Protection lightpath-based hitless spectrum defragmentation for distance adaptive elastic optical networks

Chao Wang,1 Gangxiang Shen,1,* and Limei Peng2
1School of Electronic and Information Engineering, Soochow University, China
2Department of Industrial Engineering, Ajou University, South Korea

*shengx@suda.edu.cn

Abstract: Spectrum defragmentation can improve spectrum utilization for an elastic optical network (EON). However, most of the existing studies have focused on defragmentation for working lightpaths, which may affect upper-layer network services. This paper considers protection lightpath-based hitless spectrum defragmentation for distance adaptive elastic optical networks. Without affecting working lightpaths, but defragmenting spectra for protection lightpaths, we expect to achieve truly hitless spectrum defragmentation for an EON. Shared backup path protection (SBPP) technique is employed as a representative network protection technique to evaluate the benefit of the proposed defragmentation scheme. To smooth the network spectra for future arriving lightpath requests so as to reduce bandwidth blocking probability (BBP), we propose two defragmentation triggering mechanisms, namely, defragmentation upon blocking (BTD) and batch defragmentation (BD). For each of them, we also propose two spectrum defragmentation algorithms, namely, defragmentation with sequentially releasing and re-establishing protection lightpaths (SR-D) and defragmentation with jointly releasing and re-establishing protection lightpaths (JR-D). The performances of these proposed algorithms are evaluated from perspectives of BBP and average number of reconfigurations per successfully established lightpath service (ANR). Simulation results show that compared to the case without defragmentation, the proposed scheme is effective to reduce BBP, which trades off with ANR.

©2016 Optical Society of America

OCIS codes: (060.4250) Networks; (060.4256) Networks, network optimization.

References and links

1. Introduction

Traditional wavelength division multiplexing (WDM) networks adopt fixed-size bandwidth allocation per wavelength. This leads to capacity wastage when the traffic demand is less than the capacity of an entire wavelength. To overcome this issue, researchers have been working on next-generation optical transmission systems based on elastic optical transmission techniques [1, 2]. An optical network based on this type of elastic transmission technique is called elastic optical network. Recently, the elastic optical network has received extensive interest due to its high flexibility in spectrum allocation and efficiency of spectrum utilization.

Under dynamic lightpath traffic demand, as lightpaths with different numbers of frequency slots (FSs) are established and released over time, uneven usage of FSs can lead to severe spectrum fragmentations, which will degrade spectrum utilization [3]. To overcome this issue, many spectrum defragmentation schemes have been proposed [4–16]. It is interesting to observe that most of existing studies on spectrum defragmentation have focused on working lightpaths and not considered protection lightpaths. Though efficient, working lightpath defragmentation could affect the existing service connections, which is highly not preferred by many network carriers. Meanwhile, survivability is important to an elastic optical network. Protection capacity is always required to be reserved in the network. Thus, when implementing network protection for the elastic optical network, we propose to carry out spectrum defragmentation for protection lightpaths. Compared to the working lightpath defragmentation, the key advantage of the protection lightpath defragmentation is its true “hitlessness.” This is because as long as there is no network failure, any spectrum defragmentation for protection lightpaths would not affect the existing working lightpaths as well as network services. For network protection, there are various efficient techniques such as $p$-Cycles, 1 + 1 dedicated path protection, shared backup path protection (SBPP) [17], etc. In this study, because of its operational simplicity and high spare capacity sharing efficiency, we employ SBPP as a representative technique to evaluate the benefit of protection lightpath-based spectrum defragmentation.

For working lightpath defragmentation, there can be a strategy to switch traffic from a working lightpath to its corresponding protection lightpath before carrying out defragmentation. Though this can minimize the traffic loss to an extent, the defragmentation is still not truly lossless. This is because when user traffic is switched from the working lightpath to its corresponding protection lightpath, the upper layer (e.g., IP layer) needs to do
many things for synchronization and routing table updating before it uses a new optical channel to carry traffic, which will cause traffic loss. Moreover, the defragmentation in this way would not be efficient as under the SBPP technique, only a limited number of working lightpaths can be simultaneously switched to their protection lightpaths and jointly carried out defragmentation. More specifically, only the working lightpaths that do not share common protection capacity on their corresponding protection lightpaths can be simultaneously spectrally defragmented owing to SBPP’s spare capacity sharing principle. With a small number of working lightpaths for defragmentation, a limited gain can be expected for spectrum utilization improvement.

