2024 IEEE 30th International Symposium on Local and Metropolitan Area Networks (LANMAN) | 979-8-3503-5209-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/LANMAN61958.2024.10621885

Dynamic Crosstalk-Aware Routing, Modulation, Core, and Spectrum Allocation for Sliceable Demands in SDM-EONs

Arash Rezaee† arash_rezaee@student.uml.edu
Ryan McCann† ryan_mccann@student.uml.edu
Vinod M. Vokkarane† vinod_vokkarane@uml.edu
†Electrical and Computer Engineering Department, University of Massachusetts Lowell, United States

***Abstract*—Elastic optical networks (EONs) will not be able to satisfy the ever-increasing demand of the next-generation Internet applications. Thus, space-division multiplexing (SDM) technology is introduced to increase the fiber capacity and incorporate multiple EONs (SDM-EONs), specifically through implementing multi-core fibers (MCFs). Inter-core crosstalk (XT) is the fundamental issue in MCF, leading to lower utilization and lower optical signal quality. Due to the presence of significant XT, the traditional resource allocation problem in SDM-EONs needs to incorporate dynamic XT values. This paper describes a comprehensive solution that incorporates the XT constraint during routing, modulation, core, and spectrum allocation in the software-defined networking controller. First, we present a dynamic XT-aware routing (XTAR) algorithm with two policies in which link costs are dynamically calculated based on the XT effect and length of the links. Next, we introduce a new XT-aware bandwidth-slicing resource allocation approach that considers both XT limitations and simultaneously addresses blocking of large demands due to fragmentation and optical reach. Extensive simulations on several well-known network topologies reveal that the average request blocking for both the proposed policies significantly outperforms traditional shortest-path based benchmark.**

***Index Terms*—EON, SDM, SDM-EON, Slice-ability, RMCSA, SDN, Inter-core Crosstalk (XT)**

I. Introduction

The demand for Internet services has been increasing dramatically since the 90s, and due to digital technology advancement, it is projected to continue in the coming decades. Due to this demand explosion, elastic optical networks (EONs) were introduced to overcome inefficient resource allocation in fixed-grid optical networks and use higher transmission rates [1]. With traditional single-core fiber in EONs nearing physical capacity limits, space division multiplexing (SDM) is introduced to leverage the fiber bandwidth [2]. SDM technology is capable of multiplexing the orthogonal EON signals in the space domain (SDM-EONs) and provides multi-core fibers (MCFs) with EONs (MCF-EONs) [3].

The most critical challenge of EONs is dynamic resource allocation, which satisfies the continuity and contiguity constraints [4]. By enabling the space domain, core allocation is added to the resource allocation problem of EONs and extended to routing, modulation format, core, and spectrum assignment (RMCSA). RMCSA should satisfy the spatial or space continuity constraint, which means the same core should be allocated in all links of a lightpath. Although MCF increases the transmission capacity, inter-core crosstalk (XT) is considered a potential drawback of MCF, which is a limitation on spectrum utilization and quality of transmission (QoT). So, during the RMCSA steps, the XT threshold should be considered to meet the QoT constraint [5].

Several research groups are addressing the XT-aware RMCSA problem. In [6], K-shortest paths (KSPs) are calculated, and between available spectrum blocks of KSPs, the path and block with minimum XT and nonlinear impairments that meet the minimum signal-to-noise ratio (SNR) are selected. Worst-case and precise scenario estimations of XT are evaluated for dynamic lightpath provisioning in [7]. Authors in [8] and [9] prioritized KSPs by considering three factors, XT-limitation, spectrum usage, and fragmentation, then selected the available slice window according to the presented factors and optical transmission reach (OTR) imposed by amplified spontaneous emission (ASE) noise. Machine learning (ML) approaches are considered to improve choosing modulation and obtain a favorable trade-off between spectrum usage and XT tolerance [10], [11]. Also, the SDM sliceable approach is presented to relax the contiguity constraint and reduce the consequence of fragmentation and OTR limitation by dividing the requested bandwidth into smaller segments [12].

This research presents the dynamic XT-aware routing algorithm and a precise XT-aware bandwidth-slicing modulation, core, and spectrum assignment (MCSA) framework. To the best of our knowledge, this is the first instance where link costs are dynamically calculated based on network state and XT effect. Also, taking into account a precise XT calculation and prioritized core allocation enhances the sliceable MCSA algorithm. Regarding all the previous works that used KSPs, we consider KSPs with $K = 3$ as a baseline in this work.

The rest of the paper is organized as follows: Section II describes the physical layer model. The proposed algorithms are detailed in section III. The simulation results are evaluated in section IV. Section V presents the complexity analysis. Finally, section VI concludes the paper.

