, vol. 50, pp. 56–65, May 2012.

[16] Z. Zhu, C. Chen, S. Ma, L. Liu, X. Feng, and S. Yoo, "Demonstration of cooperative resource allocation in an OpenFlow-controlled multidomain and multinational SD-EON testbed," *J. Lightw. Technol.*, vol. 33, pp. 1508–1514, Apr. 2015.

[17] N. Sambo, P. Castoldi, A. D'Errico, E. Riccardi, A. Pagano, M. Moreolo, J. Fabrega, D. Rafique, A. Napoli, S. Frigerio, M. Salas, G. Zervas, M. Nolle, J. Fischer, A. Lord, and J. Gimenez, "Next generation sliceable bandwidth variable transponders," *IEEE Commun. Mag.*, vol. 53, pp. 163–171, Feb. 2015.

[18] S. Dahlfort, M. Xia, R. Proietti, and S. Yoo, "Split spectrum approach to elastic optical networking," in *Proc.of ECOC 2012*, Sept. 2012, pp. 1–3.

[19] W. Lu, X. Zhou, L. Gong, M. Zhang, and Z. Zhu, "Dynamic multi-path service provisioning under differential delay constraint in elastic optical networks," *IEEE Commun. Lett.*, vol. 17, pp. 158–161, Jan. 2013.

[20] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *J. Lightw. Technol.*, vol. 31, pp. 15–22, Jan. 2013.

[21] W. Lu, J. Kong, L. Liang, S. Liu, and Z. Zhu, "How much can flexible ethernet and elastic optical networking benefit mutually?" in *Proc. of ICC 2019*, May 2019, pp. 1–6.

[22] L. Epstein, C. Imreh, and A. Levin, "Class constrained bin packing revisited," *Theor. Comput. Sci.*, vol. 411, pp. 3073–3089, Apr. 2010.

[23] J. Nocedal and S. Wright, *Numerical Optimization*. Springer Science & Business Media, 2006.

[24] H. M. N. S. Oliveira and N. L. S. da Fonseca, "Multipath routing, spectrum and core allocation in protected SDM elastic optical networks," in *In Proc. of 2019 IEEE Globecom*, 2019, pp. 1–6.

[25] ——, "P-cycle protected multipath routing, spectrum and core allocation in SDM elastic optical networks," in *In Proc. of 2019 IEEE ICC*, 2019, pp. 1–6.

[26] E. Lawler and D. Wood, "Branch-and-bound methods: A survey," *Oper. Res.*, vol. 14, pp. 699–719, Jan. 1966.

[27] Z. Zhu, M. Funabashi, Z. Pan, B. Xiang, L. Paraschis, and S. Yoo, "Jitter and amplitude noise accumulations in cascaded all-optical regenerators," *J. Lightw. Technol.*, vol. 26, pp. 1640–1652, Jun. 2008.

[28] Z. Pan, H. Yang, J. Yang, J. Hu, Z. Zhu, J. Cao, K. Okamoto, S. Yamano, V. Akella, and S. Yoo, "Advanced optical-label routing system supporting multicast, optical TTL, and multimedia applications," *J. Lightw. Technol.*, vol. 23, pp. 3270–3281, Oct. 2005.

[29] H. Jiang, Y. Wang, L. Gong, and Z. Zhu, "Availability-aware survivable virtual network embedding (A-SVNE) in optical datacenter networks," *J. Opt. Commun. Netw.*, vol. 7, pp. 1160–1171, Dec. 2015.

[30] L. Gong, H. Jiang, Y. Wang, and Z. Zhu, "Novel location-constrained virtual network embedding (LC-VNE) algorithms towards integrated node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 24, pp. 3648–3661, Dec. 2016.

[31] B. Li, W. Lu, S. Liu, and Z. Zhu, "Deep-learning-assisted network orchestration for on-demand and cost-effective vNF service chaining in inter-DC elastic optical networks," *J. Opt. Commun. Netw.*, vol. 10, pp. D29–D41, Oct. 2018.

[32] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Elastic bandwidth allocation in flexible OFDM-based optical networks," *J. Lightw. Technol.*, vol. 29, pp. 1354–1366, May 2011.

[33] L. Gong, X. Zhou, W. Lu, and Z. Zhu, "A two-population based evolutionary approach for optimizing routing, modulation and spectrum assignments (RMSA) in O-OFDM networks," *IEEE Commun. Lett.*, vol. 16, pp. 1520–1523, Sept. 2012.

[34] M. Zhang, C. You, H. Jiang, and Z. Zhu, "Dynamic and adaptive bandwidth defragmentation in spectrum-sliced elastic optical networks with time-varying traffic," *J. Lightw. Technol.*, vol. 32, pp. 1014–1023, Mar. 2014.

