DL-Assisted Cross-Layer Orchestration in Software-Defined IP-over-EONs: From Algorithm Design to System Prototype

Siqi Liu, Bin Niu, Deyun Li, Min Wang, Shaofei Tang, Jawei Kong, Baojia Li, Xiaokang Xie, and Zuqing Zhu, *Senior Member, IEEE*

***Abstract*—Recently, with the development of IP and elastic optical networks (EONs), the network control and management (NC&M) scheme for IP-over-EONs, which can facilitate effective cross-layer orchestration (XLyr-O), has become an interesting but challenging research topic. In this paper, we consider a software-defined IP-over-EON (SD-IPoEON), leverage deep learning (DL) to analyze and predict the traffic fluctuation on established lightpaths in it, and design a proactive DL-assisted XLyr-O scheme. Specifically, we study the DL-assisted XLyr-O scheme from algorithm design to system prototype. A DL module based on the long/short-term memory based neural network (LSTM-NN) is first designed and optimized for precise IP traffic prediction. Then, we develop algorithms to explore the traffic prediction for realizing proactive XLyr-O to deal with hard/soft failures constantly, *i.e.*, making intelligent on-line decisions to re-groom and reroute IP flows and to reconfigure lightpaths such that the performance tradeoff among lightpath utilization, congestion probability, and reconfiguration frequency is balanced well. Finally, we implement our proposed algorithm in a small-scale but real SD-IPoEON testbed to prototype the DL-assisted XLyr-O, and conduct experiments with it. Experimental results demonstrate that compared with the reactive benchmark without DL-assistance, our proposal not only invokes less network reconfigurations but also reduces packet losses significantly.**

***Index Terms*—IP over Elastic optical networks (IP-over-EONs), Multi-layer restoration (MLR), Cross-layer orchestration, Software-defined networking (SDN), Artificial intelligence (AI).**

I. Introduction

OVER past decades, the raising of new network services has pushed the traffic in backbone networks to not only grow exponentially in volume but also become more and more bursty and dynamic [1]. This has applied and would continue to apply intensive pressure on the design and operation of backbone networks [2]. To address this issue properly, we first need an agile network architecture that can adaptively allocate spectrum resources in the underlying optical network to lightpaths, and then require an effective network control and management (NC&M) scheme that can utilize the lightpaths efficiently for grooming and routing IP flows, to achieve high resource utilization as well as good quality-of-service (QoS).

The former requirement can be satisfied by leveraging the technical advances on elastic optical networks (EONs) [3, 4] and combining IP and EON technologies rationally to realize IP-over-EONs [5, 6] (as shown in Fig. 1). This is because EONs enable flexible spectrum allocation with a granularity of 12.5 GHz or even narrower in the optical layer [7–9]. As for the second requirement, the effective NC&M scheme can hardly be realized if we manage the IP and EON layers separately. For example, without the cooperation between the two layers, we cannot balance the tradeoff between the spectrum utilization and packet losses in the EON and IP layers, respectively. Hence, the NC&M scheme has to facilitate cross-layer orchestration (XLyr-O), which implies a centralized mechanism such as software-defined networking (SDN) [10, 11]. Specifically, the XLyr-O can be achieved by building a software-defined IP-over-EON (SD-IPoEON) and using centralized SDN controller(s) to manage the packet switches in the IP layer and the bandwidth-variable transponders (BV-Ts) and optical cross-connects (BV-OXCs) in the EON layer in a coordinated manner [12, 13].

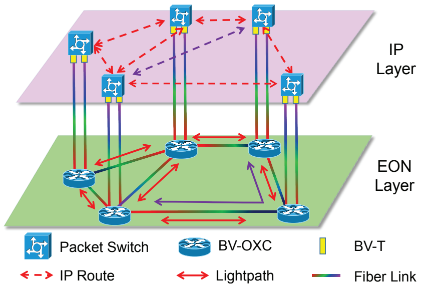


Fig. 1. A multi-layer IP-over-EON, BV-OXC: bandwidth-variable optical cross-connect, BV-T: bandwidth-variable transponder.

In an SD-IPoEON, the centralized controller not only coordinates lightpaths in the EON layer as the logic links to interconnect packet switches in the IP layer, but also steers IP traffic through the packet switches and logic links for traffic grooming and IP routing [14]. Although previous studies have already considered both the algorithms [6, 15, 16] and systems [12] to realize XLyr-O in SD-IPoEONs, the XLyr-O in SD-IPoEONs could still be improved. This is because these existing algorithms and systems make XLyr-O decisions only based on current network status. These reactive approaches, however, would make the service provisioning scenarios and traffic demands suffer from frequent mismatches, when the IP traffic is highly dynamic and bursty. Note that, due to the emerging of new services and the expansion of geographical

S. Liu, B. Niu, D. Li, M. Wang, S. Tang, J. Kong, B. Li, X. Xie 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).

Manuscript received on January 27, 2019.

coverage, the traffic fluctuation in backbone networks might not simply follow the well-known daily patterns anymore [17, 18]. Specifically, the authors of [17] observed counter-intuitive phenomena where the traffic flows in a backbone network could have various daily patterns and experience different peak/off-peak hours, while the study in [18] suggested that if a backbone network covers multiple time-zones geographically, traffic flows between different source-destination pairs might also have different peak/off-peak hours. Therefore, the operator would have the dilemma to either bear the increased NC&M complexity due to frequent network reconfigurations or suffer from IP route congestion/under-utilization constantly.

The dilemma could be resolved, if we leverage deep learning (DL) modules to analyze and predict the traffic fluctuation on established lightpaths in the SD-IPoEON and develop a proactive XLyr-O scheme. In this work, we study the DL-assisted XLyr-O scheme from algorithm design to system prototype. We first design a DL module based on the long/short-term memory based neural network (LSTM-NN) [19], and optimize its configuration to precisely capture the dynamics and self-similarity of future traffic. Then, we consider multi-layer restoration (MLR) in an SD-IPoEON as the usecase[1], and design algorithms to explore the traffic prediction results for realizing proactive XLyr-O, *i.e.*, making intelligent decisions to re-groom and reroute IP flows and to reconfigure lightpaths such that the performance tradeoff among lightpath utilization, congestion probability, and reconfiguration frequency can be balanced well. Finally, we extend our system design in [13], prototype an SD-IPoEON system with DL-assisted XLyr-O, and demonstrate our proposal experimentally.

The rest of paper is organized as follows. Section II provides a brief survey on the related work. We describe the operation principle of the DL-assisted XLyr-O and the design of the traffic predictor in Section III. Then, the algorithms to explore the traffic prediction results for realizing proactive XLyr-O are designed in Section IV, including both an integer linear programming (ILP) model and a heuristic, while they are evaluated with extensive simulations in Section V. Next, we lay out the system design and conduct experiments in a real network testbed to demonstrate our proposal experimentally in Section VI. Finally, Section VII summarizes the paper.

II. Related work

In an SD-IPoEON, the EON provides the optical interconnections to carry the traffic generated in the IP layer. The service provisioning in an EON involves the famous routing and spectrum assignment (RSA) problem, which is known to be NP-hard [20]. Hence, numerous studies have been devoted to designing the heuristics that can find near-optimal RSA solutions for building lightpaths [21, 22], light-trees [23, 24], light-forests [25, 26] and virtual optical networks [27]. Our XLyr-O scheme can leverage these algorithms to set up lightpaths or more complex structures in the EON layer. As for ensuring survivability and availability of EONs, people have considered both the path-based [28, 29] and link-based [30, 31] protection/restoration schemes. Although an EON can be protected well by these schemes, they cannot solve the problem of MLR in an SD-IPoEON, as explained in [6].

Considering various IP-over-optical networks, the studies in [32–34] have addressed multilayer protection, *i.e.*, allocating redundant backup resources in both IP and optical layers in advance for failure recovery. However, the dedicated protection schemes in [32, 33] would result in relatively low resource utilization in both IP and optical layers. Although the resource usage could be improved by introducing shared backup [34], the idling of backup IP routers, optical ports and fiber links would still affect the operator's revenue. More importantly, these multilayer protection schemes are all based on certain assumptions regarding the failure pattern, *e.g.*, the possibility of simultaneous failures can be ignored. Nevertheless, a recent analysis on the failure cases in Google's B2 and B4 networks suggested that failures in the IP layer could happen frequently [35] and thus simultaneous failures should not be overlooked. Meanwhile, as the backup resources have been reserved, the advantages of protection over restoration are shorter recovery time and guaranteed service recovery as long as the assumption on failure pattern holds. Moreover, with new techniques such as proactive protection [36], the service interruption time can be further reduced or even avoided since flows can be re-groomed onto a healthy lightpath in advance.