II. Physical Layer Model

The propagated signal in MCF experiences different types of noises that degrade its quality. This paper considers two crucial types of these noises, ASE and XT, during the RMCSA process in the software-defined networking (SDN) controller.

A. Amplified Spontaneous Emission Noise

In transparent networks, the optical amplifier amplifies the optical signal at the end of every span. This paper considers an

This paper was partially supported by NSF project award # 2008530.

TABLE I: SNR & XT thresholds for each modulation (in dB)

Threshold	QPSK	16-QAM	64-QAM
SNR [14]	8.5	14	20
XT [3]	-18.5	-25	-34

Erbium-doped fiber amplifier (EDFA) to reamplify the optical signal over a span length of 100 km. EDFAs add ASE noise to the optical signal. To guarantee the required bit error rate (BER) at the receiver, the signal-to-noise ratio (SNR) should be higher than a threshold that is determined based on the modulation format (Table I). Thus, the maximum distance that can satisfy the given SNR threshold (SNR_{min}) is calculated by:

$$L_{\text{max,SNR}} = \frac{P_s \cdot L_{\text{span}}}{SNR_{\text{min}} \cdot h \cdot f \cdot G \cdot NF \cdot R_s}, \quad (1)$$

where P_s, L_{span}, h, f, G, NF, and R_s, are launch power, span length, Plank's constant, optical signal frequency, amplifier gain, amplifier noise figure and symbol rate, respectively. According to the Equation 1, the OTR for all modulation levels is calculated (Table II) for 5.5 dB noise figure, 193.41 THz optical signal frequency, 20 dB amplifier gain, 100 km span length, and 0 dBm launch power. Also, the symbol rates are calculated for polarization multiplexing with 20%-overhead forward-error correction (FEC). The number of required slots for every modulation format is computed in Table II, assuming the mentioned overhead, dual-polarization technique, and a 12.5 GHz slot bandwidth. The OTRs are computed by considering a 4 dB margin for the required SNR thresholds (Table I) for BER of 10^{-2} [13].

TABLE II: Number of required slots & optical transmission reach (in km, based on ASE noise with 4 dB margin)

Modulation Format	QPSK		16-QAM		64-QAM	
Bit Rate	Slots	OTR	Slots	OTR	Slots	OTR
50 *Gbps*	2	10380	1	4648	1	1752
100 *Gbps*	3	5190	2	2324	1	876
200 *Gbps*	5	2595	3	1162	2	438
400 *Gbps*	10	1298	5	581	4	219

B. Inter-Core Crosstalk

The propagated signal in MCF suffers from the XT noise, which depends on the distance between cores and the frequency. For simplicity, the transmission signal is considered flat. An evanescent wave between the signals turns out to create XT and has an inverse and exponential relation with distance. For this reason, most parts of the XT effect arises from the overlapping frequency slots on neighbor cores. The XT between two cores of link e for slot s is calculated by:

$$xt_e^{c,c'}(s) = \frac{1 - \exp\left(-2 \cdot h \cdot L\left(e\right)\right)}{1 + \exp\left(-2 \cdot h \cdot L\left(e\right)\right)}, \quad h = \frac{2\kappa^2 r}{\beta \omega_{tr}}, \quad (2)$$

where $L\left(e\right)$, h, κ, r, β, and ω_{tr} represent link length, power coupling coefficient, coupling coefficient, bending radius, propagation constant, and core pitch, respectively. Equation 3 calculates the XT between core c and its active adjacent cores along the $path$ on slot s, so that $A_{c,e}$ is the list of active adjacent cores of core c on link e [5] and [15].

$$XT\left(path, c, s\right) = \sum_{e \in path} \sum_{c' \in A_{c,e}} xt_e^{c,c'}(s) \quad (3)$$

III. Proposed Crosstalk-Aware Algorithms

To overcome the XT barrier, dynamic XT-aware routing (XTAR) and the XT-aware sliceable MCSA (XTA-SMCSA) algorithms are presented in this section. The SDM-EON resource allocation in this paper is divided into two parts:

A. Routing

In this subsection, we introduce the XTAR method to minimize the XT effect in the dynamic route calculation step. As mentioned in section II, XT is caused by only active slots of adjacent cores; based on this assumption, XTAR attempts to use the links that have available slots with a minimum number of active slots on adjacent cores to minimize XT.

