Proactive Fragmentation-aware Routing, Modulation Format, Core, and Spectrum Allocation in EON-SDM

Silvana Trindade
Institute of Computing
State University of Campinas
Campinas, Brazil
silvana@lrc.ic.unicamp.br

Nelson L. S. da Fonseca
Institute of Computing
State University of Campinas
Campinas, Brazil
nfonseca@ic.unicamp.br

***Abstract*—In Elastic Optical Networks with Space-Division Multiplexing (EON-SDM), dynamic allocation and de-allocation can generate spectrum fragmentation, increasing the blocking probability. Proactive solutions attempt to minimize or prevent future fragmentation occurrence by trying to find paths and blocks of slots to allocate. These solutions increase the chances of future connection requests to be allocated. This paper presents two proactive algorithms to avoid spectrum fragmentation in EON-SDMs. The proposed algorithms take into consideration the fragmentation state of the spectrum as well as potential bottleneck formation. Results demonstrate that our algorithms can reduce significantly the blocking probability while respecting the inter-core cross-talk constraint in Multi-Core Fiber (MCF).**

***Index Terms*—Space-Division-Multiplexing (SDM), Fragmentation, Modulation Formats, Inter-Core Cross-talk, Routing, Core, and Spectrum Allocation (RCSA).**

I. Introduction

In Elastic Optical Networks with Space-Division Multiplexing (EON-SDM), requests for light-paths establishment have different arrival and tear-down times. The dynamic allocation and de-allocation of slots generate spectrum fragmentation, characterized by the availability of non-continuous slots with total bandwidth sufficient to accommodate a new request but unavailable for allocation as a consequence of not being continuous. Fragmentation increases the blocking of connection requests, causing inefficient utilization of the spectrum.

Solutions to ameliorate the fragmentation problem can be reactive or proactive [4]. Proactive ones try to minimize or prevent future fragmentation occurrence. Reactive solutions focus on network defragmentation which is the process of removal of fragmentation by rearranging already established connections, either by re-routing and/or reallocating them. However, these solutions can cause traffic disruption, and the time spent to re-allocate the set of requests may not be negligible [4]. Indeed, most of the algorithms in the literature handle the fragmentation problem when allocating the spectrum in a proactive way.

In this paper, we propose two proactive Routing, Modulation Format, Core and Spectrum Allocation (RMCSA) algorithms to alleviate the fragmentation problem in SDM scenarios. The algorithms make decision on spectrum allocation considering the state of the spectrum fragmentation as well as the potential formation of bottleneck links. prioritization of cores in resource allocation is also employed. The algorithms employ adaptative modulation to deal with different Quality of Transmission (QoT) conditions.

The remainder of the paper is organized as follows. Section II describes related works that try to reduce the spectrum fragmentation. Section III introduces the proposed algorithm to optimize the use of resources in SDM networks and describes the set of comparison algorithms. Section IV discusses the performance of our proposed algorithms. Finally, Section V concludes the paper.

II. Related Work

In [5], the authors introduce a Reconfigurable Optical Add/Drop Multiplexers (ROADM) as well as an Architecture on Demand (AoD) to avoid blocking in EON-SDM networks, considering cross-talk and different modulation formats. In [6], the First-Fit (FF) policy is used to allocate bandwidth requests in partitions of the spectrum [7].

In [8], the authors present the concept of cross-core virtual concatenation in SDM networks with the spectrum contiguity constraint relaxed. They have considered that spectral-spatial super-channels have irregular shapes and their carriers can be distributed over different fiber cores. They also use this concept to minimize the fragmentation of EON-SDM Multi-Core Fibers (MCFs).

In the algorithm proposed in [10], slots are separated into different partitions. In [11], the prioritized areas are based on immediate (IR) and advance (AR) reservation requests to reduce the blocking probability. IR requests start the transmission immediately after they arrive, whereas, AR requests can reserve future resources. However, prioritizing areas can increase blocking compared to those given by classical RCSA (Routing, Core, and Spectrum Allocation) algorithms.

In this paper, we introduced two RMCSA algorithms to ameliorate the fragmentation problem. They classify paths based on the current fragmentation state of the spectrum, as

978-1-5386-8088-9/19/$31.00 ©2019 IEEE

well as on the potentiality of bottleneck link formation. Additionally, one of the algorithms proposed uses core prioritization to prevent cross-talk interference.

