Scientific Research Publishing

Journal of Computer and Communications, 2022, 10, 90-102
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Minimizing Traffic Blocking and Inter-Crosstalk in Spatial Division Multiplexing over Elastic Optical Networking

Joseph Ncube, Nathanael Larsey, Raphael K. M. Ahiaklo-Kuz, Charles Jnr. Asiedu, Enendu Uche Okechukwu

Huzhou University, Huzhou, Zhejiang Province, China
Email: joencube4@gmail.com, natlarsey1@gmail.com, raphaelkuz@gmail.com, babbage.jnr@gmail.com, enenduuche@gmail.com

How to cite this paper: Ncube, J., Larsey, N., Ahiaklo-Kuz, R.K.-M., Asiedu, C.J. and Okechukwu, E.U. (2022) Minimizing Traffic Blocking and Inter-Crosstalk in Spatial Division Multiplexing over Elastic Optical Networking. *Journal of Computer and Communications*, **10**, 90-102.
https://doi.org/10.4236/jcc.2022.105006

Received: March 8, 2022
Accepted: May 28, 2022
Published: May 31, 2022

Abstract

As a promising solution, virtualization is vigorously developed to eliminate the ossification of traditional Internet infrastructure and enhance the flexibility in sharing the substrate network (SN) resources including computing, storage, bandwidth, etc. With network virtualization, cloud service providers can utilize the shared substrate resources to provision virtual networks (VNs) and facilitate a wide and diverse range of applications. As more and more internet applications migrate to the cloud, the resource efficiency and the survivability of VNs, such as single link failure or large-scale disaster survivability, have become crucial issues. Elastic optical networks have emerged in recent years as a strategy for dealing with the divergence of network application bandwidth needs. The network capacity has been constrained due to the usage of only two multiplexing dimensions. As transmission rates rise, so does the demand for network failure protection. Due to their end-to-end solutions, those safeguarding paths are of particular importance among the protection methods. Due to their end-to-end solutions, those safeguarding paths are of particular importance among the protection methods. This paper presents approaches that provide a failure-independent route-protecting p-cycle for path protection in space-division multiplexed elastic optical networks. This letter looks at two SDM network challenges and presents a heuristic technique (k-shortest path) for each. In the first approach, we study a virtual network embedding (SVNE) problem and propose an algorithm for EONs, which can combat against single-link failures. We evaluate the proposed POPETA algorithm and compare its performance with some counterpart algorithms. Simulation results demonstrate that the proposed algorithm can achieve satisfactory performance in terms of spectrum utilization and blocking ratio, even if with a higher backup redundancy ratio.

Keywords

Spatial Division Multiplexing, Elastic Optical Networking, Protected Routing, Spectrum, Core, Time Allocation (Popeta)

1. Introduction

With the rapid development of 5G mobile networks, the Internet of Things, cloud computing, and other emerging technologies, diversified network traffic have resulted in an exponential expansion in the amount of IP requests, with an annual growth rate of worldwide network traffics reaching 30%. This susceptibility has prompted the creation of a number of optical network protection and restoration strategies, including the p-cycle, which combines the speed of ring networks with the efficiency of topologically diverse grid networks. P-cycle uses pre-configured backup resources to allow spare capacity to be used to safeguard working pathways. Pre-configured backup pathways can cut recovery time in half. As long as these spans have end-points on the p-cycle, the P-cycle can protect both on-cycle and off-cycle spans (straddling spans). The failure-independent path-protecting (FIPP) p-cycle, which protects end-to-end primary pathways with end nodes on the p-cycle, is one of the most researched varieties of p-cycle. The shared-backup path protection (SBPP) method, which establishes pre-planned backup paths for fragmented primary paths, is another prominent scheme for path protection in optical networks. Aside from the requirement that the backup and primary pathways be disjoint, the backup path must have no shared spans with backup paths of any primary path that is not totally disjoint from its own primary path. SBPP and FIPP are both failure-independent, which implies that fault detection occurs only at the end node and no fault location is required in real time, regardless of whether a node or a span has failed or where the failure occurred.