Algorithm 1 details the pseudo-code of the XTAR method. The available spectrum slots vector (ASV) is calculated for every core of the input graph links (Lines 2-3). If free spectrum slots exist on link e, the crosstalk cost of the link (XTC_e) is computed. XTC_e is equal to the summation of the number of active slots on adjacent cores divided by the number of adjacent cores for every available slot (Lines 4-5). Also, the number of available slots in whole cores of the link e is assigned to NAS_e (Line 6). Then, according to the selected policy, the weight of the link is calculated (Lines 7-9). The link weight is considered infinite if the link doesn't have an available slot on all cores (Line 11). Finally, if the weight of all links is equal to infinite, the request is blocked; otherwise, the minimum cost path, based on calculated weights, is computed by the Dijkstra algorithm (Lines 14-19). Policy I (π_1) considers XT cost and the normalized length of the link in a weighted manner in link cost. However, Policy II (π_2), by multiplying the XT cost by the number of link spans, tries to consider the effect of length on the link's weight indirectly.

Fig. 1 demonstrates the illustrative example of the XTAR method to calculate link e's weight. Figure 1 (a) shows the spectrum usage of link e on different cores. White and gray squares indicate available and occupied slots, respectively. Link e is equipped with a fiber with seven cores and four spectrum slots per core. And Fig. 1 (b) illustrates the arrangement of cores within the fiber. Cores 1, 2, and 7 only have available spectrum slots. The number of adjacent cores (NC_c) for the mentioned cores are equal to 3, 3, and 6, respectively. Available spectrum slots for these cores ($ASV_{c,e}$) are $ASV_{1,e} = \{2\}$, $ASV_{2,e} = \{2\}$, and $ASV_{7,e} = \{3, 4\}$ (Lines 2-3). The number of overlapped active slots on adjacent cores of available slots ($NOAS_{s,c}$) are $NOAS_{2,1} = 2$, $NOAS_{2,2} = 2$, $NOAS_{3,7} = 6$ and $NOAS_{4,7} = 6$. So, XT cost for link e is calculated by (Line 5):

$$XTC_e = \frac{NOAS_{2,1}}{NC_1} + \frac{NOAS_{2,2}}{NC_2} + \frac{NOAS_{3,7}}{NC_7} + \frac{NOAS_{4,7}}{NC_7},$$

$$XTC_e = \frac{2}{3} + \frac{2}{3} + \frac{6}{6} + \frac{6}{6} = \frac{10}{3}.$$

The number of available slots on link e (NAS_e) is equal to 4 (Line 6). So, the XT cost that is considered in link weight is $\frac{XTC_e}{NAS_e} = \frac{10}{12}$, which is always lower than 1. If link length (l_e) and the longest link in the network (l_{max}) are equal to 800 km and 1000 km, respectively. Link weight (W_e) for Policy I (Line 8) is:

Algorithm 1 Crosstalk-Aware Routing (XTAR)

Inputs:

(1) $G(V, E)$: Network topology, V and E are set of network nodes and links
(2) $R_i(s_i, d_i)$: Request's source and destination nodes
(3) ℓ_e: Length of link e
(4) $\ell_{\max}$: Length of the longest network link
(5) $N_{s,e}$: Number of spans in link e
(6) α: Weight of link length in link costs
(7) π: Selected policy

Parameters:

(1) C_e: Number of cores in link e
(2) $ASV_{c,e}$: Available spectrum slot vector of core c, link e
(3) ASV_e: Available spectrum slot vector of link e
(4) NC_c: Number of adjacent cores for core c
(5) $NOAS_{s,c}$: Number of overlapped active slots on adjacent cores of slot s on core c
(6) XTC_e: Crosstalk cost of link e
(7) NAS_e: Number of available slots on link e
(8) W_e: Weight of link e

Output:

(1) $Path$: Minimum weight path

Procedure Routing

1: **for** (e in E) **do**
2: Calculate $ASV_{c,e}$ for all cores of link e
3: $ASV_e = \begin{bmatrix} ASV_{1,e} \\ ASV_{2,e} \\ \vdots \\ ASV_{C_e,e} \end{bmatrix}$
4: **if** $ASV_e \neq \emptyset$ **then**
5: $XTC_e = \sum_{c\in[1,C_e]} \sum_{s\in ASV_{c,e}} \frac{NOAS_{s,c}}{NC_c}$
6: $NAS_e = \sum_{ASV_{c,e}\in ASV_e} len(ASV_{c,e})$
7: Calculate link weight based on selected policy (π):
8: **Policy I** (π_1)**:** $W_e = \alpha\times(\frac{\ell_e}{\ell_{max}})+(1-\alpha)\times(\frac{XTC_e}{NAS_e})$
9: **Policy II** (π_2)**:** $W_e = N_{s,e} \times (\frac{XTC_e}{NAS_e})$
10: **else**
11: $W_e = \infty$
12: **end if**
13: **end for**
14: **if** All $W_e = \infty$ **then**
15: The request is blocked.
16: **else**
17: $Path =$ The shortest path between nodes s_i and d_i in
18: graph G based on calculated links weight.
19: **end if**

$$W_e = \alpha \times \frac{l_e}{l_{max}} + (1-\alpha)\frac{XTC_e}{NAS_e} = \alpha \times \frac{4}{5} + (1-\alpha) \times \frac{10}{12}.$$