The current work is an extended full version of the work in [8] and also different from the study focusing on 1 + 1 dedicated path protection [10]. The key contributions and novelties of this study are as follows:

1. We propose the scheme of protection lightpath-based hitless spectrum defragmentation, which compared to the working lightpath defragmentation, is advantageous of not affecting existing services. SBPP is employed as a representative protection technique to evaluate the benefit of such a defragmentation scheme.

2. To decide when to start a spectrum defragmentation, we consider and evaluate two defragmentation triggering mechanisms, i.e., BD and BTD, which are also referred to as re-active and pro-active defragmentation as in [4].

3. For each spectrum defragmentation process, we develop two defragmentation algorithms, i.e., (a) JR-D algorithm, which jointly releases and re-establishes all the existing protection lightpaths, and (b) SR-D algorithm, which sequentially releases and re-establishes all the protection lightpaths one by one.

4. Considering the different OSNR values of a pair of working and protection lightpaths, which is practical to assign different numbers of FSs for a pair of working and protection lightpaths, we incorporate the distance adaptive modulation scheme in the defragmentation process.

5. In the process of spectrum defragmentation, we consider whether we need to reassign spectrum for each existing protection lightpath according to its remaining service time and evaluate its impact on BBP and ANR.

6. Most of existing studies [7,11,13] divided the routing and spectrum assignment (RSA) problem into routing and spectrum assignment two subsequent steps, i.e., first use the K-shortest path algorithm to specify a set of candidate routes and then choose the first eligible route for spectrum assignment. In this study, we extend an integrated spectrum window plane (SWP)-based RSA approach [18] to jointly choose route and assign spectra for the protection lightpath during spectrum defragmentation.

The rest of this paper is organized as follows. In Section 2, we review the related works on spectrum defragmentation for EON. In Section 3, we introduce the concept of protection lightpath-based hitless spectrum defragmentation and defragmentation triggering mechanisms. In Section 4, we describe the heuristic algorithms for protection lightpath-based spectrum defragmentation. In Section 5, we present numerical results and analyze related network performance. Section 6 concludes the paper.

2. Review of related works

Prior works on working lightpath defragmentation [4–16] have mainly focused on the following two sub-problems: (i) when to trigger spectrum defragmentation? (ii) How to release and re-establish protection lightpaths in the process of defragmentation? For the first sub-problem, Wang et al. categorized defragmentation as either proactive or reactive [4]. The
proactive defragmentation periodically performs spectrum defragmentation [5–8]. There are many studies in this category. Yu et al. proposed a novel defragmentation scheme based on the term of spectrum compactness to maximize the profitability of defragmentation [5]. Cugini et al. evaluated a novel defragmentation scheme called push-pull, which is based on dynamic lightpath frequency retuning upon proper reconfiguration of allocated spectrum resources [6]. Zhang et al. proposed intelligent timing selection and adaptive defragmentation ratio selection methods to tackle the tradeoff between bandwidth blocking performance and operational complexity [7]. Wang et al. evaluated the benefit of protection path-based spectrum defragmentation under batch defragmentation [8].

In contrast, reactive defragmentation performs spectrum defragmentation only when there is rejection (blocking) of a lightpath request [9, 10]. Castro et al. proposed a spectrum reallocation approach called SPRESSO, which carries out the path-triggered spectrum defragmentation whenever not enough resources are found for a new connection request [9]. Wang et al. proposed a defragmentation upon blocking mechanism, which performs defragmentation only when a lightpath request is blocked [10].

For the second sub-problem, i.e., how to release and re-establish protection lightpaths in the process of defragmentation, there are also related studies in the literature [11–16]. These studies can be divided into two groups. Some studies focus on re-establishing lightpaths for all existing services [11–13]. Shakya et al. consolidated fragmented spectrum by either shifting the lightpath spectrum to a different group of subcarriers or assigning a new route for an existing connection [11]. Takagi et al. proposed a novel approach called make-before-break (MBB) to exploit the rerouting or spectrum reallocation for spectrum defragmentation [12]. Patel et al. proposed two algorithms, namely, Greedy-Defragmentation and SP-defragmentation, which reconfigure existing connections by either changing routes or assigning different wavelengths [13].