III. Proposed Algorithm

In this section, we introduce two algorithms: the Fragmentation-aware Routing, Modulation Format, Core, and Spectrum Allocation (FA-RMCSA) algorithm and the Fragmentation-aware Core prioritization RMCSA (FACP-RMCSA) algorithm. The RMCSA algorithm used in our algorithms is divided into two sub-problems, namely 1) routing and modulation selection and 2) spectrum allocation. In the routing and modulation format selection, the potential paths to host a request are identified and the number of slots necessary to allocate the requested bandwidth is calculated based on the length of the paths. In the core and spectrum allocation step, available slots are searched to accommodate the request. These two algorithms differ by the employment of core prioritization to avoid cross-talk as a selection criterion.

TABLE I
Notation used in Problem Formulation

Symbol	Description
$G = (V, E, W)$	Graph representing a network topology
E	Set of edges, the length being $\|E\| = C \times F \times L$
$e_{u,v,n}$	The n^{th} edge connecting nodes u and v
V	Set of nodes, where $V = \{v_1, v_2, \ldots, v_{\|V\|}\}$
W	Set of edge weights, where $W = \{w(e_{u,v}) \mid e_{u,v} \in E\}$
w	Weight of the edge $e_{u,v,n}$, where $w(e_{u,v,n}) < \infty$ if the n^{th} slot in the link connecting u to v is available and $w(e_{u,v,n}) = \infty$ if the slot is unavailable
$\hat{G}_{u,v} = (\hat{V}, \hat{E}, \hat{W})$	Labeled multi-graph, representing a graph with K-shortest paths between u and v such that $\hat{E}$ is the set of edges connecting $\{\hat{u}, \hat{v}\} \in \hat{E}$, $\hat{V}$ is the set of nodes and $\hat{W}$ is the set of costs associated with $\hat{E}$, corresponding the mapping edges from K-shortest paths in G
$\tilde{w}$	The edge $\tilde{e}_{u,v,n}$ weight, being $\tilde{w} < \infty$ if the corresponding edges in the graph are available and $\tilde{w} = \infty$, otherwise
$\tilde{V}$	Set of nodes, where $\tilde{V} = V$
$\tilde{e}_{\tilde{u},\tilde{v},n} \in \tilde{E}$	Edge connecting $\tilde{u}$ and $\tilde{v}$
$\tilde{e}_{\tilde{u},\tilde{v}}$	$\tilde{e}_{\tilde{u},\tilde{v}} = \{e_{u,v,n}\} \in \tilde{E}$ is a chain such that $\{e_{u,v,n}\}$ is the least ordered edge, $\tilde{e}_{u,v,n+\Delta}$ is the greatest ordered edge and $\mid \tilde{e}_{u,v} \mid = \Delta$
$\tilde{W}$	$\tilde{W} = \tilde{w}_n(\tilde{e}_{\tilde{u},\tilde{v}})$
$\tilde{G}_{n,\Delta} = (\tilde{V}, \tilde{E}, \tilde{W})$	The n^{th} labeled graph such that $\tilde{E}$ is the set of edges connecting $\{\tilde{u}, \tilde{v}\} \in \tilde{V}$ and $\tilde{W}$ is the set of costs associated with $\tilde{E}$.
$\tilde{E}$	Represents the mapping of Δ edges in $\hat{G}$
σ	Number of graphs extracted from multi-graphs, where $\sigma = \|\{\tilde{G}_{n,\Delta}\}\| = C \times (F - \Delta + 1)$
$\tau(\hat{G}, C, \Delta)$	Function which produces all σ graphs from $\hat{G}$
$W(p_n)$	$\sum_{\tilde{e}_{u,v} \in p_n} \tilde{e}_{u,v}$ weight of the p_n (the sum of the weights of all the edges in the chain)

A K-shortest paths algorithm finds the set of candidate paths, and each path is classified using two metrics: path fragmentation ratio and closeness centrality of the nodes along the chosen paths. The path fragmentation ratio gives the fragmentation state of the spectrum and the closeness centrality of a node indicates the potentiality of the node to contribute to the formation of bottleneck links.

In addition, the FACP-RMCSA algorithm prioritizes the cores for spectrum selection in order to prevent inter-core cross-talk. A new core prioritization method is introduced in this paper to reduce the inter-core cross-talk by avoiding allocation of the spectrum in adjacent cores. This method also reduces fragmentation by allocating bandwidth requests uniformly among the core.