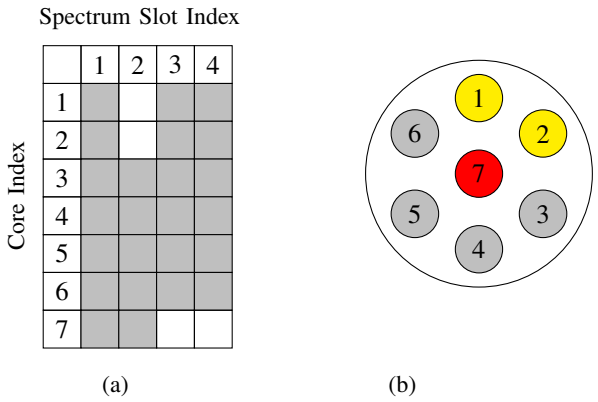

Fig. 1: Illustrative example of XT cost calculation for XTAR algorithm, (a) Spectrum Utilization, (b) Cores Location.

Assuming a span length of 100 km, the spans' number on link e ($N_{s,e}$) is 8. The link weight for Policy II (Line 9) is:

$$W_e = N_{s,e} \times \frac{XTC_e}{NAS_e} = 8 \times \frac{10}{12} = \frac{20}{3}.$$

B. Modulation, Core and Spectrum Assignment

When a path (or set of paths in the benchmark) is selected, it is time for MCSA. This paper considers an SDM sliceable framework to relax contiguity constraints and use a lower bit rate to achieve better OTR, especially for high-bandwidth requests over long distances. Also, the precise XT calculation and maximum OTR based on modulation format and bandwidth are considered in this step. Finally, the XTA-SMCSA approach is compatible with any routing algorithm.

XTA-SMCSA is presented in Algorithm 2. A path or list of paths calculated in the routing step is considered as an input of this step. XTA-SMCSA iterates over the list of input paths (Line 1). The list of modulation formats for all bit rates is filtered according to path length and the precomputed optical reaches shown in Table II and sorted in descending order of modulation level (Lines 2-3). Then, the MCSA algorithm loops through the number of allowable slices list (adopted from [12]), which is an ascending ordered consecutive subset of [1,2,4,8] starting with 1 and calculates the bandwidth (bw_{temp}) according to the given number of slices (Lines 4-5).

The XTA-SMCSA algorithm tries to use the minimum number of slices in resource allocation; therefore, the slice numbers list starts with one and increases in ascending order. Afterward, for all slices, based on the modulation format and bw_{temp}, the number of required slots is selected from Table II (Lines 6-9). The core and spectrum allocation policy selects the appropriate core and slot number (Lines 10-12). The prioritized core allocation algorithm based on [15] is assumed. The first-fit algorithm is used in the spectrum allocation step. If required spectrum slots are available, precise link-based XT [7] for the path and selected core and the slot that has the maximum number of active slots on its adjacent cores in the selected slots range are calculated (Lines 13-15). If the calculated XT meets the threshold (Table I with a margin of 7.69 dB), resources are assigned to the slice (Lines 16-17). If all slices are accommodated, selected resources for all slices and given path are returned as a response (Line 18-19). Even

Algorithm 2 XT-Aware Sliceable Modulation, Core, and Spectrum Allocation (XTA-SMCSA)

Inputs:

(1) P: Path set
(2) bw_i: Request's bandwidth
(3) NAS: List of the number of allowable slice

Parameters:

(1) bw_{temp}: Required bandwidth for each slice
(2) MFL: Modulation formats list
(3) $XT_{path,c,s}$: Calculated XT for $path$ for given c and s
(4) $s_{m,sl}$: The slot number between the selected slots ranges for slice sl, which has the maximum number of active neighbor cores