Hence, MLR should be considered since it does not reserve backup resources but obtains recovery schemes when failures actually happen. The studies in [6, 37] have discussed the algorithm designs for MLR, while SDN-based systems have been demonstrated in [12, 38]. However, they either assumed that IP flows have fixed bandwidth demands or just got the MLR schemes based on current network status. Hence, they are reactive approaches and could not avoid the mismatches between service provisioning scenario and traffic demands in the future. For instance, if we re-groom a highly bursty IP flow on a busy lightpath, congestions could happen in the future even though the current status shows no bandwidth shortage.

DL-based traffic prediction has been studied for various networks [39–42]. The work in [39] discussed how to realize precise traffic prediction but did not try to utilize the forecast for network optimization, while the authors of [40–42] considered DL-assisted network optimization. Nevertheless, none of these studies was about the XLyr-O scheme for SD-IPoEONs. Previously, we built a preliminary system to demonstrate the DL-assisted MLR in an SD-IPoEON [13]. However, the complete system design to realize DL-assisted XLyr-O has not been presented and the algorithm design was absent. Hence, this work extends our preliminary study in [13] by addressing DL-assisted XLyr-O for SD-IPoEONs from algorithm design to system prototype. To the best of our knowledge, this is the first work to cover the topic with such a scope.

III. Network Model and Operation Principle

In this section, we describe the network model of our DL-assisted XLyr-O system, explain its operation principle, and elaborate on the design of the DL-based traffic predictor.

[1]For the MLR, our proposed DL-assisted XLyr-O addresses both the hard-failure due to switch outages and the soft-failure because of traffic congestions. Hence, the algorithms designed for the MLR can be easily generalized for being used in normal service provisioning, *i.e.*, treating new IP flows to be served as the affected ones that need to be restored.

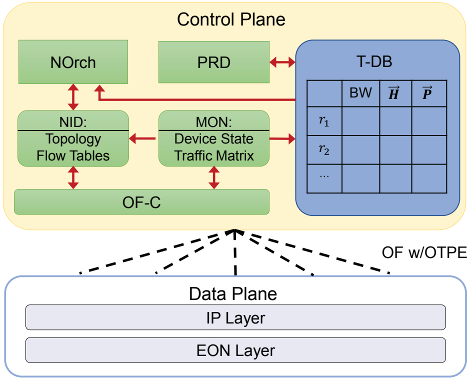


Fig. 2. Control plane design for realizing DL-assisted XLyr-O, OF w/OTPE: OpenFlow with optical transport protocol extensions, OF-C: OpenFlow controller, MON: Monitor, NID: Network information database, T-DB: Traffic database, PRD: Traffic predictor, NOrch: Network orchestrator.

A. Network Model

Fig. 1 shows the data plane of an SD-IPoEON, which consists of two layers. The EON layer is built with a few BV-OXCs interconnected by fiber links. The EON's topology is denoted as $G_o(V_o, E_o)$, where V_o and E_o represent the sets of BV-OXCs and fiber links, respectively. Each BV-OXC can transparently switch the optical spectrum of a lightpath from an input port to the desired output port, according to the instruction from the centralized controller in the control plane. Hence, the controller can set up, reconfigure, and tear down lightpaths in the EON to adapt to IP traffic.

The switches in the IP layer is also managed by the control plane. Each switch is co-located with an underlying BV-OXC and connects to it with several BV-Ts, each of which can generate or terminate the optical signal of a lightpath. Therefore, if there is a lightpath between the BV-Ts of two switches, they are directly connected with a logic link in the IP layer. Consequently, the IP layer can also be modeled as a graph $G(V, E)$, where V is the switch set and E denotes the set of the logic links that interconnect the switches in V. Specifically, if there is a logic link $(v, u) \in E$ and BV-OXCs v_o and u_o are co-located with switches v and u, respectively, there must be a lightpath to take v_o and u_o as its end nodes. Note that, the capacity of a lightpath depends on both the capability of its two BV-Ts and the quality-of-transmission (QoT) [3]. Therefore, in this work, we assume that each BV-T can support a set of line-rates and each line-rate has a maximum transmission reach under the QoT constraint. Meanwhile, since the BV-Ts on each switch are usually limited and thus precious, we also assume that when a new lightpath has to be established, the XLyr-O system always uses the highest line-rate that the lightpath's transmission distance permits.

B. Operation Principle

As shown in Fig. 2, our DL-assisted XLyr-O system resides in the control plane to coordinate the network elements (NEs) in IP and EON layers for realizing intelligent service provisioning in the SD-IPoEON. It uses the monitor (MON) to collect the traffic matrix in the IP layer periodically. MON sends the traffic matrix to the traffic database (T-DB), which records it in entries, each of which corresponds to a flow. Note that, as the SD-IPoEON is a backbone network, each flow here is actually an aggregated one that grooms numerous IP flows between a switch pair in the IP layer. Hence, the traffic on each flow could be highly dynamic, fluctuate with a unique pattern [17], and last for a reasonably long period (*e.g.*, tens of hours or even days). Meanwhile, since there might be multiple access networks behind each edge switch in the IP layer, we assume that there can be multiple flows between a switch pair. With the multi-protocol label switching (MPLS), switches can identify these flows by checking their labels.

Each entry in the traffic matrix models a flow r with the estimated maximum value of its bandwidth demand (BW), a vector for historical traffic samples ($\vec{H}$), and a vector for predicted traffic samples ($\vec{P}$). Here, the BW of a flow is assumed to be known *a priori*, its $\vec{H}$ gets updated by MON, and its $\vec{P}$ is obtained by the traffic predictor (PRD), which is based on an LSTM-NN. The network information database (NID) collects the multilayer topology of the data plane and the flow tables on the switches through the OpenFlow controller (OF-C), to get the latest flow routing schemes.

We design the DL-assisted XLyr-O system to address two issues with MLR, which are switch outages (*i.e.*, hard failures) and lightpath congestions (*i.e.*, soft failures). Here, a switch outage will take a switch offline to make all the flows, which use the switch as an end-node, as unrecoverable and interrupt all the flows using it as an intermediate node. A lightpath congestion occurs when the total instant bandwidth demand of the flows that are routed over a lightpath exceeds its capacity. Note that, the lightpath congestion can only be avoided if we groom flows to each lightpath according to their BWs. This, however, might lead to significant over-provisioning, since the flows can carry dynamic traffic with various patterns [17]. Meanwhile, even though upper-layer protocols such as TCP might have congestion-avoidance mechanisms, the flows would still experience bandwidth-shrinking during the process, which will degrade the QoS of their network services.

When MON detects an aforementioned issue in the data plane, it would invoke MLR to recover the affected flows and/or to avoid future lightpath congestions, as follows. MON first informs NID to update the flow routing schemes, and then tells T-DB to refresh the traffic prediction. Next, NID sends a request to the network orchestrator (NOrch) and lets it calculate the MLR scheme. NOrch obtains the MLR scheme based on the up-to-date multilayer topology, flow routing schemes stored in NID and the traffic prediction stored in T-DB, and feeds the scheme back to NID, which will in turn send it to OF-C for being implemented in the SD-IPoEON.

C. Generation of Traffic Data Set

The design of the DL-assisted XLyr-O system needs to consider realistic data sets to emulate the traffic generated in the IP layer of a backbone network. Specifically, to generate the dynamic traffic on a flow, we leverage the realistic traffic traces collected on an edge router, which aggregates the traffic from the access networks behind it. Here, the access networks are those for research institutions, enterprises, or even Internet

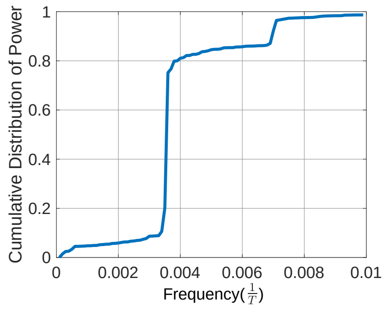


Fig. 3. DFT analysis of traffic samples in the training set.

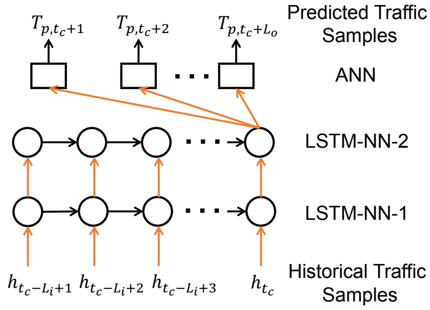


Fig. 4. DL-based traffic predictor with a three-layer model.

service providers (ISPs). We survey the related data on the Internet, and find 14 reasonable sources for the traces, each of which records real-time traffic and gets updated every a few minutes. Then, based on the realistic traces, we generate the data set to emulate dynamic traffic on the flows considered in this work. The data set includes 10 series of traffic samples, each of which consists of $59,000$ samples whose sampling interval is 5 minutes, *i.e.*, there are $590,000$ traffic samples in total. To save space, we publish the sources for realistic traffic traces, their characteristics, the procedure to generate traffic data set with them, and the generated data set on GitHub [43].