According to Zhu, R., Zhao, Y., Yang, H. [1], they advocated using K-shortest paths to calculate routes in an RMLSA solution and allocating the spectrum using the lowest starting slot in the available spectrum. Although the modulation formats were not explored by authors Horota, A., Reis, L., Figueiredo, G. [2], a FIPP p-cycle was proposed for the protection of elastic SDM-EONs. SBPP, adaptive modulation, and a multigraph representation of the spectrum are all used in the BARTRMAN method, which is also used in the POPETA algorithm to defend SDM-EON. Although FIPP p-cycles have been explored in elastic optical networks, only the by authors, Zhao, J., Yao, Q., Ren, D., Li, W. [3] protection for SDM-EONs has been proposed. No other research into modulation in p-cycle protected SDM EONs has been done to our knowledge.

SDM-EONs have the advantage of considerably increasing network capacity and allowing for more flexible and efficient use of spectrum resources. However, it introduces the RSCTA problem, which is characterized by significant crosstalk and high computational complexity. The mutual interference caused by the transmission of signals on the same frequency between adjacent cores is known as crosstalk. The core-pitch is getting narrower and smaller as the number of cores in the fiber grows, and crosstalk between nearby cores is becoming more significant. At the same time, the increased core dimension in MCF-EONs increases the computational complexity when compared to standard EONs. The impact of inter-core crosstalk in the RSCTA problem, on the other hand, can be mitigated by properly allocating core and spectrum resources to demands. The advent of elastic optical networks has prompted a number of studies, most of which have focused on RSA algorithms, however, RSCA solutions have only recently been developed. An energy efficiency grooming and hybrid crosstalk solution (EEG-HCS) algorithm was proposed, which can reduce fixed energy consumption while also protecting bandwidth by sharing an existing optical route. It is proposed to use a hybrid ICXT system that includes passive avoidance and ICXT awareness.

As a result, figuring out how to handle the RSCTA problem in SDM-EON is a challenge that should not be overlooked. The usage of SDM introduces various issues with inter-circuit interference in fiber, with a focus on inter-core crosstalk interference. Some key principles surrounding EON, as well as the characterization of SDM supporting equipment, are discussed in this letter. The survey concludes with a state-of-the-art assessment and a summary of the major difficulties identified through a thorough examination of the related literature.

In this letter, we offer a protected rOuting, sPectrum, corE, and Time Allocation method (POPETA) for protecting elastic optical networks with space division multiplexing (SDM-EONs) from failure, as well as a heuristic algorithm appropriate for large-scale network topologies. The shortest paths are chosen as principal paths by the POPETA algorithm. By borrowing some algorithms and their mathematical formulae, we propose to adopt a mixed methodology. To successfully accommodate different traffic demands, solutions to the RSCA problem in elastic optical networks are required, similar to the routing and spectrum assignment (RSA) problem in elastic optical networks.

2. The Popeta Algorithm

In a FIPP p-cycle protected network, the POPETA algorithm determines the formation of light paths. Such light paths are created if and only if the network can be secured against a single failure using a FIPP p-cycle. POPETA uses the RSCTA algorithm, which examines the distribution of the same spectrum to each fiber along a light path's route.

Table 1 shows the distance required to establish the link, the related modulation method, and the number of spectrum slots, as determined by POPETA of the transmission route. For each modulation mode, this table is generated in BPSK, QPSK, 8-QAM, 16-QAM, 32-QAM, and 64-QAM. A spectrum slot's capacity is 12.5, 25, 37.5, 50, 62.5, and 75 Gb/s, respectively. POPETA uses a labeled multigraph to simulate the spectrum availability in the network (**Figure 1(a)**).