Also, in order for fewer lightpath reconfigurations, there are studies focusing on selecting only a partial set of established lightpath services for spectrum defragmentation [14–16]. Stiakogiannakis et al. proposed several spectrum defragmentation strategies, i.e., SBN, FBN, k-FBN, and iFBN, to free some occupied slots at the boundaries of a connection to form continuous FSs for the blocked connection [14]. Zhang et al. proposed an algorithm that only needs to reroute ~30% existing lightpath connections, which however can achieve a bandwidth blocking performance close to that of the greenfield scenario (100% rerouting) [15]. Luo et al. proposed a novel link-based partial defragmentation method, which can effectively reduce lightpath blocking ratio and minimize the number of lightpath reconfigurations [16].

3. Concept of protection lightpath-based spectrum defragmentation and defragmentation triggering mechanisms

In this part, we use an example to first introduce the concept of protection lightpath-based spectrum defragmentation in the context of a distance adaptive elastic optical network. We also introduce the triggering mechanisms for spectrum defragmentation, including batch defragmentation and defragmentation upon blocking.
As shown in Fig. 1(a), we assume a four-node network, in which there are three established service connections, i.e., c1, c2 and c3. Because the connections have different lightpath distances, we assign different modulation formats and correspondingly different numbers of FSs to each of the connections. As shown in Fig. 1(b), the working lightpath (A-C) of c1 is assigned with a higher level modulation format (e.g., 8-QAM) and fewer FSs (e.g., 2 FSs) because of its shorter distance and its corresponding protection lightpath (A-B-C) is assigned with a lower level modulation format (e.g., QPSK) and more FSs (e.g., 3 FSs) because of its longer distance. Similarly, the working and protection lightpaths of c2 are assigned with the modulation formats of 8-QAM and QPSK and 3 and 4 FSs, respectively. The working and protection lightpaths of c3 are assigned with the modulation formats of 8-QAM and QPSK and 3 and 5 FSs, respectively (here we allocate more subcarriers with less complex modulation; how to calculate the number of required FSs of each lightpath with different bandwidths and modulation formats will be introduced later in Section 4.1).

Moreover, as there is no common link on working lightpaths of c2 and c3, their corresponding protection lightpaths can share protection capacity (i.e., protection lightpaths of c2 and c3 share protection capacity on link B-C from index 5 to 8).

When the spectrum defragmentation is triggered, we can release all the established protection lightpaths with the remaining spectrum usage status as shown in Fig. 1(c). The working lightpaths that carry real user traffic will not be affected when protection lightpaths are implemented with spectrum defragmentation. If the JR-D spectrum defragmentation algorithm is applied for the protection lightpaths (which defragments all the protection lightpaths together and will be introduced later), we can obtain a new spectrum usage distribution as shown in Fig. 1(d), which is clearly smoother than the original usage status as shown in Fig. 1(b).

In this study, two spectrum defragmentation triggering mechanisms are evaluated, including batch defragmentation and defragmentation upon blocking. Batch defragmentation performs a spectrum defragmentation whenever a certain number of protected services in the network are released. We use a term DEF_ROUND to measure the defragmentation frequency, i.e., whenever there are DEF_ROUND protected services released, a defragmentation is triggered. In general, a smaller DEF_ROUND corresponds to more frequent spectrum defragmentation and expects to achieve better spectrum efficiency. In contrast, defragmentation upon blocking performs a spectrum defragmentation only when a lightpath request is blocked. In general, a heavier traffic load yields a higher lightpath blocking probability, which will in turn lead to more frequent spectrum defragmentation.

4. Heuristic algorithms for protection lightpath-based spectrum defragmentation

Upon triggering a spectrum defragmentation, we need to release existing protection lightpaths and then re-establish them. In this section, we focus on heuristic algorithms for such a lightpath release and re-establishment process. For better practicality, we incorporate distance adaptive spectrum assignment in the lightpath re-establishment step. In addition, we assume that optical transponders are fully tunable so that the working and protection lightpaths can be
assigned with different contiguous spectra. The SBPP protection technique is assumed for network protection, in which the protection capacity can be shared among multiple protection lightpaths as long as their corresponding working lightpaths do not share any common link(s).