D. DL-based Traffic Prediction

We define the period of traffic collection as a time slot (TS) whose value is $T = 5$ minutes[2], since the SD-IPoEON is a backbone network and our DL-assisted XLyr-O system should care about long-term traffic fluctuations. To provide NOrch with sufficient traffic information for obtaining the cost-efficient MLR scheme, PRD forecasts a series of L_o traffic samples in the future. This means that the size of the vector for predicted samples (*i.e.*, $\vec{P}$) is L_o, *i.e.*, $\vec{P}$ covers a flow's traffic fluctuation over a period of $L_o \cdot T$.

Then, to design the DL-based traffic predictor, we first need to determine L_o since it is the number of outputs in the LSTM-NN. Here, we have a tradeoff to worry about, *i.e.*, if we use a larger L_o, NOrch calculates the MLR scheme with more information regarding future traffic fluctuation and thus can potentially avoid the mismatch between routing schemes and traffic demands for a longer time, but a larger L_o means a more complex LSTM-NN and lower prediction accuracy, and *vice versa*. Therefore, we come up with a simple method based on the Fourier analysis. Specifically, we generate a time series with the traffic samples in the training set, perform discrete Fourier transform (DFT) on it to get the frequency domain information regarding the traffic fluctuation, and determine L_o based on the DFT result. In the following, we will use an illustrative example to explain how to determine L_o.

The DFT provides us the cumulative distribution function (CDF) of the frequency components' power as in Fig. 3. It can be seen that the CDF increases sharply at around the frequency of $0.00348 \cdot \frac{1}{T}$, which implies that there is a significant component at the frequency in the traffic fluctuation.

[2]According to our survey on the real traffic traces in backbone networks, a commonly-used traffic sampling interval is 5 minutes. This interval is only for traffic sampling while a switch outage is detected and handled immediately.

Therefore, to record such a frequency component in traffic prediction, we need to set $L_o > \frac{T}{0.00348} = 287.3 \cdot T$, which makes $L_o = 288 \cdot T$. Specifically, for the traffic data in [43], if we set $L_o = 288 \cdot T$, the corresponding CDF result can be around 62% as shown in Fig. 3. Meanwhile, the DFT analysis also helps us determine the size of the inputs to the LSTM-NN. Specifically, to optimize the prediction accuracy, the input series should include the major low frequency components. For the traffic data in [43], the DFT analysis in Fig. 3 suggests that the low frequency component will not reach noticeable total power until $0.001 \cdot \frac{1}{T}$. This makes the LSTM-NN's input length as $L_i = 1008 \cdot T$. After determining the lengths of its inputs and outputs, we design a three-layer model for the DL-based traffic predictor, which includes two LSTM-NN layers to record and analyze the characteristics of traffic fluctuation, and one layer of artificial neural network (ANN) to buffer the prediction results (as shown in Fig. 4). The input historical traffic vector is denoted as $\vec{H} = [h_1, h_2, \cdots, h_{L_i}]^\top$.

To train and evaluate the DL-based traffic predictor, we divide the 10 traffic traces in [43] into training and testing sets with $540,000$ and $50,000$ samples, respectively. The predictor's output over a randomly-selected trace (*i.e.*, with $5,000$ samples) in the testing set is shown in Fig. 5, which indicates that the predicted and actual traces match with each other well. The trained traffic predictor provides prediction accuracies of 95.77% and 95.33% on the training and testing sets, respectively, and its training takes us 39.6 minutes.

IV. Algorithm Design for Cross-Layer Orchestration

In this section, we design the algorithms with which the NOrch in our XLyr-O system can calculate the MLR schemes based on traffic prediction.

A. Integer Linear Programming Model

As we have explained in Section III-A, our XLyr-O system needs to monitor the data plane of the SD-IPoEON constantly and implement the proper MLR scheme timely when a soft/hard-failure is detected. Specifically, the MLR scheme reconfigures the network elements in the data plane to restore the flows that have been interrupted by a switch outage or will be affected by future congestions. Here, the optimization objective is to restore all the affected but recoverable flows by reconfiguring the smallest number of flows and setting up the

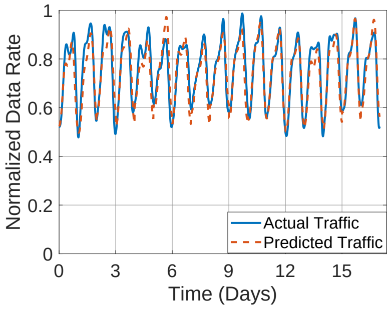


Fig. 5. Comparison of predicted and actual traffic.

fewest new lightpaths. Note that, we assume that the spectrum resources in the EON are more than sufficient to carry the traffic generated in the IP layer, which is usually the case in practical backbone networks. In other words, if needed, a direct lightpath can be established to connect any two nodes in the EON. Therefore, the RSA problem becomes trivial and we ignore it in our algorithm design.

Notations:

- $G(V,E)$: the IP layer topology, where V and E are the sets of switches and logical links, respectively.
- V^f: the set of failed switches.
- $e=(u,v,j)$, $e\in E$: the j-th logical link between switches u and v.
- K: a large number to denote the upper-bound on the lightpaths that can be set up between two switches.
- $k_{u,v}$: the number of active logical links between switches u and v.
- $C_{u,v}$: the bandwidth capacity (in Gbps) of the lightpath that can be set up between switches u and v.
- $\mathbf{R}=\mathbf{R^c}+\mathbf{R^f}$: the affected traffic matrix, where $\mathbf{R^c}$ denotes the set of flows that will be affected by future congestions, and $\mathbf{R^f}$ is the set of flows that have been interrupted by a switch outage but are recoverable[3].
- r: a flow in $\mathbf{R}$, *i.e.*, $r=(s_r,d_r)\in\mathbf{R}$, where s_r and d_r are its source and destination switches.
- L_o: the length of the predicted traffic samples.
- T_t^r: the t-th predicted traffic sample for flow $r\in\mathbf{R}$.
- ${x'}_e^r$: the indicator that equals 1 if flow $r\in\mathbf{R}$ originally uses logical link $e\in E$.
- b_t^e: the sample of unaffected background traffic on logical link $e\in E$ at the t-th TS in the future.
- M: a large number, *i.e.*, $M=\max(C_{u,v})\cdot K\cdot|\mathbf{R}|\cdot|V|^2$, where $|\cdot|$ returns the number of elements in a set.

Variables:

- x_e^r: the boolean variable that equals 1 if flow $r\in\mathbf{R}$ is routed on logical link $e\in E$.
- z^r: the boolean variable that equals 1 if flow $r\in\mathbf{R}$ is reconfigured.
- y_e: the boolean variable that equals 1 if a new lightpath e is established.

Objective:

[3]Here, an interrupted flow is recoverable if and only if the failed switch is not one of its end-nodes [6].

The optimization objective is to first minimize the number of reconfigured flows and then minimize the total bandwidth of newly-established lightpaths.

$$\text{Minimize} \quad M\cdot\sum_{r\in\mathbf{R}} z^r+\sum_{u,v\in V}\sum_{j=k_{u,v}+1}^{K} y_{(u,v,j)}\cdot C_{u,v}. \tag{1}$$

Constraints:

$$\sum_{v\in V}\sum_{j=1}^{K} x^r_{(u,v,j)}-\sum_{v\in V}\sum_{j=1}^{K} x^r_{(v,u,j)}=\begin{cases}1, & u=s_r,\\ -1, & u=d_r,\\ 0, & \text{others},\end{cases} \quad \forall r\in\mathbf{R}, \tag{2}$$

$$\sum_{v\in V}\sum_{j=1}^{K} x^r_{(u,v,j)}\le 1,\quad \forall r\in\mathbf{R},\ u\in V, \tag{3}$$

$$\sum_{u\in V}\sum_{j=1}^{K} x^r_{(u,v,j)}\le 1,\quad \forall r\in\mathbf{R},\ v\in V. \tag{4}$$

Eqs. (2)-(4) are the flow conservation constraints to guarantee that flow $r\in\mathbf{R}$ is routed from s_r to d_r over a single path. More specifically, Eq. (2) makes sure that r is correctly routed from s_r to d_r, while Eqs. (3) and (4) ensures that r only takes one routing path. Note that, Eqs. (3) and (4) also allow r to be routed on a newly-established logical link.

$$M\cdot z^r\ge\sum_{u,v\in V}\sum_{j=1}^{K}{x'}^r_{(u,v,j)}\cdot\left(1-x^r_{(u,v,j)}\right)\ge z^r,\quad \forall r\in\mathbf{R}. \tag{5}$$

Eq. (5) ensures that the value of z^r is correctly set. Specifically, if the routing scheme found by the ILP for r is different from its original one, z^r will be set as 1, and *vice versa*.

$$y_e\ge x_e^r,\quad \{e=(u,v,j),\ r\in\mathbf{R}:\forall e\in E,\ j>k_{u,v}\}. \tag{6}$$

Eq. (6) ensures that flow $r\in\mathbf{R}$ can only be routed on existing logical links.

$$b_t^{(u,v,j)}+\sum_{r\in\mathbf{R}} x^r_{(u,v,j)}\cdot T_t^r\le C_{u,v},\quad \forall u,v\in V,\ t\le L_o,\ j\le K. \tag{7}$$

Eq. (7) ensures that the traffic on any logic link will not exceed its capacity at any time within the prediction period.

$$y_{(u,v,j)}=0,\quad \forall u\in V^f,\ \forall v\in V^f. \tag{8}$$

$$x^r_{(u,v,j)}=0,\quad \forall u\in V^f,\ \forall v\in V^f. \tag{9}$$

Eqs. (8)-(9) ensures that none of the failed switches is used.