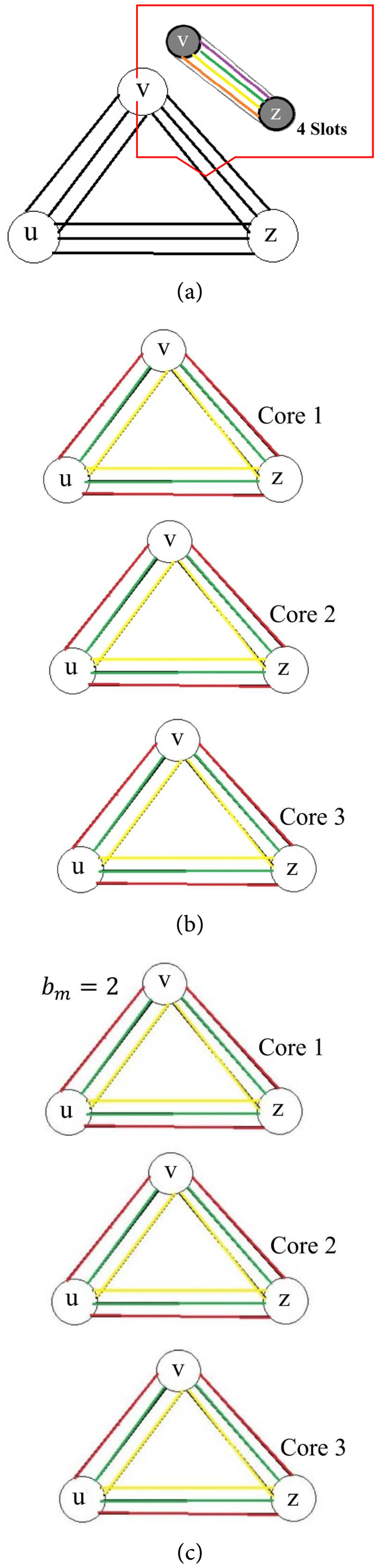


Figure 1. (a) Network with 3 cores and 4 slots; (b) The Multigraph separated by cores each one representing 4 slots; (c) The Multigraph in that set edges are mapped in to one edges, contiguity constraint.

Table 1. Equation abbreviations and meanings

A. NOTATION
S: source node;
D: destination node;
B: bandwidth demand;
N: number of slots between two nodes;
C: number of core;
V: set of nodes
$e_{u,v,n}$: the n^{th} edges connecting u and v;
$E = \{e_{u,v,n}\}$: set of edges;
$G = (V, E, W)$: label multigraph composed of edges E and a set of edge weight, W.
$m = 1, \cdots, M$: modulation formats;
B_m: bandwidth demand in slots in the basis of the modulation format chosen;
$R(s, d, b)$: request from the node s to the node d with bandwidth demand b;
$\delta(G, r(s, d, b_m))$: shortest path between s and d in G that satisfies the request for b_m slots:
$\omega(e_{u,v,n})$: Weight of the edge $e_{u,v,n}$;
$\tilde{G}_{n,b_m} = (\tilde{V}, \tilde{E}, \tilde{W})$: 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 $\tilde{E}$. The edges in $\tilde{E}$ correspond to the mapping of b_m edges in G, starting at the n^{th} edges;
$\sigma = \left\|\{\tilde{G}_{n,b_m}\}\right\| = C \times (N - b_m + 1)$: Number of graphs extracted from the multigraph;
$\tau(G, C, b_m) = \{\tilde{G}_{n,b_m}\}$: Function which produces all σ graphs from G;
P_n : chain of $\tilde{G}_{n,b_m}$ such that the source node s is the least ordered node and d is the greatest ordered node;
$W(P_n)$: Weight of the path P_n, which is the sum of the weights of all the edges in the chain;
$W_{P_{s,d}}$ = weight of the shortest path between s and d;
B_n : Chain of $\tilde{G}_{n,b_m}$ such that the number of vertices is equal to the number of edges, and every vertex has degree 2;
$B_{u,v}$: set of all p-cycles containing the vertices u and v in G;
$\theta(\tilde{G}_{n,b_m}, P_n, r(s, d, b))$: Shortest cycle between s and d in $\tilde{G}_{n,b_m}$, which $P_{B_{s,d}}$ ate link disjoint to P_n;
$\upsilon(P_n, B_{u,v}, r(s, d, b))$: P-cycle in $B_{u,v}$ which $P_{B_{u,v}}$ are link disjoint to P_n and satisfies the request of bandwidth b;
$W(B_n)$: The weight of the p-cycle B_n, which is the sum of the weights of all the edges in the chain;
$W_{B_{s,d}}$ = weight of the p-cycle which protects the path between s and d;

Continued

B. POPETA

The POPETA algorithm is introduced in Algorithm 1. Line 1 transforms the multigraph into $C \times (N - b_m + 1)$ graphs. Line 2 finds the shortest path for all $\tilde{G}_{n,b_m}$ graphs and chooses the cheapest one. If all of the shortest path weights are ∞, it signifies that there is no path for demand b that observes the contiguity requirement.
Line 3 chooses the shortest path with the lowest weight value out of all the shortest paths. There is no path in the network that satisfies the request for b_m slots under the contiguity constraint if the weight of all the shortest paths is ∞ (Line 4). The request is blocked if no path is available (Line 5).

