

Deep Reinforcement Learning-aided Fragmentation-aware RMSA Path Computation Engine for Open Disaggregated Transport Networks

1st Javier Errea
Nokia Bell Labs Massy, France
javier.errea_moreno@nokia.com

2nd Deborah Djon
DHBW Stuttgart, Germany
deborah.djon@gmail.com

3rd Huy Quang Tran
Nokia Bell Labs Massy, France
huy.h.tran@nokia.com

4th Dominique Verchere
Nokia Bell Labs Massy, France
dominique.verchere@nokia-bell-labs.com

5th Adlen Ksentini
EURECOM Biot, France
adlen.ksentini@eurecom.fr

Abstract—Optical network control platforms are facing an unprecedented increase of complexity due to the requirements for openness and optical system dis-aggregations. While Reinforcement Learning (RL) and Deep RL (DRL) methods can be used for network control decision-configuration solutions, these methods can be unsuitable to find correct policies to control open disaggregated transport networks (ODTN). In this context, this article proposes DeepSF-PCE, a single-agent deep reinforcement learning spectrum fragmentation aware path computation engine to solve the Routing, Modulation and Spectrum Assignment (RMSA) problems in ODTN. DeepSF-PCE engine learns fragmentation-aware policies that maximize the number of allocated service requests. Simulation results show that DeepSF-PCE can increase by more than 40% the number of configured optical connectivity services compared to existing solutions.

Index Terms—RMSA, DRL, SDN, ODTN.

I. INTRODUCTION

6G requires stringent connectivity services for the far edges, where future transport networks will have to play a new role in enabling end-to-end optical connectivity services. To guarantee critical requirements between network edges, transport network control platforms must break through to enable optimal end-to-end optical connections as connectivity services provisioned over open, disaggregated, i.e., multi-vendor, optical network infrastructures. Moreover, optical-switched network controllers will have to coordinate more with packet-switched network controllers and to move jointly IP-over-WDM network control to the edges with the integration of the next-generation coherent optical terminal modules [1].

However, the convergence of IP-Optical networks and Elastic Optical Networks (EONs) will increase drastically the complexity of transport network service provisioning. This convergence requires the coordinated control of optical channel connections, i.e., Quality of Transmission (QoT), and IP flows that interconnect the far edges. More re-configurable optical network systems are required to adapt with the changes of the demands of 6G services. Dynamic transport connectivity services will be provisioned by unified Software Defined

Network (SDN) Control platforms to configure the required resources on demand. Therefore, design, planning, and resource allocation in optical networks need to be reconsidered.

There has been a lot of proposals in applying RL and DRL for optical network control and management. The Routing, Modulation format and Spectrum Assignment (RMSA) problem is one of the fundamental challenges in EONs. The objective is to select the optimal optical channel paths between a source and destination as well as the frequency slots (FS) while ensuring spectrum continuity, contiguity, and non-overlapping constraints in a dynamic network environment. Per [7], Deep Reinforcement Learning (DRL) has been able to solve a wide range of complex decision-making tasks. Subsequently, DRL is considered a good candidate to solve the RMSA problem. In [2], a DRL framework for RMSA in EONs is described to reduce the blocking ratio of optical channel connection requests by more than 20,3% in the NSFNET topology and 14,3% in COST239 topology. However, DeepRMSA best performance is obtained for one candidate FS block per k-path and this is similar to first-fit (FF) spectrum allocation. For cases in which multiple candidate paths can be considered for spectrum allocation, DeepRMSA starts experimenting spectrum fragmentation (SF) issues. SF is a well-known problem in EONs, where frequent configuration changes derive in higher SF and subsequently, it significantly increases the service blocking probability [8].

In this paper, we propose a new DRL-assisted SF-aware Path Computation Engine (DeepSF-PCE) for Cloud-native SDN platforms to control open and fully disaggregated optical networks as in [5]. DeepSF-PCE performances were evaluated with NSFNET topology and the simulation results verify its superiority with respect to existing RMSA algorithms.