B. Heuristic Algorithm

Although the ILP model can obtain the optimal MLR solution to manage the SD-IPoEON well over a future period of $L_o\cdot T$, it would become intractable for large scale problems and thus cannot be used in dynamic service provisioning. Hence, we try to develop a heuristic to make sure NOrch can obtain MLR schemes time-efficiently. Note that, the MLR scheme needs to recover two kinds of flows in $\mathbf{R^f}$ and $\mathbf{R^c}$, respectively. For the flows in $\mathbf{R^f}$, they need to be reconfigured anyway and thus we should just try to restore them with the least bandwidth from the newly-established lightpaths. In other words, we should try to re-groom the flows in $\mathbf{R^f}$ onto the existing lightpaths as many as possible. However, since the traffic demands of all the flows in the SD-IPoEON are time-variant, how to determine the re-grooming scheme is more complex than the traditional re-grooming problem that addresses fixed-bandwidth flows. For instance, if a few flows whose peak time overlaps with each other are groomed on the same lightpath, congestions can happen frequently on the lightpath in the future even though its bandwidth utilization is not very high in most of the time.

On the other hand, it would be more complicated to restore the flows in $\mathbf{R^c}$, since we need to minimize the number of re-configured flows and the total bandwidth of newly-established lightpaths simultaneously. Therefore, selecting which flows to reconfigure would be an interesting and tricky problem. Based on these considerations, we divide the MLR problem into two subproblems, *i.e.*, 1) selecting the flows for reconfiguration, and 2) determining their reconfiguration schemes. To solve the first subproblem, we define a metric, namely, "congestion-relieving value (CRV)", for quantifying the relief on congestion if a flow r gets reconfigured.

$$\Delta_r = \sum_{e \in E_r} T^r_{t^c_e}, \tag{10}$$

where Δ_r is the CRV of flow r, E_r is the set of logical links that r traverses, t^c_e is the most congested time on logical link e in the predictable future, and $T^r_{t^c_e}$ is the traffic demand of r at time t^c_e. Note that, if a logical link e will not congest in the predictable future, $T^r_{t^c_e} = 0$. Then, *Algorithm* 1 is designed to solve the first subproblem.

Algorithm 1: Selecting Flows for Reconfiguration

Input: Affected flow set $\mathbf{R^c}$, congested link set E^c, and traffic prediction of all the flows $\{\vec{T}^r\}$.
Output: $\mathbf{R}'$ as the set of flows for reconfiguration

```
1  R' = ∅;
2  while E^c ≠ ∅ do
3      update R^c based on E^c;
4      for each flow r in R^c do
5          calculate Δ_r with Eq. (10);
6      end
7      r* = argmax_{r∈R^c}(Δ_r);
8      insert r* into R';
9      remove r* hypothetically and update E^c;
10 end
11 return R';
```

Here, *Line* 1 is for the initialization. The while-loop that covers *Lines* 2-10 selects flows to insert into $\mathbf{R}'$ iteratively, based on their CRVs. We update $\mathbf{R^c}$ by removing the flows whose logic links will not be congested in *Line* 3, according to the link set E^c. Then, *Lines* 4-6 calculate the CRVs of all the flows in $\mathbf{R^c}$. Note that, when calculating Δ_r, we also consider the scenario in which flow r goes through multiple congested logic links. Next, in *Lines* 7 and 8, we select the flow whose CRV is the maximum and insert it into $\mathbf{R}'$. Finally, *Line* 9 remove the selected flow from the network hypothetically and update E^c accordingly. For each while loop, *Algorithm* 1 takes out one request or at least one lightpath in E^c. Hence, the while-loop will run for at most $\max(|E^c|, |\mathbf{R^c}|)$ times. In each iteration, the complexity of calculating Δ_r is $O(|E^c| \cdot |\mathbf{R^c}| \cdot |\vec{T}^r|)$. Thus, the time complexity of *Algorithm* 1 is $O(\max(|E^c|, |\mathbf{R^c}|) \cdot |E^c| \cdot |\mathbf{R^c}| \cdot |\vec{T}^r|)$.

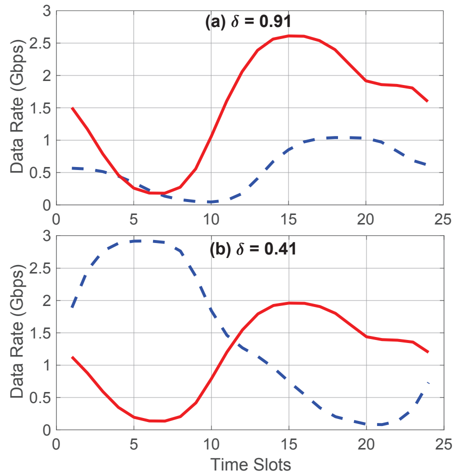


Fig. 6. Examples on peak time overlapping measurement with CS.

Next, we consider how to re-groom/reroute the flows selected by *Algorithm* 1 together with the interrupted ones in $\mathbf{R^f}$. To achieve this, we first need to design a method to evaluate the peak time overlapping among flows. Suppose we have the traffic prediction of two flows, *i.e.*, $\vec{P}^1 = [T^1_1, T^1_2, \cdots, T^1_{L_o}]$ and $\vec{P}^2 = [T^2_1, T^2_2, \cdots, T^2_{L_o}]$, their fluctuation trends can be highlighted with the following preprocessing

$$\vec{P} = \vec{P} - \min(\vec{P}). \tag{11}$$

Then, the peak time overlapping between two flows can be measured by calculating the cosine similarity (CS) of their preprocessed traffic prediction samples.

$$\delta_{\vec{P}^1,\vec{P}^2} = \frac{\vec{P}^1 \bigodot \vec{P}^2}{\sqrt{\sum_{i=1}^{L_o} (T^1_i)^2} \cdot \sqrt{\sum_{i=1}^{L_o} (T^2_i)^2}}. \tag{12}$$

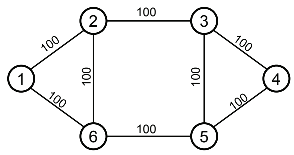
For any two series of predicted traffic samples, the more peak time overlapping they have, the larger their CS is. Fig. 6 shows two examples, which indicate that the CS of the two traces in Fig. 6(a) is larger than that of those in Fig. 6(b). Apparently, the two traces in Fig. 6(a) oscillate with each other in the time domain, while those in Fig. 6(b) fluctuates almost oppositely. Note that, grooming two flows on a lightpath, which fluctuate oppositely, helps to set apart their peak time, and thus the lightpath's bandwidth can be utilized more efficiently. With this idea, we design *Algorithm* 2 based on CS.

Algorithm 2 is designed to determine the reconfiguration schemes, *i.e.*, solving the second subproblem. The for-loop covering *Lines* 1-16 reconfigures the flows in the descending

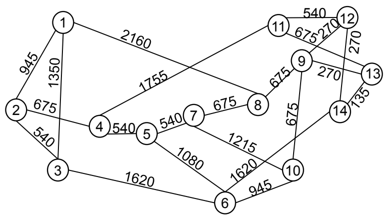
Algorithm 2: Determining Reconfiguration Schemes

Input: Set of flows to be reconfigured $\mathbf{R^a} = \mathbf{R}' \cup \mathbf{R^f}$, up-to-date IP layer topology $G(V, E)$, and traffic prediction of all the flows $\{\vec{T}^r\}$.