Otherwise, a p-cycle is required to preserve this light-path (Line 7). When a p-cycle shields both an active and a new request, a light-path (Line 8) is constructed, with the weight of the associated edges in the multi-graph G altered to ∞ (Line 9).
If no such p-cycle exists, one is constructed to protect the newly established light-path (Line 12). The shortest possible cycle between source and destination nodes is considered while creating the p-cycle, however if no such p-cycle can be found, the request is blocked (Line 15).

Aside from that, the major path and the p-cycle (Line 17) have been established. Lines 18 and 19 alter the weight of relevant edges in the multi-graph G to ∞, indicating that the slots have been assigned to the newly formed light-path. The PERFECTA algorithm's complexity is examined next. $M \times O(\|E\| + \|V\|)$, where M is the number of modulation levels that can be employed, is the complexity of changing the original multi-graph into alp graphs. The Dijkstras algorithm is run at least $M \times C \times (N - b)$ times for the major path. The Suurballe algorithm is run at least $M \times C \times (N - b)$ times to generate p-cycles. Given that both Dijkstra's and Suurballe's algorithms have a complexity of $O(\|E\| + \|V\| \log \|V\|)$, Because C, N, M, and b are constants, the PERFECTA algorithm's complexity is $O(\|E\| + \|V\| \log \|V\|)$.

Algorithm 1 POPETA

1: $\tau(G, C, b_m) \forall m \in M$
2: $(W(P_n), P_n) = \delta(G_{n,bm,} r(s,d,b)) \forall n \in \sigma$
3: $W_{P_{s,d}} = W(P_n) | \forall i \; W(P_n) \le W(P_i)$
4: **if** $W_{P_{s,d}} = \infty$ **then**
5: block $r(s,d,b)$
6: **else**
7: **if** $\exists (P_n, B_{s.d}, r(s,d,b))$ **then**
8: establish $r(s,d,b) as\ P_n\ and\ B_{s,d}$
9: $w(e_{u,v,t}) = \infty \forall \{u,v\} \in P_t$
10: **else**
11: $\tau(G, C, b_m) \forall m \in M$
12: $(W(B_n), B_n) = \theta(G_{n,bm,} P_n, r(s,d,b)) \forall n$
13: $W_{B_{s,d}} = W(B_n) | \forall i \; W(B_n) \le W(B_i)$
14: **if** $W_{B_{s,d}} = \infty$ **then**
15: block $r(s,d,b)$
16: **else**
17: establish $r(s,d,b) as\ P_n\ and\ B_n$
18: $w(e_{u,v,t}) = \infty \forall \{u,v\} \in P_t$
19: $w(e_{u,v,t}) = \infty \forall \{u,v\} \in B_t$
20: **endif**
21: **endif**
22: **endif**

The availability of a slot is indicated by a label on one of the edges. If no existing light-path is using a slot and the crosstalk on that slot is less than a pre-defined threshold value, it is considered accessible.

In **Figure 1(b)**, the multi-graph is divided into C multi-graphs, where C specifies the number of cores.

Each multi-graph is then turned into a new multi-graph with $N-b_m+1$ edges (**Figure 1(c)**), where b_m is the bandwidth demand in slots based on the modulation format selected. Yin, S., Chen, Y., Ding, S., Zhang, Z. [4]. After that, each of these multigraphs is converted into $N-b_m+1$ graphs.

3. Performance Evaluation

Simulation tests with 7 core fibers and the Complex Elastic Optical Network Simulator (CEONS) were used to evaluate POPETA's performance in multi-core networks. 100,000 requests were created in each simulation, and the identical sets of seeds were used in all of the algorithms. Using the independent replication method, confidence intervals of 95 percent confidence were calculated. There were seven different sorts of requests, and the bandwidth demand of each was chosen at random from 25, 50, 125, 200, 500, 750, and 1000 Gbps.

The efficiency of POPETA was assessed using the Pan-European (**Figure 1(a)**) and National Science Foundation (NSF) topologies (**Figure 2(b)**). The Pan-European topology contains 28 nodes and 39 linkages, while the NSF topology has 16 nodes and 25 links. The traffic load was increased in 25 erlang increments. The spectrum was divided into 240 slots, each with a frequency of 12.5 GHz.