4.1 Preliminaries

We represent a general elastic optical network as $G(V, E)$, where $V$ is the set of nodes and $E$ is the set of (bi-directional) fiber links between node pairs. For the dynamic lightpath service scenario, in which arrive service requests follow a Poisson distribution and the holding time of each established service follows a negative exponential distribution. A SBPP-lightpath request is represented as $CR(S, D, R, T)$, where $S$ and $D$ are the source and destination nodes of the request, $R$ is the bandwidth of the request in units of Gb/s, and $T$ is the service holding time of the request.

Table 1. FS capacities and optical reaches of different modulation formats [19].

<table>
<thead>
<tr>
<th>Modulation Format</th>
<th>FS Capacity (Gbit/sec)</th>
<th>Transparent Reach (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPSK</td>
<td>12.5</td>
<td>4000</td>
</tr>
<tr>
<td>QPSK</td>
<td>25</td>
<td>2000</td>
</tr>
<tr>
<td>8-QAM</td>
<td>37.5</td>
<td>1000</td>
</tr>
</tbody>
</table>

We assume three modulation formats (i.e., BPSK, QPSK, and 8-QAM) are employed in the study. For each of the modulation formats, we assume a maximal optical reach as shown in Table 1, in which the bandwidth of each FS is assumed to be 12.5 GHz and each row shows the FS capacity and transparent reach of the corresponding modulation format. BPSK has the longest transparent reach, but the lowest FS capacity. In contrast, 8-QAM has the highest FS capacity, but the shortest transparent reach.

Given a bit rate $R$, we can easily find the required number of FSs given a certain modulation format. The relationship between them is $F \times B \times SE \geq R$, where $F$ is the number of required FSs, $B$ is the bandwidth of each FS in units of GHz, and $SE$ is the spectrum efficiency (in units of bit/s/Hz). For BPSK, QPSK, and 8-QAM, the corresponding $SE$s are 1, 2, and 3 bit/s/Hz, respectively. The $F$ FSs are required to be spectrally contiguous along a lightpath if the sub-band virtual concatenation (VCAT) technique is not allowed. In addition, because a pair of working and protection routes normally has different physical distances, different modulation formats and numbers of FSs are often (re)assigned to them.

For each SBPP service request, we need to carry out routing and spectrum assignment (RSA) for the working and protection lightpaths. For protection lightpath-based spectrum defragmentation, we need to carry out another RSA for the protection lightpath. Thus, RSA is key to the whole SBPP service provisioning and spectrum defragmentation process.

For RSA, three important constraints should be satisfied for the EON, i.e., spectrum continuity, spectrum contiguity, and transparent reach. Spectrum continuity means that all the fiber links traversed by a lightpath must use the same set of frequency slots (FSs). Spectrum contiguity means that all the FSs assigned to a lightpath must be spectrally neighboring. Transparent reach means that in order to establish a lightpath without intermediate signal regeneration, a certain modulation format should be chosen such that its corresponding maximal transparent transmission distance is no smaller than the physical distance of the lightpath. In this study, we assume that the longest transparent reach can cover the whole network, and thus no intermediate signal regenerators are needed.

4.2 Spectrum window plane (SWP)-based RSA

RSA is key to the whole SBPP service provisioning and spectrum defragmentation process. In this study, we have applied the spectrum window plane (SWP)-based algorithm for RSA [18], which can carry out routing and spectrum assignment while easily satisfying all the above three constraints. An SWP is made up of a set of connected Spectrum Window (SW) links. The concept of SW was introduced in [20], which is defined as a set of spectrally contiguous
FSs in a fiber link. A fiber link with \( N \) FSs contains a total of \( N-S+1 \) SWs in size of \( S \) FSs. An SW is considered available only if all the contained FSs are available. For example, in Fig. 2(a) a fiber link with 28 FSs has a total of 23 SWs in size of 6 FSs and there is only one SW (i.e., SW 10) available. SWP is a network-wide concept. A physical network can be split into multiple SWPs as shown in Fig. 2(b), where in each plane a virtual link between a pair of nodes is connected if the SW in the corresponding fiber link is available.

Lightpath (re)establishment can be carried out through scanning SWPs. Given a certain lightpath service request, we can first decide the required number of FSs if the lightpath is applied with a certain modulation format given a certain data rate. According to this modulation format, we can decide the transparent reach for the lightpath. Meanwhile, according to the number of required FSs, we can decide the size of each SW, which is then used to create a list of SWPs. We can scan the whole set of SWPs to see if a route can be found between a pair of nodes on any SWP. Note that when searching for the route, we need to keep on watching the transparent reach limit to ensure that any found route is no longer than the limit. If any eligible route can be found on a SWP, we can assign the set of FSs of the SWP to the lightpath, in which the constraints of spectrum continuity and contiguity can always be met.