To facilitate the selection of slots for allocation, the spectrum is represented as a multi-graph [9], [13]. In a multi-graph, several links connecting two nodes represent the possibilities of the use of the network link spectrum to allocate the requested bandwidth. The original multi-graph is divided into C multi-graphs, where C is the number of cores. These graphs are transformed into other multi-graphs, each with $N - \Delta + 1$ edges, with Δ being the bandwidth demand converted into a number of slots on the basis of the modulation format chosen. These multi-graphs are then transformed into $N - \Delta + 1$ graphs. In other words, the original multi-graph is transformed into $C \times (|F| - \Delta + 1)$ graphs. Each edge in these graphs represents a combination of Δ slots. This representation assures spectrum contiguity in the solution. In these graphs, an ∞ label value means that at least one slot is unavailable, whereas the value 1 means that all slots are available for allocation. An ∞ value is also assigned to an edge if a slot represented by the edge has an unacceptable cross-talk level.

A. Routing Selection

The paths chosen by the execution of K-shortest paths algorithm are not necessarily completely disjoint, i.e., the paths may have links and/or nodes in common, and this can lead to the formation of bottleneck links and the frequent allocation/de-allocation of slots in these links increases fragmentation.

In general, existing RCSA algorithms do not consider the fragmentation state of the spectrum as a metric in the routing selection; fragmentation is usually handled in the allocation of spectrum steps of these algorithms.

In our algorithms, after computing K-shortest paths, we calculate the path fragmentation ratio (FR) [14]. This metric captures the impossibility of using the spectrum due to the generated fragmentation along a path. Besides the allocation, a cross-talk value higher than a threshold values makes the slot unavailable.

Let B_i be the number of slots in the block. The availability value of each block, $\nu(\mathrm{B}_i)$, is defined as the maximum total data rate that can be provisioned using the slot [14]. The path fragmentation ratio for each candidate path is defined as:

$$FR = 1 - \left(\frac{\sum_i \nu(\mathrm{B}_i)}{\nu(\sum_i (\mathrm{B}_i))} \right). \quad (1)$$

The ratio is computed using the value of the fragmented spectrum and that of the non-fragmented one. $\sum_i \nu(\mathrm{B}_i)$ is the current value of all available blocks along the path $p_{s,d}$, while $\nu(\sum_i(\mathrm{B}_i))$ is the availability value from the blocks that can be allocated. $\sum_i \nu(\mathrm{B}_i) = \sum_k r_k^* \times y_k$, where $r_k^* = R_b \times (b-1)$ represents the raw data rate and $b-1$ the number of slots, R_b is the data rate achieved for a modulation format m, and y_k represents the number of light-paths that can be allocated having b_k bandwidth. FR is in the range $[0,1]$, where 0 is non-fragmented and 1 indicates a high spectrum fragmentation. In other words, if a path has FR close to 0, this path is a promising candidate to host a request.

The closeness centrality [15] expresses how close a node is to all other nodes in a network. Nodes with high closeness centrality values impact on the disjointness of paths and, consequently, favor the formation of bottleneck links, increasing blocking. The closeness centrality is defined as the inverse of the sum of the shortest distances between each node and every other node in the network. The closeness centrality measure is defined by:

$$CC(i) = 1/\sum_{i \neq j} d(i,j) \tag{2}$$

where $CC(i)$ represents the closeness centrality of the node i, n represents the number of nodes in the network, $d(i,j)$ is the shortest path distance between i and j. To minimize the formation of the bottleneck in the path selection step, we calculate the closeness centrality of each node in a network and we use this value as a metric to decide the path that has the lowest closeness centrality.

Numerical values of FR and CC are normalized by the sigmoid function: $\alpha = \gamma/(1+\exp(x-\mu))$, where x is the values, given by either Equations 1 or 2 and μ is a threshold. This sigmoid function dampens the given values, so variations in the FR or CC values are less intense when combining them in Equation 3. It also avoids that numerical values of one factor dominating the other in the resulting value. For the NSFNET and USA topologies, an empirical evaluation of the μ thresholds for FR and CC that produced the least blocking resulted in 0.8 and 0.5, respectively. The following equation combines FR and CC into a single metric, which represents the adequateness of a path for allocation:

$$\omega = \omega_{FR} \times \omega_{CC}, \tag{3}$$

where ω indicates the potentiality of path $p_{s,d}$ to accommodate the new request, ω_{FR} represents the normalized path fragmentation ratio, and ω_{CC} the normalized sum of closeness centrality.