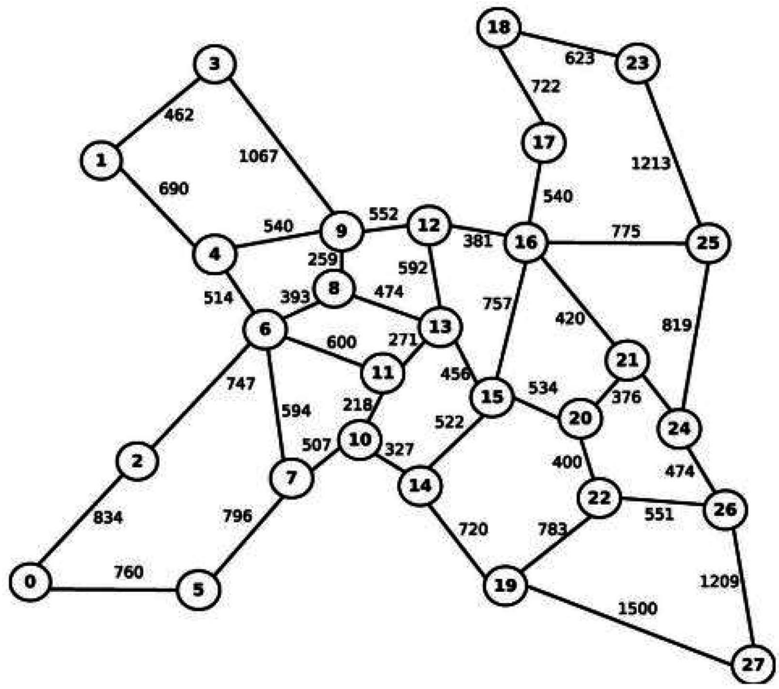
The findings for networks implementing the crosstalk-aware provisioning strategy with dedicated path protection (Cap-DPP) algorithm proposed by authors Yin, S., Chen, Y., Ding, S., Zhang, Z. in [6], but with adaptive modulation are shown in the figures. The findings for networks using the SSCAM (shared backup spectrum and core allocation and modulation) algorithm based on the methods provided in [7] by authors Nunes da Silva Oliveira, H.M. but incorporating protection and adaptive modulation are shown in curves labeled SSCAM (shared backup spectrum and core allocation and modulation).

The routing problem, the spectrum problem, and core and mode assignment are all tackled separately in SSCAM. This method uses many primary and backup routes that have been pre-calculated.

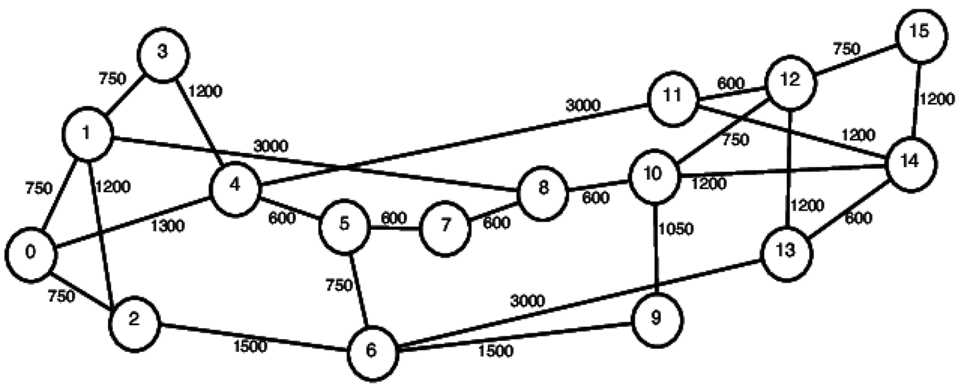
For the SSCAM algorithm, the backup path uses a 1:N scheme. The bandwidth blocking ratio (BBR) for the Pan-European topology is shown in **Figure 3(a)**. While Cap-DPPM and SSCAM begin blocking requests at 100 and 125 erlangs, POPETA and BARTRMAN begin blocking requests only at 225 erlangs. For the role range of load, the BBR produced by POPETA and BARTRMAN are unremarkable. Under such loads, the BBR produced by the POPETA algorithm differs by one and two orders of magnitude from those produced by the SSCAM and Cap-DPPM algorithms, respectively. The low BBR produced by BARTRMAN and POPETA demonstrates the advantages of creating primary and backup routes using a multi-graph representation of the spectrum.