Output: Path and allocated resources to slices

Procedure: Modulation, Core and Spectrum Allocation

1: **for** ($path$ in P) **do**
2: Calculate MFL based on $path$ length and Table II
3: Sort MFL from highest to lowest modulation level.
4: **for** (max_{slice} in NAS) **do**
5: $bw_{temp} = \frac{bw_i}{max_{slice}}$
6: **for** (sl **from** 1 **to** max_{slice}) **do**
7: **for** (mf_{sl} in MFL) **do**
8: ns_{sl} = Find number of required slots based
9: on mf_{sl} from Table II for bw_{temp}
10: (c_{sl}, s_{sl}) = Selecting core and slots using
11: prioritized core assignment and first fit
12: spectrum allocation algorithms
13: **if** $(c_{sl}, s_{sl}) \neq \emptyset$ **then**
14: Find $s_{m,sl}$.
15: $XT_{path,c,s}$ = XT using Eq. 3 for $(c_{sl}, s_{m,sl})$
16: **if** $XT_{path} < XT_{th,mf_{sl}}$ **then**
17: Assign (mf_{sl}, ns_{sl}, c_{sl}, s_{sl}) to slice sl
18: **if** all slices are accommodated **then**
19: **return** $path$ and all allocated resources.
20: **end if**
21: **end if**
22: **end if**
23: **end for**
24: **end for**
25: Release all resources allocated to this request
26: **end for**
27: **end for**
28: **return** Request is blocked.

if one slice cannot be accommodated, the resources reserved by other slices of the given request are released (Line 25). Finally, the request is blocked if the appropriate resources for the request aren't allocated (Line 28).

The proposed XTA-SMCSA algorithm handles the accommodation of high-bandwidth requests in conjunction with a bandwidth slicing approach. Consider a scenario in which a 400 $Gbps$ request arrives between Oslo to London in the Pan-European network with seven cores on each link with 5 spectrum slots. Fig. 2 shows the shortest path from Oslo to

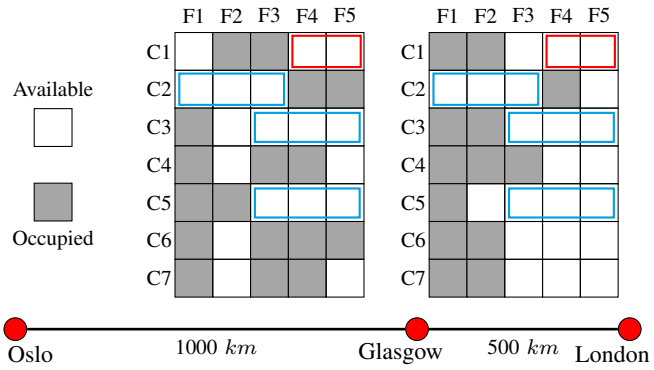

Fig. 2: Illustrtive example of slicing approach in XTA-SMCSA

London and spectrum usage for each link. The request would be blocked as the shortest path between the cities is 1500 km and based on the OTR from Table II, launching the lightpath is not possible using any modulation format. Applying the bandwidth slicing approach allows a 400 $Gbps$ request to be divided into either two 200 or four 100 or eight 50 $Gbps$ connections. The OTR using two 200 $Gbps$ slices is feasible only with QPSK modulation. This would require two blocks of five continuous and contiguous slots. As shown in the figure, this slicing option will be blocked. Therefore, the algorithm checks if four 100 $Gbps$ connections using the modulation formats can be scheduled. Using 16-QAM and QPSK, an 100 $Gbps$ connection would require two and three slots, respectively. As Fig. 2 shows, red and blue rectangles specify two and three slots continuous and contiguous channels on the end-to-end path. If these configurations satisfy the XT constraint, four 100 $Gbps$ lightpaths are established; otherwise, the algorithm proceeds to slice the request into eight 50 $Gbps$.

We independently reproduced both the prioritized core allocation and bandwidth slicing approaches, with our results affirming the achievements reported in [15] and [12]. However, we do not depict these results in our performance plots to maintain clarity and avoid redundancy.

IV. Performance Evaluation

Extensive numerical simulations assess the presented algorithms. All simulations are performed on the Pan-European network topology adopted from [16]. All network links are equipped with the MCF fiber with seven cores, and each core contains 320 frequency slots with 12.5 GHz bandwidth. The generated traffic follows the uniform source and destination pair selection. Also, inter-arrival times are considered with a Poisson distribution. The inter-arrival and holding times of requests follow the exponential distribution with an average of $1/\lambda$ and $1/\mu$ seconds. Erlang is calculated by λ/μ. To consider normalized traffic load, the inter-arrival time is divided by the number of cores. In all of the scenarios, the mean holding time is fixed at $3600s$. Also, the request's bandwidths are selected among 40, 100, 200, and 400 $Gbps$, following a distribution ratio of 1:5:3:1. The blocking probability results meet either 95% confidence interval or a maximum of 200 independent iterations (seeds) with 25,000 requests per iteration. In the XTA-SMCSA algorithm, h equals 3.78×10^{-9} for XT calculation. We support 50, 100, 200, and 400 $Gbps$