II. DEEPSF-PCE SYSTEM ARCHITECTURE & DESIGN

A. Transport Control Architecture

The software architecture shown in Fig. 1 indicates how DeepSF-PCE can be deployed as a network control func-

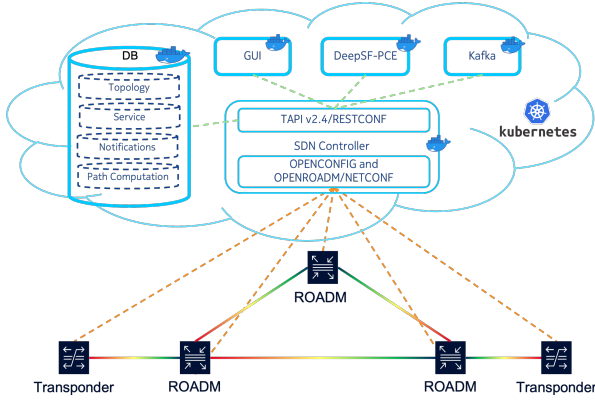


Fig. 1: System Architecture

tion that interfaces with Transport SDN controller through Transport-API [11] standard interface. Upon reception of a connectivity service request through the Graphical User Interface (GUI), the SDN controller retrieves the current network state from its database and sends a path computation request to DeepSF-PCE. Then, DeepSF-PCE provides an RMSA scheme for the SDN controller. The controller attempts to provision the corresponding connectivity service along the computed path using NETCONF protocol with open device data models (e.g., OpenConfig [12]). The deployment of a Kafka broker enables to follow the service provisioning state by subscribing to the corresponding Kafka topic.

The objective of DeepSF-PCE is to learn optimal RMSA policies that maximize the number of served requests.

B. DeepSF-PCE Design

DeepSF-PCE involves three components: the network state (i.e., the observation vector), the set of actions that can be performed by the DRL-agent (i.e., action space), and the reward function that gives feedback to the agent.

- The **action space**, A is a set of discrete actions of size $k \times J + 1$, where k is the number of candidate paths for each source-destination pair in the network and J determines the number of candidate FS blocks in each path. The additional action represents the possibility of rejecting a request.
- The **observation vector**, o_t contains the source-destination pair in 1-hot encoding (s, d), a normalized value of the service capacity requirement (c in Gbps), the service duration (τ) and the spectrum utilization of k shortest paths (SP) between source and destination. For each path, we compute the starting indices $x_{1,j}^k$, the size of the FS block $x_{2,j}^k$, the resulting fragmentation state $x_{3,j}^k$ in case of selecting that particular FS block, the number of FS needed x_4^k and the average FS block size x_5^k . For calculating the fragmentation state, we use the Shannon entropy (H_{frag}) at two different levels: link-based (LB) and path-based (PB), as proposed in [6]. The objective is to provide to the agent the resulting spectrum fragmentation measure after allocating the service

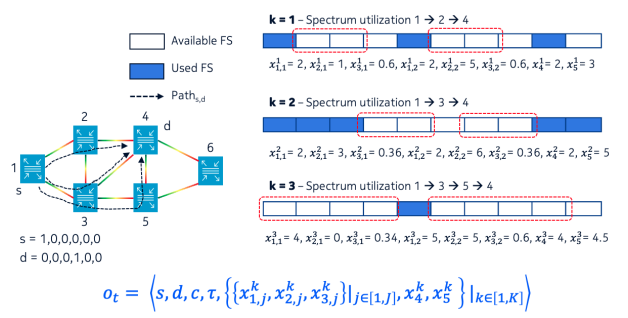


Fig. 2: DeepSF-PCE state representation

request. Higher level of SF lead to large values of H_{frag} , therefore we normalize H_{frag} , between [0-1]: $(e^{-H_{frag}})$. Thus, a common scale is preserved for every parameter in the observation vector. Fig. 2 shows an example of the state representation using the PB fragmentation approach.