Cap-DPPM generates a high BBR as a result of not sharing backup pathways. Despite the bandwidth reservation for pre-provisioning of backup lines, these findings show that the POPETA algorithm delivers acceptable blockage for SDM-EON. The BBR for the NSF topology is shown in **Figure 3(b)** as a function of traffic load.

While Cap-DPPM, SSCAM and POPETA begin blocking requests at 100, 200, and 300 erlangs, the BARTRMAN algorithm begins blocking requests at 375 erlangs. The difference between the BBR produced by the POPETA algorithm and that produced by the SSCAM and Cap-DPPM algorithms for loads of 300 erlangs is nearly three and four orders of magnitude, respectively.

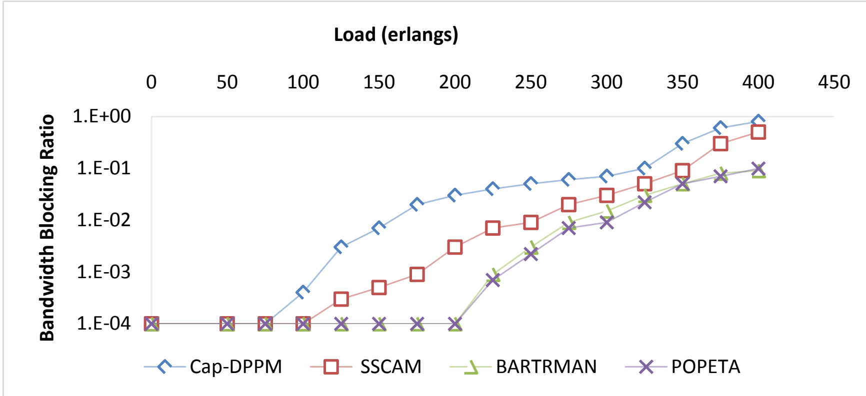


(a)

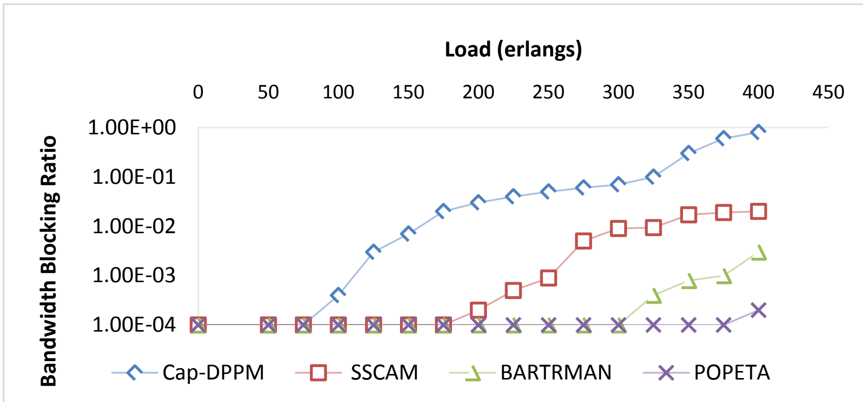


(b)

Figure 2. Topologies. Tode, H. and Hirota, Y. [5]. (a) Pan-European Topology; (b) NSF Topology.



(a)

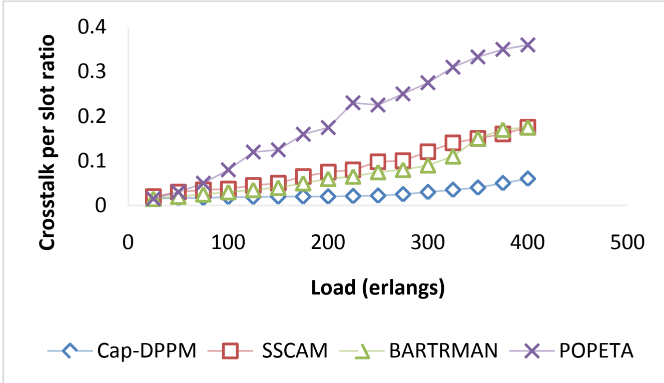


(b)

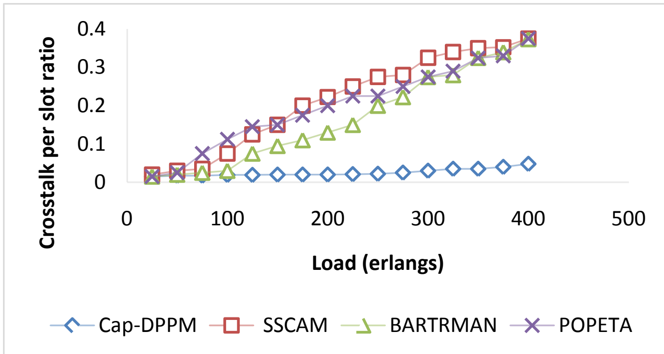
Figure 3. Bandwidth Blocking Ratio; (a) Pan-European Topology; (b) NSF Topology.