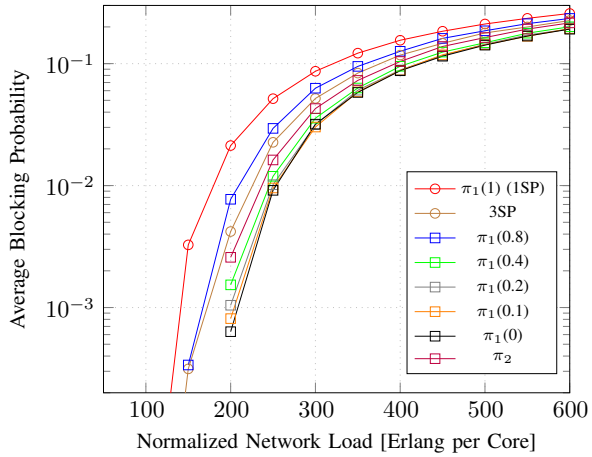

Fig. 3: Average request blocking probability for XTAR, 1SP and 3SP algorithms and XTA-SMCSA with up to eight slices.

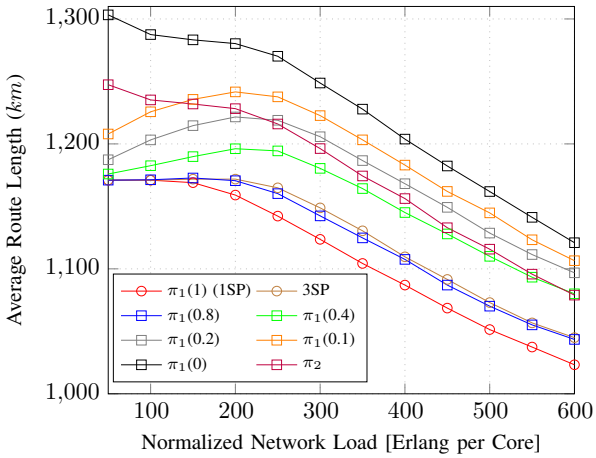

Fig. 4: Average successful lightpath length for XTAR, 1SP and 3SP algorithms and XTA-SMCSA with up to eight slices.

bit rates in the slicing step. As Table II shows, QPSK, 16-QAM, and 64-QAM modulation levels are considered in our scenario. Finally, one slot is considered as a guard band.

Fig. 3 illustrates the average request blocking probability versus the normalized traffic load (Erlang per core). The simulation results are performed for XTAR and KSP routing algorithms. XTAR with Policy I and α value is denoted as $\boldsymbol{\pi_1(\alpha)}$, and $\boldsymbol{\pi_2}$ refers to XTAR with Policy II. Also, 1SP is indicated to KSP for $K = 1$ and Policy I for $\alpha = 1$. Finally, the benchmark (KSP with $K = 3$) is shown by 3SP. In all policies and algorithms, blocking probability is increased due to a lack of resources by increasing the traffic load. As shown in the figure, 1SP, by selecting only the shortest path for requests, increases the congestion and causes the highest blocking probability among all algorithms. Increasing congestion leads to more XT, specifically in the higher loads. 3SP, by iteration over more candidate paths, decreases the blocking probability compared to 1SP. Also, Policy I for $\alpha = 0.8$ ($\pi_1(0.8)$) dedicates more weight to the length of the link during the link cost calculation, so its blocking is higher than 3SP.

During the link weight calculation, Policy II intensifies the XT cost of the link in proportion to the number of spans. It results in a lower blocking probability compared to the benchmark. As in 200 Erlangs per core, the average blocking drops around 35%. Policy II (π_2) reduces the mean blocking by 27% on average on all Erlangs. α decline means the portion of XT cost in the links' weight is increased. As the figure indicates, blocking probability is decreased by reducing the α, especially in low and moderate traffic loads. By using XTAR with Policy I for all α values lower than 0.4 and Policy II, the blocking probability achieves zero, for traffic loads less than 200 Erlangs per core. By assigning zero to α, XTAR with Policy I turns to a minimum XT greedy routing algorithm. Policy I for $\alpha = 0$ ($\pi_1(0)$) reduces the blocking probability around 85% and 42% in 200 Erlangs and, on average across all Erlangs, respectively.