B. The Fragmentation-aware RMCSA Algorithm

Algorithm 1 shows the steps to allocate a request $r(s,d,b)$. Line 1 computes the K-shortest paths for a pair of nodes s and d. If there is no path the request is blocked (Line 22). In Lines 4-8, the algorithm classifies each path, where $\mathbb{S}$ stores the computation of Equation 3 for every path in P (Line 7). These paths are then ordered in ascending order based on the corresponding result in the set $\mathbb{R}$ (Line 9). For each path (Lines 10-21), the algorithm tries to allocate $r(s,d,b)$. Lines 11-12 compute the number of slots Δ necessary to allocate the request with modulation format m. Line 13 computes the availability of the spectrum using the adopted multi-graphs representation of the spectrum. If all the weights for the shortest path $p_n \in P$ are ∞, there is no path to support the requests with demand b which observes the contiguity constraint (Line 14-16). To accommodate a request $r(s,d,b)$ after a path is selected, we need to find a block of available slots considering the cross-talk level on the slots (Line 14). For the FA-RMCSA, we use the First-Core First-Fit algorithm [16] to find the available slots. If there is no block of slots available, then we decrease the modulation format m and go back to Line 12. Otherwise, in Lines 15-16, we establish $r(s,d,b)$ along the path p_n and update the weights of the edges to ∞ for the slots that are allocated to this request. If all possibilities are tried and there is no chance to allocate $r(s,d,b)$, the request is blocked (Line 22).

Algorithm 1: Fragmentation-aware RMCSA

1 $P \leftarrow KSP(G,s,d)$;
2 **if** $P \neq \emptyset$ **then**
3     $\mathbb{S} \leftarrow \emptyset$;
4   **foreach** *path* $p_n \in P$ **do**
5         $\omega_{CC_{p_n}} \leftarrow SumOfClosenessCentrality(p_n)$;
6         $\omega_{FR_{p_n}} \leftarrow FragmentationRatio(p_n)$;
7         $\mathbb{S} \leftarrow \mathbb{S} \cup (\omega_{CC_{p_n}} \times \omega_{FR_{p_n}})$;
8   **end foreach**
9     $P \leftarrow R(p_n) \quad | \quad \forall_i R(p_n) \geq R(P_i)$;
10   **foreach** *path* $p_n \in P$ **do**
11         $m \leftarrow m(p_n) \quad | \quad \forall_i m(p_n) \geq m_i(p_n)$;
12         $\Delta \leftarrow$ determine the number of slots;
13         $\tau(\hat{G}_{p_n}, \tilde{C}, \Delta)$ for m;
14   **if** $\exists \ \delta(\tilde{G}_{n,\Delta}, r(s,d,b))$ **then**
15   Establish $r(s,d,b)$ as p_n;
16             $W(\tilde{e}_{u,v,i}) \leftarrow \infty \quad \forall \{u,v\} \in p_n$;
17   **else if** $m > BPSK$ **then**
18   Downgrade m and go back to Line 12;
19   **end if**
20   **end foreach**
21 **end if**
22 Block the request $r(s,d,b)$;

C. The Fragmentation-aware Core prioritization RMCSA Algorithm

In MCF, cross-talk is generated by the propagation of optical signals between cores due to an evanescent wave. An evanescent wave decays exponentially with the distance travelled, and the propagation of optical signals depends on the propagation constant of each core. To calculate the cross-talk in SDM networks, we use the equations given in [12]. Keeping the overall cross-talk of the network low is key to reduce the fragmentation effect since it presents a slot to be unavailable for allocation.

By carefully choosing the order of the cores in which slots are selected, we are able to prevent high cross-talk in a way that allocations are placed in non-adjacent cores whenever possible. This allows the network to accept more new requests and accept requests with a high modulation format.

We adopted the graph-complement approach to define the core prioritization criterion. The priority of the cores for allocation is pre-computed, considering an MCF as a graph, each core as a node and the edges connection to the adjacent cores. To generate the complement of this graph, one fills in all the missing edges required to form a complete graph and removes all the edges that were previously there. The only edges left are between non-adjacent cores, the proposed criterion sorts the cores considering the distance between each other.