And the difference between the SSCAM algorithm's performance and that of the SSCAM algorithm is nearly one order of magnitude. This occurs as a result of the NSF topology's poor node connectivity, which causes bottlenecks. Although the gap in BBR between POPETA and BARTRMAN is over two orders of magnitude, under 400 erlangs, the disparity drops to one order. Inter-core crosstalk occurs when many cores are used. The ratio of the crosstalk index to the maximum value of the crosstalk index determines the crosstalk value associated with each spectrum slot. According to authors Fujii, S., Hirota, Y., Tode, H. [5], the average crosstalk value for all slots is used to calculate the crosstalk ratio. The crosstalk per slot (CpS) for the Pan-European architecture is shown in Figure 4(a) as a function of traffic load.

POPETA produces CpS values that are higher than those produced by the other algorithms. Figure 4(b) shows the crosstalk per slot as a function of traffic load for the NSF topology. The poor connectivity of nodes in the NSF topology results in bottlenecks and higher CpS values than those obtained by the Pan-European topology. The CpS values produced by the SSCAM method are the highest, especially for heavy loads. The CpS values obtained by the POPETA and BARTRMAN algorithms are similar to those generated by SSCAM.



(a)



(b)

Figure 4. Crosstalk per slot ratio. (a) Pan-European Topology; (b) NSF Topology.

4. Conclusions

The manuscript addressed the network Routing, Spectrum, Core, and Time Allocation (RSCTA) problem and proposes a heuristic method for facilitating the creation of lightpaths in elastic optical networks with SDM and FIPP p-cycle protection. The algorithm was tested using various topologies and traffic intensities. The outcomes were compared to the outcomes of four other approaches. The POPETA and BARTRMAN algorithms use adaptive modulation and a multi-graph representation of the spectrum. To reduce energy consumption and inter-core crosstalk in SDM-EONs, a resource allocation system based on a hybrid crosstalk solution (HCS) and crosstalk-aware POPETA algorithm are proposed. When creating a light route, this technique considers the spatial, frequency, and time domains.

In terms of performance, these two algorithms are distinguished from the SSCAM and Cap-DPPM algorithms by these two properties. Despite the fact that the BARTRMAN algorithm causes less blocking than the POPETA technique, it suffers from the same SBPP limitations as the POPETA strategy. SBPP uses only pre-planned paths, unlike the FIPP p-cycle, which uses pre-connected pathways. In the event of failure, an SBPP scheme must dynamically construct the backup, whereas a FIPP scheme's backup is already established.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Zhu, R., Zhao, Y., Yang, H., Yu, X., Zhang, J., Yousefpour, A., Wang, N. and Jue, J.P. (2016) Dynamic Time and Spectrum Fragmentation-Aware Service Provisioning in Elastic Optical Networks with Multi-Path Routing. *Optical Fiber Technology*, **32**, 13-22. https://doi.org/10.1016/j.yofte.2016.08.009

[2] Horota, A., Reis, L., Figueiredo, G. and Saldanha, F.N.L. (2016) Routing and Spectrum Assignment Algorithm with Most Fragmented Path First in Elastic Optical Networks. *IEEE Latin America Transactions*, **14**, 2980-2986. https://doi.org/10.1109/TLA.2016.7555285

[3] Zhao, J., Yao, Q., Ren, D., Li, W. and Zhang, N. (2015) A Novel Weighted Energy Efficient Routing and Spectrum Assignment Algorithm in Flexible Optical Networks. *Journal of Optical Communications*, **36**, 217-223. https://doi.org/10.1515/joc-2014-0063

[4] Yin, S., Chen, Y., Ding, S., Zhang, Z. and Huang, S. (2022) Crosstalk-Aware Routing, Spectrum, and Core Assignment based on AoD Nodes in SDM-EONs with Bidirectional Multicore Fibers. *Optical Switching and Networking*, **43**, Article ID: 100647. https://doi.org/10.1016/j.osn.2021.100647