The average accepted requests' path length versus normalized traffic load is depicted in Fig. 4. 1SP ($\pi_1(1)$) has the lowest average length in all loads. The mean distance for Policy I with α equals to 0.8 ($\pi_1(0.8)$) is close to the 3SP algorithm and higher than 1SP for medium and high loads. The average route path is increased by decreasing the α value in XTAR with Policy I. Due to ignoring link length, the minimum XT greedy ($\pi_1(0)$) achieves the highest lightpath length. So, at 50 Erlangs, the average distance is 1300 km, which is 130 km higher than the benchmark and 1SP. Also, at 600 Erlang, the average length of greedy policy stands around 7% and 10% higher than 3SP and 1SP. Overall, the average successful lightpaths' length by growing the load is reduced because the acceptance rate of shorter lightpaths is greater than that of longer ones in high loads.

As shown in Fig. 4, the behavior of the average length of lightpaths served by the XTAR algorithm with Policy II, across loads ranging from 50 to 200 Erlangs, is similar to that observed with Policy I when $\alpha = 0$ ($\pi_1(0)$). However, it is significantly different from other α values and the benchmark. Because of multiplying XT cost by the number of spans, Policy II at the starting point (when a network is empty) tries to select the links with shorter lengths (minimum span). Using short links increases the XT cost for them compared to longer links. Although the longer links have a higher number of spans, the low XT cost leads to selecting them greedily by Policy II in the mentioned loads. By increasing the load, the usage of short and long links is balanced, so the behavior of lightpaths' length follows all other algorithms.

The average number of transponders for successful requests versus the normalized traffic load is shown in Fig. 5. Regarding the XTA-SMCSA attempts to use the lower number of slices and the slight number of request blocking in the low loads, the algorithm uses the lower number of slices. Therefore, the average number of transponders in low loads is around 12% lower than in high loads because, in lower loads, the number of free slots and the length of available chunks of slots is high. Policy I with $\alpha = 0$ ($\pi_1(0)$), due to using longer paths, utilizes more slices and consequently more transponders in low and high loads. Policy I for $\alpha = 0$ consumes around 2% more transponders than 3SP in all loads except 150-450 Erlangs. Both proposed Policies in the moderate network loads use a lower number of transponders in comparison to KSPs

due to using a diverse range of links with low slot usage.

Table III illustrates the average blocking probability improvement of our proposed algorithm when compared to the benchmarks: 1SP and 3SP. The subsequent columns indicate the slight percentage increase in average path length and mean transponder usage per request under three different network topologies. Due to space constraints, we have included the average values over the load ranging from 50 to 600 Erlangs, for the XTAR with Policy I and α = 0 ($\pi_1(0)$) results compared to the benchmarks. We can clearly notice that our proposed algorithm significantly outperforms the baselines in terms of request blocking probability for all network topologies, with a slight increase in the average path length of established lightpaths for $\pi_1(0)$ and similar mean transponder usage. We can also observe that the high nodal connectivity of the Pan-European network leads to the highest improvement.

TABLE III: Performance comparison: XTAR with Policy I and α = 0 ($\pi_1(0)$) versus benchmarks.

Network	Benchmark	Blocking	Length	Transponder
Pan-European	1SP	55.26	-10.54	-0.87
	3SP	42.68	-8.89	0.22
NSFNet	1SP	38.62	-12.52	-4.98
	3SP	17.35	-4.54	2.19
USNet	1SP	31.39	-6.23	-3
	3SP	17.79	-3.62	1.51

V. Complexity Analysis: Time and Storage

In this section, the time complexity of the proposed algorithm is analyzed and compared with traditional benchmarks. The time complexity of KSP is in order of $O(K \cdot |V| \cdot (|E| + |V| \cdot \log|V|))$. The complexity of XTAR is expressed as $O(|E| \cdot C^2 \cdot S) + O((|V| + |E|) \cdot \log|V|)$ for a graph with $|V|$ vertices and $|E|$ edges, with every link containing C cores with S slots on each core. Given that S is significantly larger than $|V|$ and $|E|$, the $O(|E| \cdot C^2 \cdot S)$ term is the dominant factor in the XTAR complexity expression. Typically, in an SDM-EON network, as the number of cores (C) increases, the overall computational complexity of the algorithm will also increase. However, it is important to note that calculating the XT cost of each link is unnecessary for every route calculation. Initially, the weights of all links can be pre-calculated and stored. Subsequently, upon each request's arrival and departure, only the links' weight along the selected path is recalculated. In terms of storage complexity, KSP algorithm is $O(K \cdot |V|)$. In XTAR, storage complexity is attributed to storing spectrum slot states across all cores on each link. This is obtained by $O(|E| \cdot C \cdot S) + O(|V|)$.