- The **reward function** considers the resource allocation capability of the agent for a given request and the resulting fragmentation caused by the allocation. If the agent can accommodate the service request, it will get a positive reward $+1$ plus an additional fragmentation-aware metric. If not, it will receive a negative reward -1 . For that reason, we introduce the difference in fragmentation (ΔH_{frag}) between t and $t - 1$. The reward function is defined as follows:

$$r_t = \begin{cases} 1 + e^{-\Delta H_{frag}} & \text{successfully allocated} \\ -1 & \text{otherwise} \end{cases} \quad (1)$$

III. EXPERIMENTAL SET UP AND PERFORMANCE EVALUATION

A. Experimental Set up

The DRL agent was trained using two algorithms: Advantage Actor-Critic (A2C) and Proximal Policy Optimization (PPO) provided by the stable-baselines library [3]. Both agents are modeled with two fully-connected Deep Neural Networks (DNNs): Actor and Critic. The input layer size is equal to the length of o_t . 5 hidden layers, each one of 128 neurons and the output layer consisting of $k \times J$ neurons for the Actor network and 1 neuron for the Critic network. The difference between A2C and PPO lies on the learning phase. PPO estimates the policy gradients using the ratio between the new and old policy instead of using the logarithm of the new policy as in A2C [10]. We selected a discounted factor $\gamma = 0.96$ and a learning rate $\alpha = 10^{-4}$ as they produced the best results. The simulations consider an extended dynamic network environment from [9], where service requests arrive following a Poisson process with arrival rate $\lambda = 10$ and a mean service duration of 25 units of time. The source-destination pair is randomly selected, and the capacity demand follows a uniform distribution between [10-200] Gbps. The C-band is considered for each link of the network topology (384 FSs of 12.5GHz). The distance-adaptive based impairment aware

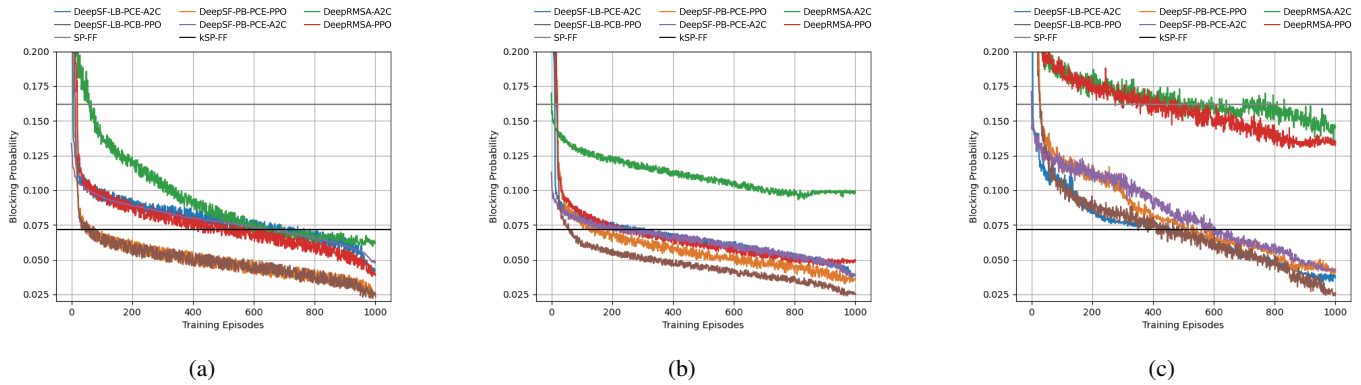


Fig. 3: Blocking probability for $J = 2, 3, 5$

model explained in [4] is used to select the modulation format and, subsequently, determine the number of FS needed to accommodate the service request. The agents were trained for 1000 episodes, where each episode consisted of 1000 service requests. An episode begins with a fully available spectrum, and ends when all service requests have been processed i.e., provisioned or rejected, by the agent. The objective is to prove the benefit of including spectrum fragmentation related information when quantifying the service blocking probability.

B. Performance evaluation

The performance of DeepSF-PCE is evaluated using NSFNET topology (14 nodes, 22 links). We set $k = 5$ and consider three different values of $J = 2, 3, 5$.