1 **for** *each* $r \in \mathbf{R^a}$ *in descending order of peak bit-rate* **do**
2 get the logic links that can accommodate r without causing congestions and store them in E';
3 **for** *each* $e \in E'$ **do**
4 get the predicted traffic samples of r and e, and calculate their CS $\delta_{r,e}$ with Eqs. (11) and (12);
5 calculate the NRC η_e of e with Eq. (13);
6 set the weight of e as $w_e = \delta_{r,e} \cdot \eta_e$;
7 **end**
8 **if** *the least weighted path can be found for* r **then**
9 store the path in p_r;
10 reconfigure r to use p_r and update $G(V, E)$;
11 **else**
12 set up a direct lightpath from s_r to d_r for r;
13 store the newly-established lightpath in p_r;
14 reconfigure r to use p_r and update $G(V, E)$;
15 **end**
16 **end**



(a) Six-node topology



(b) NSFNET topology

Fig. 7. EON topologies with fiber lengths in km marked on links.

order of their peak bit-rates, since a flow with a larger peak bit-rate may potentially cause more future congestions and thus is harder to be re-groomed. In each iteration, *Line* 2 is for the initialization. Then, we use *Lines* 3-7 to obtain the weight of each feasible link $e \in E'$. Specifically, we get the traffic predictions of r and e and use Eqs. (11) and (12) to calculate their CS $\delta_{r,e}$ (*Line* 4), and the normalized remaining capacity (NRC) of e is obtained as (*Line* 5)

$$\eta_e = \frac{C_e - \overline{T}_e}{\max\limits_{e' \in E} \left(C_{e'} - \overline{T}_{e'} \right)}, \tag{13}$$

where C_e is the capacity of link e, and $\overline{T}_e$ returns the average value of the predicted traffic samples on link e. Then, *Line* 6 calculates the weight of link e as $w_e = \delta_{r,e} \cdot \eta_e$. Note that, the smaller $\delta_{r,e}$ is, the less peak time overlapping between r and e, while the smaller η_e is, the less available will be on link e in the foreseeable future. Hence, if we use the least weighted path to carry r (*Lines* 8-10), it can be re-groomed in the way that can not only improve the utilization of existing lightpaths but also leave more available bandwidth for the subsequent flows. However, if such a feasible path cannot be found, *Lines* 12-14 sets up a new direct lightpath to carry r from s_r to d_r. In *Algorithm* 2, the complexity of sorting $\mathbf{R^a}$ is $O(|\mathbf{R^a}| \cdot \log_2 |\mathbf{R^a}|)$. The for-loop runs for $|\mathbf{R^a}|$ times. In each iteration, the complexity of calculating the weight of each link is $O(|E| \cdot |\vec{T}^r|)$ and the complexity of finding the least weighted path is $O(|E| + |V| \cdot \log |V|)$. Thus, the complexity of *Algorithm* 2 is $O(|\mathbf{R^a}| \cdot \max(\log_2 |\mathbf{R^a}|, |E| \cdot |\vec{T}^r|, |V| \cdot \log |V|))$

V. Numerical Simulations

A. Simulation Setup

The numerical simulations consider two topologies for the EON layer of the SD-IPoEON, *i.e.*, the six-node and NSFNET topologies in Figs. 7(a) and 7(b), respectively. We assume that each BV-T can support line-rates in $\{10, 25, 40, 50, 75, 100\}$ Gbps and the maximum transmission reaches of the line-rates are $\{3732, 2995, 2671, 2438, 2112, 1880\}$ km, respectively [44]. To save the BV-Ts on each switch, the XLyr-O system always uses the highest feasible line-rate when setting up a new lightpath. In each simulation, the initial logic links in the IP layer (*i.e.*, lightpaths) are first established. Here, the number of initial lightpaths between each switch pair is randomly selected within $[0, 2]$, while the total numbers of initial lightpaths in the six-node and NSFNET topologies are distributed within $[17, 21]$ and $[44, 50]$, respectively. Then, the SD-IPoEON starts to accept dynamic flow requests generated according to the Poisson model, and the traffic samples of the flows follow the data set that we generated with realistic traffic traces [43]. Since there is no historical information about the flows when they first come in, we provision them with a simple auxiliary graph based algorithm [6] that routes the flows over the shortest feasible path in the SD-IPoEON.

Next, we run the SD-IPoEON for a random period, stop to invoke one or more switch outages in the IP layer, and use the XLyr-O system to handle both the hard failure due to switch outage and the potential soft failure due to future congestions on the lightpaths. In addition to the ILP and the heuristic (CRV) discussed in the previous section, the simulations also consider two benchmark algorithms.

- **Expand**: With traffic prediction, it first selects the largest flow at the most congested time of a to-be-congested lightpath, and then tries to reroute the selected flow using a lightpath that shares the same s-d pair of the to-be-congested one and has the smallest CS from Eq. (12). If such a lightpath does not exist, it sets up a new one between the s-d pair to reroute the flow. The procedure is repeated until there is no lightpath congestion.
- **Mean**: It follows the similar procedure of CRV, but in *Algorithm* 1, it selects the flows that use the to-be-congested lightpaths and have the predicted traffic samples whose average values are the largest, and puts them in $\mathbf{R^f}$ for reconfiguration.

TABLE I
AVERAGE RUNNING TIME PER FLOW (SECONDS)

Traffic Volume before Outage (Tb/s)	1.25	1.5	1.75	2.25	2.5
ILP	32.647	48.215	51.146	53.545	73.107
CRV	0.072	0.059	0.047	0.041	0.064
Expand	0.059	0.042	0.036	0.042	0.034
Mean	0.090	0.053	0.055	0.043	0.039

In the simulations, we average the results from 10 independent runs to get one data point for sufficient statistical accuracy.

B. Small-Scale Simulations with Six-Node Topology

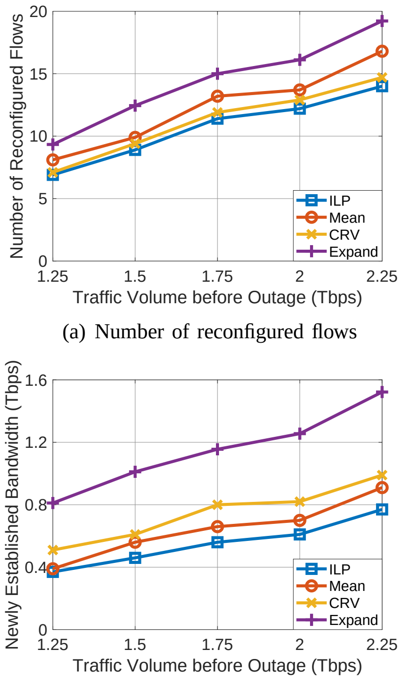
We first conduct small-scale simulations with the six-node topology to compare the heuristics with the ILP. Here, we only randomly generate one switch outage in each simulation, and select the peak rate of each flow within $[20, 80]$ Gbps, since each lightpath in the six-node topology provides a capacity of 100 Gbps. The simulation results are shown in Fig. 8. It can be seen that the algorithms' performance on the number of reconfigured flows (in Fig. 8(a)) is comparable. This is because for all the to-be-configured flows in $\mathbf{R^a} = \mathbf{R}' \cup \mathbf{R^f}$, the flows in $\mathbf{R^f}$ (interrupted by the switch outage) are generally many more than those in $\mathbf{R}'$ (selected ones to relieve future congestions), while the size of $\mathbf{R^f}$ is determined by the switch outage and cannot be optimized by the algorithms.

As expected, the ILP invokes the least reconfigurations in Fig. 8(a) to accomplish the XLry-O. CRV performs only slightly worst than the ILP but better than the two benchmarks, which suggests that CRV can fully explore traffic prediction to avoid unnecessary flow reconfigurations. Expand performs the worst among the algorithms on the number of reconfigured flows. In Fig. 8(b), the ILP also establishes the smallest amount of bandwidth for reconfiguring the flows. Among the four algorithms, Expand sets up the most new bandwidth, which is because it does not consider the relation among the congestions occurring on lightpaths between different switch pairs, and thus cannot achieve global optimization, *i.e.*, the newly established bandwidth from Expand can hardly be shared by the flows. CRV performs slightly worse than Mean, and both of them establish more bandwidth than the ILP. However, the results on average running time in Table I indicate that the heuristics are much more time efficient than the ILP, and CRV runs as fast as the two benchmarks.

C. Large-Scale Simulations with NSFNET Topology

We then perform large-scale dynamic simulations with the NSFNET topology to further evaluate the heuristics. We invoke 4 switch outages in each simulation. The interval between two consecutive outages is set as $1,000$ TS' (*i.e.*, each TS is 5 minutes), each simulation runs for $4,500$ TS'. The peak rate of each flow is randomly selected within $[4, 10]$ Gbps.