VI. Conclusion

In conclusion, this paper presents a new dynamic crosstalk-aware routing, modulation, core, and spectrum allocation algorithm for sliceable demands in SDM-EONs. We evaluated two new policies of the proposed algorithm that reduce the impact of XT while allocating demands in the SDN controller of SDM-EONs. We also develop the XTA-SMCSA algorithm that can relax the spectrum contiguity constraints, assign the best modulation-format based on transmission reach, and establish reliable connections by checking the precise value of their XT. Based on in-depth performance analysis, we show that XTAR with $\pi_1(0)$ achieves the lowest request blocking probability compared to all other allocation methods.

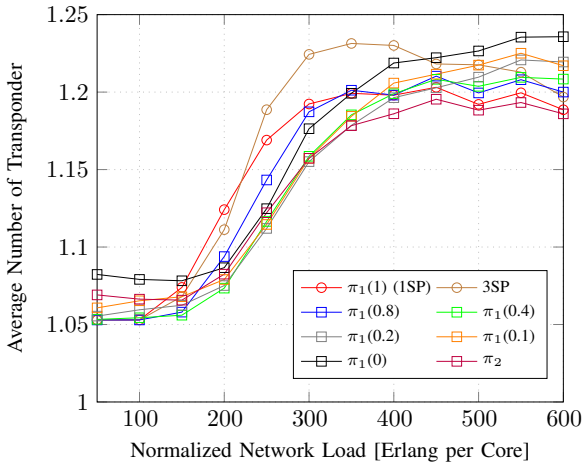

Fig. 5: The average number of transponders usage for XTAR, 1SP and 3SP algorithms and XTA-SMCSA with up to 8 slices.

Area of future work would be to use ML and deep-reinforcement learning approaches to select a subset of core and slots for XT cost calculation of XTAR.

References

[1] O. Gerstel et al., "Elastic optical networking: A new dawn for the optical layer?" *IEEE Commun. Mag.*, vol. 50, no. 2, pp. s12–s20, 2012.

[2] D. J. Richardson et al., "Space-division multiplexing in optical fibres," *Nature photonics*, vol. 7, no. 5, pp. 354–362, 2013.

[3] G. M. Saridis et al., "Survey and evaluation of space division multiplexing: From technologies to optical networks," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2136–2156, 2015.

[4] A. Rezaee et al., "PLI-aware dynamic routing in software defined elastic optical networks (SD-EONs)," in *ONDM 2023*. IEEE, 2023, pp. 1–3.

[5] E. E. Moghaddam et al., "Crosstalk-aware resource allocation in survivable space-division-multiplexed elastic optical networks supporting hybrid dedicated and shared path protection," *JLT*, vol. 38, no. 6, pp. 1095–1102, 2020.

[6] J. Su et al., "Dynamic impairment-aware RMCSA in multi-core fiber-based elastic optical networks," *Optics Communications*, vol. 518, p. 128361, 2022.

[7] M. Klinkowski et al., "Dynamic crosstalk-aware lightpath provisioning in spectrally–spatially flexible optical networks," *JOCN*, vol. 11, no. 5, pp. 213–225, 2019.

[8] S. Petale et al., "Tridental resource assignment algorithm for spectrally-spatially flexible optical networks," in *ICC 2021*. IEEE, 2021, pp. 1–6.

[9] ——, "TRA: an efficient dynamic resource assignment algorithm for MCF-based SS-FONs," *JOCN*, vol. 14, no. 7, pp. 511–523, 2022.

[10] ——, "An ML Approach for Crosstalk-Aware Modulation Format Selection in SDM-EONs," in *ONDM 2022*. IEEE, 2022, pp. 1–6.

[11] ——, "Machine learning aided optimization for balanced resource allocations in SDM-EONs," *JOCN*, vol. 15, no. 5, pp. B11–B22, 2023.

[12] Y. Wang et al., "Light-segment: Crosstalk-and modulation-aware spectrum allocation with segmentation in SDM-EON," in *ICC 2020*. IEEE, 2020, pp. 1–6.

[13] P. Jordi, "Flex-grid/SDM backbone network design with inter-core XT-limited transmission reach," *JOCN*, vol. 8, no. 8, pp. 540–552, 2016.

[14] R.-J. Essiambre et al., "Capacity limits of optical fiber networks," *JLT*, vol. 28, no. 4, pp. 662–701, Feb 2010.

[15] S. Fujii et al., "On-demand spectrum and core allocation for reducing crosstalk in multicore fibers in elastic optical networks," *JOCN*, vol. 6, no. 12, pp. 1059–1071, 2014.

[16] A. Rezaee et al., "Quality of transmission-aware control plane performance analysis for elastic optical networks," *Computer Networks*, vol. 187, p. 107755, 2021.