In the Fragmentation-aware Core prioritization (FACP) RM-CSA algorithm, the difference is that this algorithm uses a specific pre-computed order of cores to select one core and then apply the First-Fit policy to select a block of slots, in the same way as for the FA-RMCSA algorithm. To apply core prioritization, we need to change Line 14 in Algorithm 1 to:

$$\exists\ \delta(\tilde{G}_{n,\Delta}, r(s,d,b), \mathbb{T}), \tag{4}$$

where $\mathbb{T}$ is the result of core prioritization, such that the slots will be selected from cores following the order in $\mathbb{T}$.

In Algorithm 1, the for-loop (Lines 6-10) runs K times the fragmentation calculation $(\hat{V} \times \hat{E})$, where K represents the number of candidate paths, $\hat{E}$ and $\hat{V}$ are the number of edges and the number of nodes in $\hat{G}$, respectively. In Algorithm 1, the main for-loop (Lines 12-22) runs K times where K is a constant. For each candidate path, the steps to determine the modulation format and to find the available slot blocks (Lines 13-21) can run up to M times, where M is the set of modulation formats. The innermost condition, the one to find a slot block to allocate $r(s,d,b)$, has time complexity $\hat{V} \times \hat{E}$. So in the worst case, the total execution time will be $(K \times (\hat{V} \times \hat{E})) + (K \times M \times \hat{V} \times \hat{E})$ if the condition in Line 16 runs for all options before establishing or blocking a request. Conditions are nested, so the bounds can be multiplied and the complexity of the Algorithm 1 is $\mathcal{O}(K \times |M| \times |\hat{V}| \times |\hat{E}|)$.

IV. Performance Evaluation

We compared our proposed algorithms with algorithms employing the First-Core First-Fit (FCFF) and Partition policies. In [16], among the evaluated algorithms by the authors, the lowest blocking probability was achieved by using FCFF in SDM networks. In [6], the authors proposed a partition mechanism with core prioritization (PT). In this policy, the spectrum is partitioned into areas having a specific number of slots, and a shared area to allocate connections which could not be accepted in a designated area. To make the comparison fair, we introduced thresholds for the cross-talk and adaptative modulation formats to these RCSA algorithms.

The number of slots depends on the requested capacity and the modulation format. We consider six modulation formats, BPSK, QPSK, and x-QAM, where $x \in \{8, 16, 32, 64\}$, with the transmission reach and supported bit-rates as modeled in [17], and each with a cross-talk threshold of $\{-14, -18.5, -21, -25, -27, -34\}$, respectively. Since the minimum distance between symbols is most strongly affected by the rotated interferer, we expect a strong impact on the cross-talk generated. The values for calculating the cross-talk [12] employed the formulation in [5].

We implemented our algorithms in the FlexGrid simulator [18]. Each simulation was run for 10^5 requests, and the load (using Erlang distribution) was increased in steps of 25 erlangs for each simulation from 0 to 500. The bandwidth requests were generated between 50 Gb and 400 Gb, with granularities of 50 Gb.

In the simulations, the NSFNET and USA topologies were employed. The NSFNET topology (Figure 1) has 14 nodes and 21 fiber links and the USA topology (Figure 2) has 24 nodes and 43 fiber links. Each fiber link is bidirectional and contains 7 cores, each with 320 slots and slot spacing of 12.5 GHz.

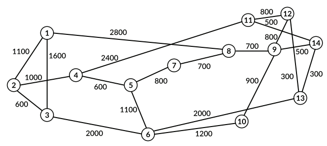

Fig. 1. NSFNET topology with 14 nodes and 21 links

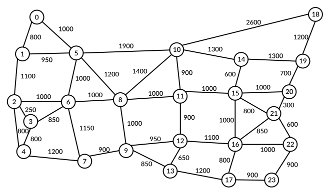

Fig. 2. USA topology with 24 nodes and 43 fiber links

To evaluate the algorithms, we used the following metrics: Bandwidth Blocking Ratio (BBR); mean cross-talk generated; and network fragmentation [4]. The network fragmentation is defined as $\chi = 1 - \phi$, where ϕ represents the available slot ratio between the maximum size of the block of slots and the total number of slots in the network.