Fig. 3 illustrates the results in terms of request blocking probability. As the number of candidate spectrum allocation schemes increases, it is essential that the agent selects the solution (path, modulation format and FS block) that will maximize the system expected cumulative reward, rather than the immediate reward. By adding the network fragmentation state to the agent observation, DeepSF-PCE is able to learn improved RMSA policies. DeepSF-PCE algorithm outperforms DeepRMSA, SP-FF and kSP-FF by accepting more requests at the end of training. It reduced the blocking probability by 28.5%, 50% and 80.7% for $J = 2, 3$ and 5 respectively, as compared to DeepRMSA. These results demonstrate the benefit of adding the link/path fragmentation state in the observation vector and the reward function. However, learning good policies is more time-expensive for higher values of J , since both o_t and A become larger. Therefore, more training episodes are needed for $J = 5$ (Fig. 3c) to reduce the blocking probability as for $J = 2, 3$ (Fig. 3a and Fig 3b). In addition, DeepSF-LB outperforms DeepSF-PB for higher values of J . The PB approach captures the end-to-end spectral voids, and does not have the information regarding to free FS blocks that are only available on some of the links along a path. Lastly, PPO achieves lower blocking probability than A2C when comparing any results as it relies on specialized clipping in the objective function (expected cumulative reward). This

technique accelerates learning and reduces the impact of extreme observations making the model more robust.

IV. CONCLUSION

A Fragmentation-aware Path Computation Engine based on DRL was described with its performance reducing the blocking probability with respect to traditional RMSA techniques and DRL methods. It proves that spectrum fragmentation aware policies can be defined and learned efficiently to maximize the spectrum grid utilization. A cooperative Multi-Agent approach of DeepSF-PCE for service provisioning over a multi-domain optical network is left for future studies.

REFERENCES

- [1] M. Thottan, G. Atkinson, T.V. Lakshman, D. Verchere, "Packet/Optical Networks and SDN Control at the Edge," in *Optical access networks for fixed and mobile services of Optical Fiber Communication Conf.*, OFC'2021-F41.1
- [2] X. Chen et. al., "DeepRMSA: A Deep Reinforcement Learning Framework for Routing, Modulation and Spectrum Assignment in Elastic Optical Networks," in *Journal of Lightwave Technology*, vol. 37, no. 16, pp. 4155-4163, 15 Aug.15, 2019
- [3] A. Raffin et. al., "Stable-Baselines3: Reliable Reinforcement Learning Implementations," in *2021 Journal of Machine Learning Research*
- [4] B. Kozicki, H. Takara, Y. Sone, A. Watanabe and M. Jinno, "Distance-adaptive spectrum allocation in elastic optical path network (SLICE) with bit per symbol adjustment" in *2010 Conference on Optical Fiber Communication (OFC)*
- [5] Q. Pham-Van et. al., "Demonstration of Container-based Microservices SDN Control platform for Open Optical Networks," in *Optical Fiber Communication Conference (OFC) 2019, M3Z.5*
- [6] P. Wrigth, M. C. Parker and A. Lord, "Simulation results of Shannon entropy based flexgrid routing and spectrum assignment on a real network topology" in *39th European Conference and Exhibition on Optical Communication (ECOC 2013)*
- [7] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare and J. Pineau (2018), "An Introduction to Deep Reinforcement Learning", in *Foundations and Trends in Machine Learning: Vol. 11: No. 3-4*.
- [8] D. Amar et. al., "Spectrum fragmentation issue in flexible optical networks: analysis and good practices", in *Photonic Network Communications* 29, 230-243 (2015)
- [9] C. Natalino and P. Monti, "The Optical RL-Gym: An open-source toolkit for applying reinforcement learning in optical networks," in *2020 22nd International Conference on Transparent Optical Networks (ICTON)*
- [10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," 2017. [Online]. Available: arxiv:1707.06347
- [11] ONF Transport API, [online] Available: <https://github.com/OpenNetworkingFoundation/TAPI>
- [12] OpenConfig, [online] Available: <https://github.com/openconfig/public>