[35] H. Wu, F. Zhou, Z. Zhu, and Y. Chen, "On the distance spectrum assignment in elastic optical networks," *IEEE/ACM Trans. Netw.*, vol. 25, pp. 2391–2404, Aug. 2017.

[36] W. Lu and Z. Zhu, "Dynamic service provisioning of advance reservation requests in elastic optical networks," *J. Lightw. Technol.*, vol. 31, pp. 1621–1627, May 2013.

[37] H. Wu, F. Zhou, Z. Zhu, and Y. Chen, "Analysis framework of RSA algorithms in elastic optical rings," *J. Lightw. Technol.*, vol. 37, pp. 1113–1122, Feb. 2019.

[38] S. Zhang, M. Tornatore, G. Shen, J. Zhang, and B. Mukherjee, "Evolving traffic grooming in multi-layer flexible-grid optical networks with software-defined elasticity," *J. Lightw. Technol.*, vol. 32, pp. 2905–2914, Aug. 2014.

[39] S. Liu, B. Niu, D. Li, M. Wang, S. Tang, J. Kong, B. Li, X. Xie, and Z. Zhu, "DL-assisted cross-layer orchestration in software-defined IP-over-EONs: From algorithm design to system prototype," *J. Lightw. Technol.*, vol. 37, pp. 4426–4438, Sept. 2019.

[40] M. Ruiz, O. Pedrola, L. Velasco, D. Careglio, J. Fernandez-Palacios, and G. Junyent, "Survivable IP/MPLS-over-WSON multilayer network optimization," *J. Opt. Commun. Netw.*, no. 8, pp. 629–640, Mar. 2011.

[41] P. Papanikolaou, K. Christodoulopoulos, and E. Varvarigos, "Joint multilayer survivability techniques for IP-over-elastic-optical-networks," *J. Opt. Commun. Netw.*, vol. 9, pp. A85–A98, Jan. 2017.

[42] W. Lu, X. Yin, X. Cheng, and Z. Zhu, "On cost-efficient integrated multilayer protection planning in IP-over-EONs," *J. Lightw. Technol.*, vol. 36, pp. 2037–2048, May 2018.

[43] J. Zhang, Y. Ji, M. Song, Y. Zhao, X. Yu, J. Zhang, and B. Mukherjee, "Dynamic traffic grooming in sliceable bandwidth-variable transponder-enabled elastic optical networks," *J. Lightwave Technol.*, vol. 33, pp. 183–191, Jan. 2015.

[44] F. Tang, W. Shao, X. Lian, S. Bose, and G. Shen, "Mixed channel traffic grooming for IP over EON with SBPP-based cross-layer protection," *J. Lightwave Technol.*, vol. 35, pp. 3836–3848, Jul. 2017.

[45] S. Liu, W. Lu, and Z. Zhu, "On the cross-layer orchestration to address IP router outages with cost-efficient multilayer restoration in IP-over-EONs," *J. Opt. Commun. Netw.*, vol. 10, pp. A122–A132, Jan. 2018.

[46] J. Zhang, Y. Zhao, X. Yu, J. Zhang, M. Song, Y. Ji, and B. Mukherjee, "Energy-efficient traffic grooming in sliceable-transponder-equipped IP-over-elastic optical networks," *J. Opt. Commun. Netw.*, vol. 7, pp. 142–152, Jan. 2015.

[47] S. Trowbridge, "Ethernet and OTN: 400G and beyond," in *Proc. of OFC 2015*, Mar. 2015, pp. 1–3.

[48] T. Hofmeister, V. Vusirikala, and B. Koley, "How can flexibility on the line side best be exploited on the client side?" in *Proc. of OFC 2016*, Mar. 2016, pp. 1–3.

[49] A. Patel, K. Kanonakis, P. Ji, J. Hu, and T. Wang, "Flexible-client: The missing piece towards transport software-defined networks," in *Proc. of OFC 2014*, Mar. 2014, pp. 1–3.
[50] A. Eira and J. Pedro, "How much transport grooming is needed in the age of flexible clients?" in *Proc. of OFC 2017*, Mar. 2017, pp. 1–3.
[51] P. M. Moura, N. L. S. da Fonseca, and R. A. Scaraficci, "Fragmentation aware routing and spectrum assignment algorithm," in *In Proc. of 2014 IEEE ICC*, 2014, pp. 1137–1142.
[52] N. Karmarkar and R. Karp, "An efficient approximation scheme for the one-dimensional bin-packing problem," in *Proc. of SFCS 1982*, Nov. 1982, pp. 312–320.
[53] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.