Figure 3 shows the BBR as a function of the load for the NSFNET topology. The PT and FCFF algorithms start blocking requests under loads of 50 erlangs, and FA-RMCSA algorithm under loads of 125 erlangs. But, the FACP-RMCSA algorithm starts blocking requests only after 375 erlangs. The FA-RMCSA algorithm produces BBR values one order of magnitude lower than the BBR produced by the PT algorithm under loads of 200 erlangs. The FACP-RMCSA algorithm produces the lowest BBR values under all loads, the difference being two orders of magnitude when compared to the BBR

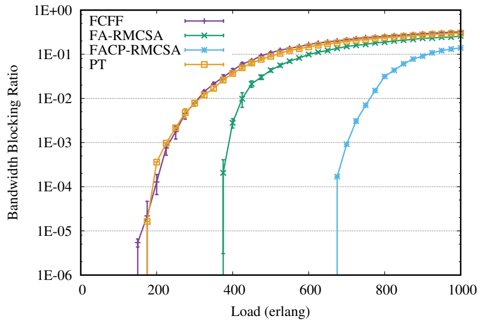

Fig. 3. Bandwidth blocking probability as a function of the load for the NSFNET topology

given by the other algorithms under loads of 400 erlangs. The difference in blocking produced by FA-RMCSA and FACP-RMCSA algorithms results from the adoption of core prioritization by the latter.

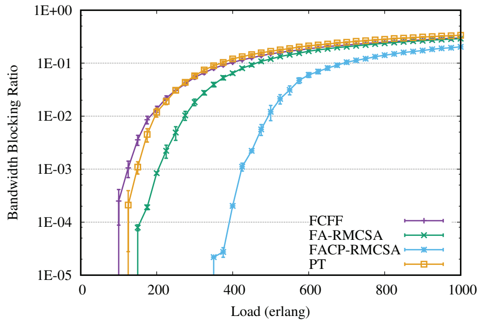

Fig. 4. Bandwidth blocking probability as a function of the load for the USA topology

Figure 4 shows the BBR as a function of the load for the USA topology. While PT and FCFF algorithms start blocking under loads of 25 erlangs and the FA-RMCSA algorithm under loads of 100 erlangs, the FACP-RMCSA algorithm starts blocking requests only under loads of 300 erlangs. The BBR produced by the FA-RMCSA algorithm is three orders of magnitude lower than those produced by the PT and FCFF algorithms under loads of 100 erlangs. The FACP-RMCSA algorithm produces the lowest BBR regardless of the load, confirming that our core prioritization contributes to the avoidance of blocking. When compared to the PT and the FCFF algorithms the BBR values given by the FACP-RMCSA algorithm is almost four orders of magnitude lower than those produced by the PT algorithm, which uses another core prioritization criterion.

Figure 5 shows the inter-core cross-talk as a function of the load for the NSFNET topology. The cross-talk value increases for all algorithms as the traffic load increases. The cross-talk value produced by the FCFF algorithm is 6 dB higher than that produced by FACP-RMCSA algorithm under loads of 25 erlangs. Regardless of the load, the FACP-RMCSA algorithm produces the lowest cross-talk, as a consequence of the core prioritization criteria adopted.

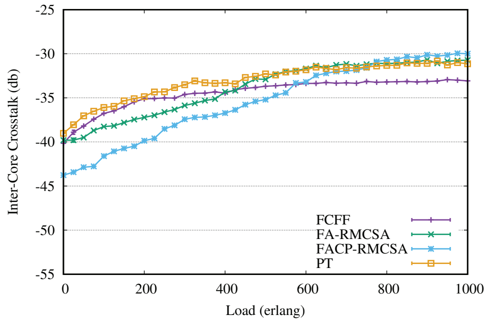

Fig. 5. Average inter-core cross-talk as a function of the load for the NSFNET topology

Figure 6 shows the inter-core cross-talk as a function of the load for the USA topology. The difference in cross-talk between algorithms is smaller for the USA topology than they are for the NSFNET topology. The cross-talk produced by FACP-RMCSA algorithm is 5 dB lower than that given by the FCFF algorithm under loads of 25 erlangs. The cross-talk given by the FACP-RMCSA algorithm is similar to the one produced for the NSFNET topology, being the lowest cross-talk value produced among all the values produced by the other algorithms. The mean difference between the cross-talk given by the FACP-RMCSA algorithm and the FCFF algorithm is around 14%, and for loads lower than 500 erlangs.