For each SBPP request, we should establish a pair of working and protection lightpaths. Only if both of them are established, can an SBPP service be established. In this study, we will employ the above-introduced SWP-based algorithm for working and protection lightpaths establishment. For the working lightpath, we need to search for a route on an SWP, in which all the FSs constrained in the SW links are free. For the protection lightpath, we first need to make a judgment on the sharability of each FS. Under the SBPP technique, an SW link on an SWP is unavailable for a protection lightpath if any one of the following conditions is met: (1) the corresponding physical link of the SW link is traversed by the corresponding working lightpath; (2) any contained FS in the SW link has been used by an established working lightpath; (3) any contained FS in the SW link has been used by an established protection lightpath, but this FS is not sharable. All these non-eligible SW links should be removed from the SWP.

For protection lightpath searching, it is important to maximize protection capacity sharing among protection lightpaths. To find the most efficient protection route on a SWP, we have formulated the cost for each SW link containing \( F \) FSs as follows:

\[
\sum_{i=1}^{\gamma_i} \frac{1}{\gamma_i+1}
\]

where \( F \) is the size of the SW and \( \gamma_i \) is the number of SBPP services that are sharing the \( i^{\text{th}} \) FS in the SW. It is intuitive to assign an FS shared by more protection lightpaths (i.e., a larger \( \gamma_i \)) with a lower cost such that more future requests tend to use this existing protection FS and avoid bringing in a new unused FS. Note that in (1), when \( \gamma_i = 0 \), which means that the FS is not used yet, then its cost is 1.0. Based on such cost setting, Dijkstra’s shortest path algorithm is employed to search for a protection route with the lowest cost.
In addition, based on the SWP concept, we apply the first-fit strategy to stop scanning the remaining SWPs once an eligible route is found on an SWP. Whenever an eligible route is found on an SWP, we will stop scanning the remaining SWPs for computational efficiency. This first-fit strategy has been applied for both working and protection lightpaths searching.

4.3 Algorithm for protection lightpath-based spectrum defragmentation

With the introduction to the SWP-based RSA algorithm, in this section, we focus on describing the algorithms of protection lightpath-based spectrum defragmentation, which include defragmentation with sequentially releasing and re-establishing protection lightpaths (SR-D) and defragmentation with jointly releasing and re-establishing protection lightpaths (JR-D). Next we describe these two algorithms.

For the algorithms, we have the following terms. \( \text{reqlist} \) is defined as a list to store all the current operating lightpath services existing in the network when a spectrum defragmentation is triggered. \( \text{opr} \) and \( \text{opi} \) represents the original protection lightpath route and the first FS index of the original protection lightpath, respectively. \( \text{npr} \) and \( \text{npi} \) represent the route of a newly found protection lightpath and the first FS index of the newly established protection lightpath, respectively. Finally, \( \text{rec} \) is used to record the total number of lightpath reconfigurations during the defragmentation (see Algorithm 1).

---

**Algorithm 1: SR-D**

**Input:** \( \text{reqlist} \)

1: Put all the existing protected services in \( \text{reqlist} \) in a descending order based on the maximum FS indexes of the corresponding protection lightpaths;