[5] Tode, H. and Hirota, Y. (2016a) Routing, Spectrum, and Core and/or Mode Assignment on Space-Division Multiplexing Optical Networks [Invited]. *Journal of Optical Communications and Networking*, **9**, A99-A133. https://doi.org/10.1364/JOCN.9.000A99

[6] Nunes da Silva Oliveira, H.M. and Saldanha da Fonseca, N.L. (2017) The Minimum Interference p-Cycle Algorithm for Protection of Space Division Multiplexing Elastic Optical Networks. *IEEE Latin America Transactions*, **15**, 1342-1348. https://doi.org/10.1109/TLA.2017.7959516

[7] Fujii, S., Hirota, Y., Tode, H. and Murakami, K. (2014) On-Demand Spectrum and Core Allocation for Reducing Crosstalk in Multicore Fibers in Elastic Optical Networks. *Journal of Optical Communications and Networking*, **6**, 1059-1071. https://doi.org/10.1364/JOCN.6.001059

Annexes: Code

```
package
POPETAAlg;
        public class FirstFitPOPETAAlgTest {
        private static final Logger log = LoggerFactory.getLogger(FirstFitPOPETAAlgTest.class);
            private FirstFitPOPETAAlg alg;
            public void init() {
                UndirectedWeightedGraphBuilderBase builderBase = SimpleWeightedGraph.builder(EonEdge.class);
                SimpleWeightedGraph<EonVertex, EonEdge> graph = (SimpleWeightedGraph<EonVertex,
                EonEdge>)builderBase.build();
                alg = new FirstFitPOPETAAlg(graph, null, null);
            }
            public void searchLowestAvaiIndexTest() {
                ArrayList<Integer> list = new ArrayList<>();
                list.add(1);
                list.add(3);
                list.add(4);
                list.add(6);
                list.add(7);
                list.add(8);
                list.add(10);
                list.add(12);
                list.add(13);
                list.add(14);
                list.add(15);
            }
            @Test
            public void firstFitPOPETATest() {
                ArrayList<SimpleWeightedGraph<EonVertex, EonEdge>> netList = SimulationPlotline.parseNets();
                double rou = 20;
                double miu = 2;
                Calendar startTime = Calendar.getInstance();
                Calendar endTime = Calendar.getInstance();
                endTime.setTimeInMillis(startTime.getTimeInMillis());
                endTime.add(Calendar.HOUR, 24);
                int minRequiredSlotNum = 1;
                int maxRequiredSlotNum = 5;
                int startIndex = 1;
                for (SimpleWeightedGraph<EonVertex, EonEdge> graph : netList) {
```

Continued

```
                ArrayList<Integer> avaiVertexes = generateVertexList(graph.vertexSet().size());
                ServiceGenerator generator = new ServiceGenerator(avaiVertexes, rou, miu, startTime, endTime,
        minRequiredSlotNum, maxRequiredSlotNum, startIndex);
                ArrayList<Service> services = generator.generateServices();
                ServiceQueue serviceQueue = new ServiceQueue();
                serviceQueue.addServiceList(services);
                ArrayList<Timestamp> serviceOrderedQueue = serviceQueue.sortQueue();
                FirstFitPOPETAAlg ffPOPETA = new FirstFitPOPETAAlg(graph, services, serviceOrderedQueue);
                ffPOPETA.allocate();
                // data collection
                ArrayList<Double> bp = ServiceBlockingProbability.getInstance().calculateBP(startTime, endTime,
                        ffPOPETA.getPassedServices(), ffPOPETA.getBlockedServices(), 20, null);
                log.info("The BP of network is {}.", bp);
            }
        }
        private ArrayList<Integer> generateVertexList(int size) {
            ArrayList<Integer> rtn = Lists.newArrayList();
            for (int i=1; i<=size; i++) {
                rtn.add(i);
            }
            return rtn;
        }
        @Test
        public void smtTest() throws Exception{
            File file = new File
            ("USLIKENET_2017-03-12_1days_MLSPDwithFirstFitPOPETA_SpectrumMigratingTime.data");
            ObjectInputStream inputStream = new ObjectInputStream(new FileInputStream(file));
            ArrayList<Pair<Calendar, Integer>> smt = (ArrayList<Pair<Calendar,
            Integer>>)inputStream.readObject();
            log.info("size of smt is : {}.", smt.size());
        }
    }
```