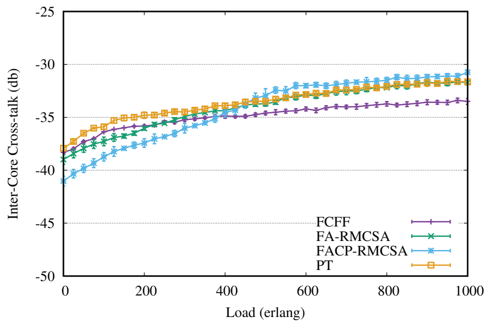

Fig. 6. Average of inter-core cross-talk as a function of the load for the USA topology

Figure 7 shows the network fragmentation for the NSFNET topology. The network fragmentation metric is not representative for the PT algorithm, as the algorithm divides the spectrum into partitions, grouping requests by bandwidth. This limitation causes a misleading impression that it has more available slots, where in reality, it doesn't. As a consequence, we are not going to use its results for the following considerations. The network fragmentation produced by FACP-RMCSA algorithm is 17% lower when compared to the results produced by the FA-RMCSA algorithm under loads of 125 erlangs. The FCFF, FA-RCSA and FACP-RCSA algorithms produce similar results after loads of 200 erlangs.

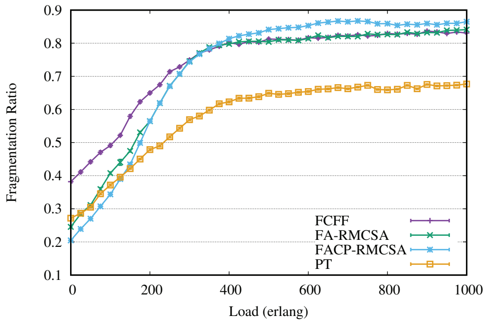

Fig. 7. Network fragmentation used to establish a connection request as a function of the load for the NSFNET topology

Figure 8 shows the network fragmentation for the USA topology. Under loads of 75 erlangs, FCFF algorithm produces the lowest network fragmentation compared to the FACP-RMCSA and FA-RMCSA algorithms, a difference of 26% to FA-RMCSA and 2% to FACP-RMCSA. After loads of 200 erlangs, the results produced by the FCFF algorithm are close to the ones produced by the FA-RMCSA and FA-RMCSA algorithms.

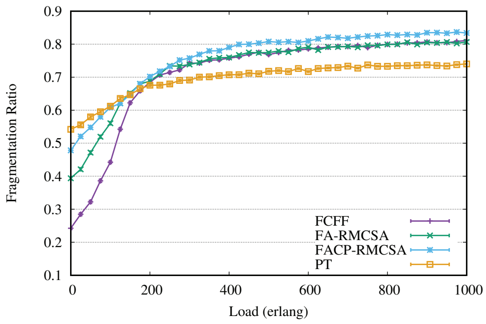

Fig. 8. Network fragmentation used to establish a connection request as a function of the load for the USA topology

V. Conclusion

In this paper, we investigated the problem of fragmentation in SDM networks. We introduced two new algorithms to proactively avoid fragmentation in dynamic scenarios.

FACP-RMCSA and FA-RMCSA algorithms improve significantly the amount of provisioned bandwidth. The FACP-RMCSA algorithm starts blocking only after 300 erlangs, while the other algorithms start blocking under much lower loads. This is due to the way it handles fragmentation and the impact that the fragmentation has on the BBR. Under all loads, the FA-RMCSA and FACP-RMCSA algorithms are able to achieve significant lower BBR compared to the FCFF and PT algorithms. The FACP-RMCSA algorithm produces BBR three orders of magnitude lower than those produced by the FCFF and PT algorithms. When comparing FACP-RMCSA algorithm with the FA-RMCSA algorithm, we can clearly see the benefits that the core prioritization technique has on the fragmentation. For the same network fragmentation, FACP-RMCSA algorithm can produce a BBR much lower than that given by the FA-RMCSA algorithm, demonstrating a clear advantage of our core prioritization method.

VI. Acknowledgment

The authors would like to thank the National Council for Scientific and Technological Development (CNPq) and grant 15/24494-8, São Paulo Research Foundation (FAPESP) for the financial support.