In the dynamic simulations, we first verify the effectiveness of our method to determine the number of the predicted samples in the DL-based traffic predictor (*i.e.*, L_o). Specifically, we simulate CRV by changing L_o from 72 to 360. The results



(a) Number of reconfigured flows

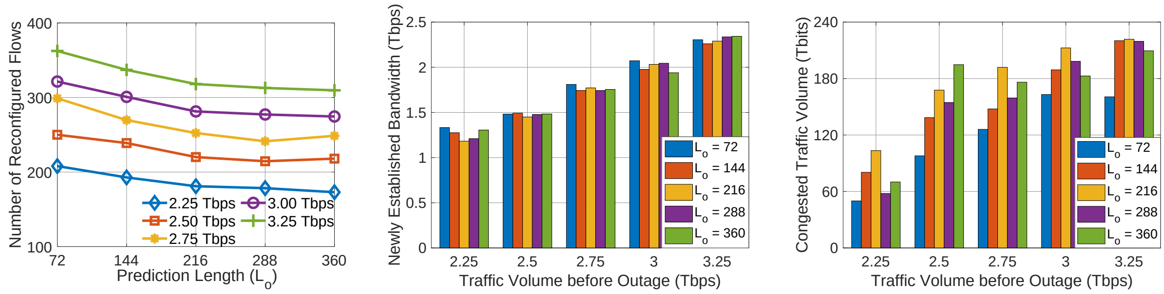
(b) Newly established bandwidth

Fig. 8. Results from simulations with six-node topology.

in Fig. 9(a) indicate that for all the traffic loads (*i.e.*, average traffic volume before each outage), increasing L_o can reduce the number of reconfigured flows and the reduction tends to converge when $L_o \geq 288$. This is because with a larger L_o, the XLyr-O system has more information regarding future traffic and thus can reconfigure the flows better to reduce overall reconfigurations. Meanwhile, the convergence of the reduction in Fig. 9(a) suggests that $L_o = 288$ (*i.e.*, a look-ahead period of 24 hours) is the proper choice, and thus the method designed in Section IV-B is effective. On the other hand, L_o does not significantly impact the newly established bandwidth in Fig. 9(b). This confirms the robustness of CRV, *i.e.*, no matter how many future samples are provided by the traffic predictor, it can fully utilize the bandwidth in the SD-IPoEON.

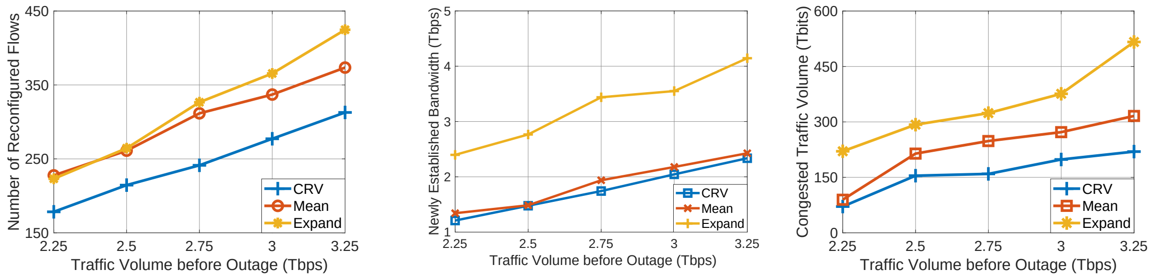
Note that, as 100% prediction accuracy is impossible, future congestions can still happen due to prediction errors. Hence, we plot the results on the congested traffic volume, which is the total volume of the traffic that cannot be delivered due to congestions, in Fig. 9(c). It is interesting to notice that the congested traffic volume generally increases with L_o before $L_o = 288$. This confirms our analysis in Section IV-B, *i.e.*, choosing L_o to cover the major frequency components in traffic fluctuation would help to balance the tradeoff between the length of look-ahead period and prediction error.

Next, we fix $L_o = 288$ and compare the performance of the heuristics. The results in Fig. 10(a) show that CRV invokes the least flow reconfigurations among the three algorithms, while Expand reconfigures comparable numbers of flows as Mean at low traffic loads but its number of reconfigured flows is the most when the traffic load is larger than 2.5 Tbps. This is because Expand can hardly achieve global optimization. Fig. 10(b) shows that Expand establishes the



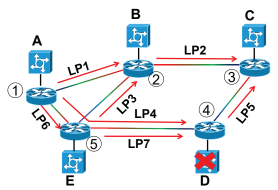
(a) Number of reconfigured flows (b) Newly established bandwidth (c) Congested traffic volume

Fig. 9. Simulation results with NSFNET topology (CRV with different L_o).



(a) Number of reconfigured flows (b) Newly established bandwidth (c) Congested traffic volume

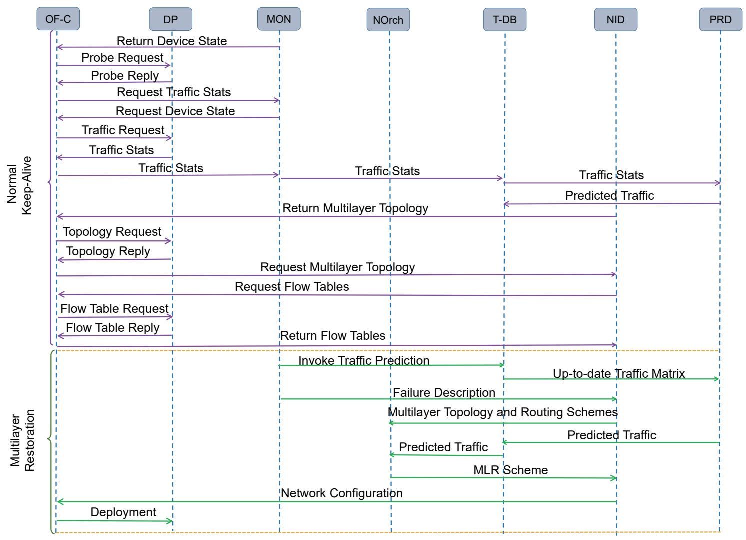
Fig. 10. Simulation results with NSFNET topology (different heuristics with $L_o = 288$).



(a)

Flow	IP Route	Lightpaths
F1	A->D->C	LP4, LP5
F2	E->D->C	LP7, LP5
F3	A->E->D->C	LP6 ,LP7, LP5
F4	E->B->C	LP3, LP2
F5	E->B	LP3
F6	A->B	LP1

(b)



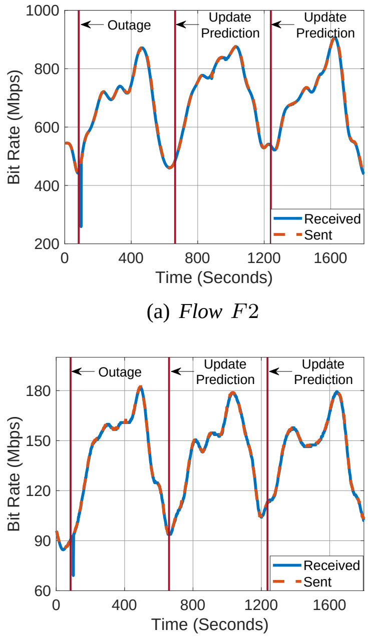
(c)

Fig. 11. Experimental setup and scenario, (a) topology of SD-IPoEON data plane, (b) initial flow routing schemes, and (c) operation procedure of DL-assisted XLyr-O, DP: Data plane.

most bandwidth to accommodate the flows in $\mathbf{R}$. As CRV and Mean use the same method to provision the flows, the newly established bandwidth from them is similar. However, Mean reconfigures many more flows than CRV in Fig. 10(a), which is because when Mean selects the flows to reconfigure, it chooses those with the most average predicted traffic but does not consider the actual traffic fluctuation. The results in Fig. 10(c) on the congested traffic volume also confirm the superior performance of CRV, since for all the traffic loads, its congested traffic volumes are the smallest. In all, since CRV considers future traffic fluctuation when selecting and provisioning the flows in $\mathbf{R}$, it reconfigures the least number of flows, induces the least congested traffic volume, with comparable newly established bandwidth as that from Mean.

VI. System Prototype and Experimental Demonstrations

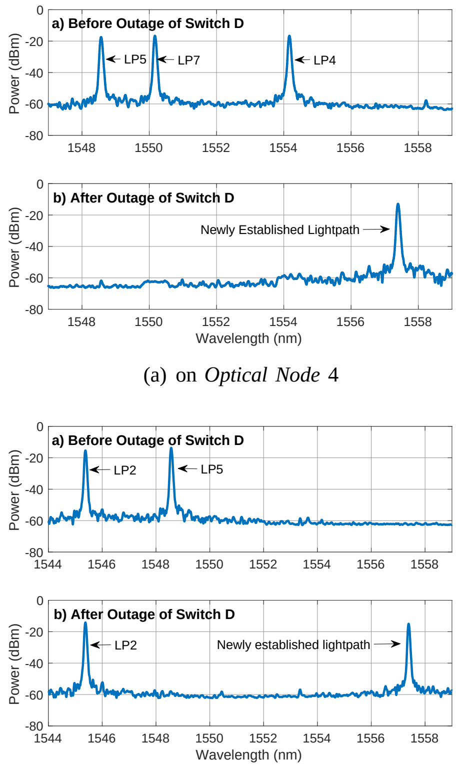
In this section, we implement CRV to prototype an SD-IPoEON system with DL-assisted XLyr-O, and demonstrate that it can utilize closed-loop operations to constantly deal with switch outages and lightpath congestions.



(a) *Flow* $F2$

(b) *Flow* $F3$

Fig. 12. Results on sending and receiving bandwidth.