References

[1] D. Richardson, J. Fini, and L. Nelson, “Space-division multiplexing in optical fibres,” *Nature Photonics*, vol. 7, no. 5, p. 354, 2013.

[2] E. Goldstein, L. Eskildsen, and A. Elrefaie, “Performance implications of component crosstalk in transparent lightwave networks,” *IEEE Photonics Technology Letters*, vol. 6, no. 5, pp. 657–660, 1994.

[3] Y. Yin, H. Zhang, M. Zhang, M. Xia, Z. Zhu, S. Dahlfort, and S. B. Yoo, “Spectral and spatial 2d fragmentation-aware routing and spectrum assignment algorithms in elastic optical networks,” *Journal of Optical Communications and Networking*, vol. 5, no. 10, pp. A100–A106, 2013.

[4] B. C. Chatterjee, S. Ba, and E. Oki, “Fragmentation problems and management approaches in elastic optical networks: A survey,” *IEEE Communications Surveys & Tutorials*, 2017.

[5] A. Muhammad, G. Zervas, and R. Forchheimer, “Resource allocation for space-division multiplexing: optical white box versus optical black box networking,” *Journal of Lightwave Technology*, vol. 33, no. 23, pp. 4928–4941, 2015.

[6] H. Tode and Y. Hirota, “Routing, spectrum, and core and/or mode assignment on space-division multiplexing optical networks,” *Journal of Optical Communications and Networking*, vol. 9, no. 1, pp. A99–A113, 2017.

[7] S. Fujii, Y. Hirota, H. Tode, and K. Murakami, “On-demand spectrum and core allocation for reducing crosstalk in multicore fibers in elastic optical networks,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 6, no. 12, pp. 1059–1071, Dec 2014.

[8] Y. Zhao and J. Zhang, “Crosstalk-aware cross-core virtual concatenation in spatial division multiplexing elastic optical networks,” *Electronics Letters*, vol. 52, no. 20, pp. 1701–1703, 2016.

[9] P. M. Moura, N. L. da Fonseca, and R. A. Scaraficci, “Fragmentation aware routing and spectrum assignment algorithm,” in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1137–1142.

[10] W. Fadini, B. C. Chatterjee, and E. Oki, “A subcarrier-slot partition scheme with first-last fit spectrum allocation for elastic optical networks,” *Computer Networks*, vol. 91, pp. 700–711, 2015.

[11] S. Sugihara, Y. Hirota, S. Fujii, H. Tode and T. Watanabe, “Dynamic resource allocation for immediate and advance reservation in space-division-multiplexing-based elastic optical networks,” *Journal of Optical Communications and Networking*, vol. 9, no. 3, pp. 183–197, 2017.

[12] T. Hayashi, T. Taru, O. Shimakawa, T. Sasaki, and E. Sasaoka, “Design and fabrication of ultra-low crosstalk and low-loss multi-core fiber,” *Optics express*, vol. 19, no. 17, pp. 16 576–16 592, 2011.

[13] H. M. N. S Oliveira, N. L. da Fonseca, “Protection, Routing, Modulation, Core, and Spectrum Allocation in SDM Elastic Optical Networks,” in *IEEE Communications Letters*. IEEE, 2018, pp. 1806–1809.

[14] R. Wang and B. Mukherjee, “Spectrum management in heterogeneous bandwidth optical networks,” *Optical Switching and Networking*, vol. 11, pp. 83–91, 2014.

[15] L. C. Freeman, “Centrality in social networks conceptual clarification,” *Social networks*, vol. 1, no. 3, pp. 215–239, 1978.

[16] M. Klinkowski, P. Lechowicz, and K. Walkowiak, “Survey of resource allocation schemes and algorithms in spectrally-spatially flexible optical networking,” *Optical Switching and Networking*, 2017.

[17] C. T. Politi, V. Anagnostopoulos, C. Matrakidis, A. Stavdas, A. Lord, V. López, and J. P. Fernández-Palacios, “Dynamic operation of flexi-grid ofdm-based networks,” in *Optical Fiber Communication Conference*. Optical Society of America, 2012, pp. OTh3B–2.

[18] P. M. Moura, D. M. André, and S. Trindade, “Flexgridsim: Flexible grid optical network simulator,” 2018. [Online]. Available: https://www.lrc.ic.unicamp.br/FlexGridSim/.