(a) on *Optical Node* 4

(b) on *Optical Node* 3

Fig. 13. Optical spectrum measurements.

A. System Prototype

For the control plane, we realize CRV together with the DL-assisted XLyr-O in the open network operating system platform (ONOS). The extended ONOS, which follows our design in Fig. 2, runs on a Linux server and manages both the IP and EON layers. For simplicity, our experiments use a UDP-based IP flow to emulate an aggregated flow in a backbone SD-IPoEON. Specifically, we generate each UDP flow according to a series of traffic samples in the traffic data set in [43], where the flow's instant bandwidth demands are scaled within $[39, 910]$ Mbps and the sampling interval between two traffic samples (*i.e.*, T) is reduced to 2 seconds to shorten the time used for each experiment. Hence, MON also collects traffic statistics with a period of 2 seconds. It extracts real-time traffic statistics of each flow from OpenFlow statistics messages (*OFPC_FLOW_STATS*), and passes the information to PRD through T-DB. The traffic statistics of each flow is collected at its ingress switch to the SD-IPoEON. For instance, in Fig. 11(b), the ingress switch of $F1$ is Switch A. We implement NOrch as an ONOS application to get the XLyr-O schemes when hard/soft failures occur. Fig. 11(c) shows the interactions to realize the DL-assisted XLyr-O.

The data plane of the SD-IPoEON testbed has the topology in Fig. 11(a), where there are 5 nodes in the IP and EON layers, respectively. Each optical node in the EON layer is built with Finisar 1×9 BV-WSS', which can set up lightpaths within the wavelength range of $[1528.43, 1566.88]$ nm, by allocating the spectra on fiber links with a granularity of 12.5 GHz. Specifically, each BV-WSS is equipped with an OpenFlow agent (OF-AG) [11], which can communicate with OF-C using the OpenFlow protocol including optical transport protocol extensions (OF w/OTPE), parse the received *FlowMod* messages for lightpath management instructions, and configure the BV-WSS accordingly. The switches in the IP layer are Pica-8 switches that have 10GbE optical ports. Hence, a connection between the optical ports on two switches is essentially a lightpath. Our experiments limit the capacity of each lightpath as 1 Gbps, for emulating lightpath congestions.

B. Experimental Demonstrations

Fig. 11(b) shows the initial experimental scenario before the switch outage in the IP layer. Here, we have 6 UDP flows (*i.e.*, $\{F1, \cdots, F6\}$) routed by 5 switches (*i.e.*, $\{A, B, C, D, E\}$) that are interconnected by 7 lightpaths (*i.e.*,$\{LP1, \cdots, LP7\}$). In the experiment, a switch outage is first emulated by disconnecting *Switch* D. After the outage, *Lightpaths* $LP4$, $LP5$ and $LP7$ are disrupted to affect the packet transmission of *Flows* $F1$, $F2$ and $F3$. Then, the DL-assisted XLyr-O kicks in.

Fig. 12 shows the results on sending and receiving bandwidth of *Flows* $F2$ and $F3$. We can see that at the first time line (*i.e.*, when *Switch* D fails), both $F2$ and $F3$ experience packet loss since they go through the failed switch. Here, the packet loss of $F2$ is severer than that of $F3$. This is because $F2$ is recovered by setting up a new lightpath in the EON layer. Specifically, after it has detected the switch outage, the ONOS controller checks the predicted traffic provided by PRD and finds out that, if $F2$ is groomed onto $LP3\rightarrow LP2$, there would be severe congestion on $LP3$. Therefore, our XLyr-O decides to establish a new direct lightpath to recover $F2$. As $F2$ cannot be restored until the new lightpath has been established and optical reconfiguration takes much longer time than IP reconfiguration, the packet loss of $F2$ is severer.

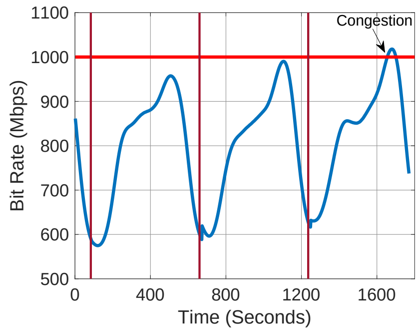


Fig. 14. Predicted total traffic on $LP3$

As for $F1$ and $F3$, our XLyr-O finds that there would be enough capacity on existing lightpaths in foreseeable future, to re-groom them for restoration. Hence, they are rerouted as $F1$: $LP1 \rightarrow LP2$ and $F3$: $LP6 \rightarrow LP3 \rightarrow LP2$. Meanwhile, since updating flow tables in switches takes much shorter time than reconfiguring the BV-WSS', the packet loss of $F3$ in Fig. 12(b) is much lighter than that of $F2$ in Fig. 12(a). Specifically, the recovery of $F2$ with a new direct lightpath takes 2.31 seconds and causes an instant packet loss rate[4] of 67.38%, while the recovery of $F1$ and $F3$ with electrical re-grooming only needs 0.26 second and the instant packet loss rate is 6.57%. Note that, the bandwidth results for $F1$ are similar to those for $F3$, and thus they are omitted to save space. Fig. 13 shows the optical spectrum measurements, which confirm that the outage brings down *Lightpaths* $LP4, LP5$ and $LP7$ while our XLyr-O sets up a new lightpath to restore $F2$ end-to-end.

Note that, the network reconfiguration at the first time line can only make sure that there is no traffic congestion before the second time line, when the traffic prediction provided by PRD will expire. Therefore, the DL-assisted XLyr-O kicks in again at the second and third time lines to check network status, and it finds out that the current configuration will lead to traffic congestion on $LP3$ after the third time line (in Fig. 14). This makes our system reroute flows proactively. Specifically, since there is no enough capacity on other existing lightpaths to re-groom any of the flows on $LP3$, our XLyr-O decides to set up a new lightpath for handling the bandwidth crunch. As the lightpath establishment is proactive and before the actual congestion on $LP3$ (*i.e.*, it is done with the "make-before-break" scheme [45]), we do not observe any noticeable packet loss on $F2$ and $F3$ in Fig. 12 during the reconfiguration.

For comparison, we conduct a benchmark experiment to evaluate the XLyr-O scheme without DL-assistance. Specifically, the difference between the benchmark and our proposal is that the benchmark does not leverage DL-based modules for traffic prediction and makes XLyr-O decisions only based on current network status. In other words, the benchmark is a reactive scheme that only reconfigures the SD-IPoEON testbed upon hard/soft failures. Fig. 15 shows the results on sending and receiving bandwidth of the flows. It can be seen that after the router outage, $F1$, $F2$ and $F3$ are recovered with IP reconfigurations. Based on the current network status, the benchmark decides to re-groom $F1$, $F2$ and $F3$ on the shortest available paths as $F1$: $LP1 \rightarrow LP2$, $F2$: $LP3 \rightarrow LP2$ and $F3$: $LP1 \rightarrow LP2$. This, unfortunately, will lead to traffic congestion on $LP2$ (*i.e.*, carrying $F1$-$F4$) soon, which explains the packet losses in Fig. 15 on $F1$-$F3$ shortly after they have been recovered from the outage. To address the congestion, the benchmark establishes a new direct lightpath to reroute $F2$, since it uses the most capacity of $LP2$ during the congestion. Later on, there will be another congestion on $LP1$ (*i.e.*, carrying $F1$, $F3$ and $F6$), and the benchmark sets up another direct lightpath to reroute $F1$. This is the reason why we observe severe packet losses on $F1$ in Fig. 15. The second congestion does not induce noticeable packet losses on $F3$ (*i.e.*, the severe packet losses on $F1$ is mainly due to the optical reconfiguration), and this is because the actual congestion is not severe and the throughput of $F3$ is much smaller than those of $F1$ and $F6$. To this end, we can see that without DL-assistance, the reactive benchmark not only invokes more reconfigurations but also suffers from severer packet losses.

[4]Each instant packet loss rate is measured over 4 seconds (*i.e.*, $2T$).

VII. Conclusion

In this paper, we studied the DL-assisted XLyr-O scheme from algorithm design to system prototype. A DL module based on LSTM-NN was first designed to capture the dynamics and self-similarity of end-to-end IP traffic for precise traffic prediction. Then, we considered the MLR in an SD-IPoEON as the usecase, and designed algorithms to explore the traffic prediction for realizing proactive XLyr-O. Finally, we implemented our proposed algorithm CRV in a small-scale but real SD-IPoEON testbed to prototype the DL-assisted XLyr-O, and demonstrated our proposal experimentally. Experimental results verified that compared with the reactive benchmark without DL-assistance, our proposal not only invoked less reconfigurations but also reduced packet losses significantly.

Acknowledgments

This work was supported by the NSFC projects 61871357 and 61701472, CAS key project (QYZDY-SSW-JSC003), and NGBWMCN key project (2017ZX03001019-004).

References

[1] Cisco visual networking index: Forecast and methodology, 2017-2022. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html#_Toc529314186

[2] P. Lu *et al.*, "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.

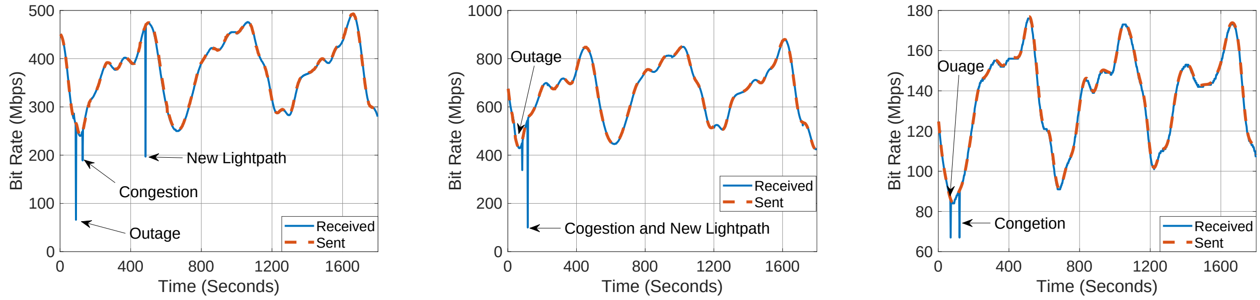
[3] O. Gerstel, M. Jinno, A. Lord, and S. Yoo, "Elastic optical networking: A new dawn for the optical layer?" *IEEE Commu. Mag.*, vol. 50, pp. s12–s20, Feb. 2012.

[4] 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.

[5] O. Gerstel *et al.*, "Multi-layer capacity planning for IP-optical networks," *IEEE Commun. Mag.*, vol. 52, pp. 44–51, Jan. 2014.

[6] 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.

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



(a) *Flow* $F1$ (b) *Flow* $F2$ (c) *Flow* $F3$

Fig. 15. Results on sending and receiving bandwidth from the benchmark experiment without DL-assistance.

[8] N. Sambo *et al.*, "Next generation sliceable bandwidth variable transponders," *IEEE Commun. Mag.*, vol. 53, pp. 163–171, Feb. 2015.

[9] S. Liu, W. Lu, and Z. Zhu, "Cost-efficient multi-layer restoration to address IP router outages in IP-over-EONs," in *Proc. of OFC 2017*, pp. 1–3, Mar. 2017.

[10] L. Liu *et al.*, "OpenSlice: an OpenFlow-based control plane for spectrum sliced elastic optical path networks," *Opt. Express*, vol. 21, pp. 4194–4204, Feb. 2013.

[11] Z. Zhu *et al.*, "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.

[12] I. Maor *et al.*, "First demonstration of SDN-controlled multi-layer restoration and its advantage over optical restoration," in *Proc. of ECOC 2016*, pp. 1–3, Sept. 2016.

[13] S. Liu, B. Li, and Z. Zhu, "Realizing AI-assisted multi-layer restoration in a software-defined IP-over-EON with deep learning: An experimental study," in *Proc. of OFC 2018*, pp. 1–3, Mar. 2018.

[14] P. Lu and Z. Zhu, "Data-oriented task scheduling in fixed- and flexible-grid multilayer inter-DC optical networks: A comparison study," *J. Lightw. Technol.*, vol. 35, pp. 5335–5346, Dec. 2017.

[15] S. Zhang *et al.*, "Evolving traffic grooming in multi-layer flexible-grid optical networks with software-defined elasticity," *J. Lightw. Technol.*, vol. 32, pp. 2905–2914, Aug. 2014.

[16] W. Lu, X. Yin, X. Cheng, and Z. Zhu, "On cost-efficient integrated multilayer protection planning in IP-over-EONs," *J. Lightw. Technol.*, vol. 35, pp. 5335–5346, Dec. 2017.

[17] S. Bhattacharyya, C. Diot, and J. Jetcheva, "Pop-level and access-link-level traffic dynamics in a Tier-1 POP," in *Proc. of ACM SIGCOMM IMW 2001*, pp. 39–53, Nov. 2001.

[18] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez, "Inter-datacenter bulk transfers with netstitcher," in *Proc. of ACM SIGCOMM 2011*, pp. 74–85, Aug. 2011.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, Dec. 1997.

[20] 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.

[21] Y. Yin *et al.*, "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.

[22] 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.

[23] L. Gong *et al.*, "Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. 836–847, Aug. 2013.

[24] X. Liu, L. Gong, and Z. Zhu, "On the spectrum-efficient overlay multicast in elastic optical networks built with multicast-incapable switches," *IEEE Commun. Lett.*, vol. 17, pp. 1860–1863, Sept. 2013.

[25] L. Yang *et al.*, "Leveraging light forest with rateless network coding to design efficient all-optical multicast schemes for elastic optical networks," *J. Lightw. Technol.*, vol. 33, pp. 3945–3955, Sept. 2015.

[26] Z. Zhu *et al.*, "Impairment- and splitting-aware cloud-ready multicast provisioning in elastic optical networks," *IEEE/ACM Trans. Netw.*, vol. 25, pp. 1220–1234, Apr. 2017.

[27] L. Gong, Y. Wen, Z. Zhu, and T. Lee, "Toward profit-seeking virtual network embedding algorithm via global resource capacity," in *Proc. of INFOCOM 2014*, pp. 1–9, Apr. 2014.

[28] Y. Sone *et al.*, "Bandwidth squeezed restoration in spectrum-sliced elastic optical path networks (SLICE)," *J. Opt. Commun. Netw.*, vol. 3, pp. 223–233, Mar. 2012.

[29] M. Liu, M. Tornatore, and B. Mukherjee, "Survivable traffic grooming in elastic optical networks - shared protection," *J. Lightw. Technol.*, vol. 31, pp. 903–909, 2013.

[30] F. Ji *et al.*, "Dynamic p-cycle protection in spectrum-sliced elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 1190–1199, Mar. 2014.

[31] X. Chen, F. Ji, and Z. Zhu, "Service availability oriented p-cycle protection design in elastic optical networks," *J. Opt. Commun. Netw.*, vol. 6, pp. 901–910, Oct. 2014.

[32] M. Ruiz *et al.*, "Survivable IP/MPLS-over-WSON multilayer network optimization," *J. Opt. Commun. Netw.*, no. 8, pp. 629–640, Mar. 2011.

[33] W. Kmiecik, R. Goscien, K. Walkowiak, and M. Klinkowski, "Two-layer optimization of survivable overlay multicasting in elastic optical networks," *Opt. Switch. Netw.*, vol. 14, pp. 164–178, Aug. 2014.

[34] A. Mayoral *et al.*, "Minimizing resource protection in IP over WDM networks: Multi-layer shared backup router," *J. Opt. Commun. Netw.*, vol. 7, pp. A440–A446, Mar. 2015.

[35] R. Govindan *et al.*, "Evolve or die: High-availability design principles drawn from Google's network infrastructure," in *Proc. of ACM SIGCOMM 2016*, pp. 58–72, Aug. 2016.

[36] C. Lai *et al.*, "Cross-layer proactive packet protection switching," *J. Opt. Commun. Netw.*, vol. 4, pp. 847–857, Oct. 2012.

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

[38] A. Felix *et al.*, "Multi-layer SDN on a commercial network control platform for packet optical networks," in *Proc. of OFC 2014*, pp. 1–3, Mar. 2014.

[39] D. Park, "Structure optimization of BiLinear Recurrent Neural Networks and its application to Ethernet network traffic prediction," *Inf. Sci.*, vol. 237, pp. 18–28, Jul. 2013.

[40] F. Morales *et al.*, "Virtual network topology adaptability based on data analytics for traffic prediction," *J. Opt. Commun. Netw.*, vol. 9, pp. A35–A45, Jan. 2017.

[41] 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.

[42] J. Guo and Z. Zhu, "When deep learning meets inter-datacenter optical network management: Advantages and vulnerabilities," *J. Lightw. Technol.*, vol. 36, pp. 4761–4773, Oct. 2018.

[43] S. Liu and Z. Zhu, "Generating data sets to emulate dynamic traffic in a backbone IP over optical network," *Tech. Rep.*, 2019. [Online]. Available: https://github.com/lsq93325/Traffic-creation/blob/master/README.md?tdsourcetag=s_pctim_aiomsg

[44] A. Klekamp, U. Gebhard, and F. Ilchmann, "Efficiency of adaptive and mixed-line-rate IP over DWDM networks regarding CAPEX and power consumption," *J. Opt. Commun. Netw.*, vol. 4, pp. B11–B16, Nov. 2012.

[45] M. Zhang, C. You, and Z. Zhu, "On the parallelization of spectrum defragmentation reconfigurations in elastic optical networks," *IEEE/ACM Trans. Netw.*, vol. 24, pp. 2819–2833, Oct. 2016.