2: for each service in \( \text{reqlist} \) do
3: Release its protection lightpath \( (\text{npr}, \text{opi}) \);
4: for each modulation format \( (M_F) \) with a specific SE (from 8-QAM to BPSK in Table 1) do
5: Decide the number of FSs in each SWP based on the modulation format as ;
6: Create a list of SWPs, of which each has \( F \) FSs;
7: Remove all the SW links from each of the SWPs if corresponding SWs are not available;
8: for every SWP (from the lowest to highest index) do
9: \( \text{npi} \) the first FS index of the current SWP; use Dijkstra’s shortest path algorithm to search for a route with the least cost (the cost of each SW link is calculated by (1));
10: if successful and the route distance of \( \text{npr} \) is shorter than the transparent reach of the current \( M_F \) then
11: \( \text{rec}++ \);
12: end if
13: else
14: Use the corresponding spectrum of the current SWP to re-establish the protection lightpath;
15: Move to the next request;
16: end if
17: Move to the next SWP;
18: end if
19: end for
20: if \( \text{npr} = \text{Null} \) then
21: Move to the next \( M_F \);
22: end if
23: end for
24: end for
In the above algorithm, when the defragmentation is triggered, we firstly sort all the established protection lightpaths that are involved in the defragmentation in a descending order based on their maximum FS indexes (i.e., the established protection lightpath with the highest FS index first) and try to reassign spectra for each of them from the SWP with the lowest FS index. This process can fast tune down the FS indexes of the lightpaths with high FS indexes. This is because we are using the spectrum fragmentations in the SWPs whose FS indexes are the lowest to accommodate the lightpath. If a lightpath is reconfigured successfully, there is a very low chance for this lightpath to be further tuned down with its FS index (i.e., a low chance of reconfiguration) in the future since its FS index has been low. Thus, the above process can help to reduce the number of reconfigurations. In SR-D, we release and re-establish the protection lightpath of each service one by one. In the process of re-establishing the protection lightpath, we consider three modulation formats, i.e., PBSK, QPSK, and 8-QAM. For each of the modulation formats, we calculate the required number of FSs by \[ F = \lceil \frac{R}{(B \times SE)} \rceil. \] We create a list of SWPs, of which each contains \( F \) continuous FSs. We then remove all unavailable SW links on each SWP, if any SW link meets any one of the three conditions mentioned in Section 4.2. Then, we scan all the SWPs, and use Dijkstra’s algorithm to search for a least-cost route based on the link cost calculated by (1). If an eligible route is found on an SWP, we will terminate the searching process and use the corresponding spectrum of the current SWP to re-establish the protection lightpath. Because we can reassign its original used spectrum to each service in the worst case, SR-D algorithm can always ensure to successfully re-establish all the protection lightpaths. In addition, after re-establishing the protection lightpath for a service, we will check whether a lightpath reconfiguration is required. Specifically, either the route or the spectrum of a lightpath is changed, we say that a reconfiguration is required and will accumulate term \( \text{rec} \) by one.

The JR-D algorithm is similar to the SR-D algorithm, except that we jointly release all the protection lightpaths and then re-establish the protection lightpaths for each service. Before re-establishing the protection lightpaths, we sort them based on the numbers of required FSs from the largest to the smallest and remaining service times (RSTs) from the largest to the smallest. Here RST is defined as a time for a service connection to elapse before it completes its mission. It is a relative value to the average holding time for each service connection. For example, if the average holding time of each service connection is 4 days and RST is 0.25, then the actual remaining service time of a connection is 4 \times 0.25 = 1 day. In addition, an enhancement is made to JR-D, that is, if it is impossible to re-establish all the protection lightpaths upon the joint release and re-establishment process, we will recover the spectra of all the protection lightpaths to their original status and then refer to the SR-D algorithm to defragment spectra for the protection lightpaths. Because of this step, the JR-D algorithm can always outperform the SR-D algorithm. The detail of the JR-D algorithm is as follows (see Algorithm 2).

```
Algorithm 2: JR-D
Input: reqlist
1. Put all the established SHPP services in reqlist based on the numbers of required FSs from the largest to the smallest and remaining service times from the largest to the smallest;
2. Release all the established protection lightpaths;
3. for each service in reqlist do
4.    Replaced with steps from 4 to 23 in the SR-D algorithm;
5.    if npr == Null then
6.    Release all previous re-established protection lightpaths and restore original spectra for them;
7.    End JR-D and subsequently implement the SR-D algorithm;
8.    Break;
9.    end if
10. end for
```
To better understand the performance of the two algorithms, we use a four-node network example as shown in Fig. 1(a). Assume each link has 10 FSs. Figure 1(b) shows the current network spectrum usage status. Working lightpaths of services c₁ and c₂ are (A-C) and of service c₃ is (B-D) and corresponding protection lightpaths of services c₁ and c₂ are (A-B-C) and of service c₃ is (B-C-D). We also assume that the remaining service times of c₁, c₂, and c₃ are 0.8, 0.3, and 0.5 day, respectively. Figures 3(a) and 3(b) show the status of network spectrum usage after implementing SR-D and JR-D, respectively. In SR-D, we sequentially release the protection lightpaths of c₃, c₂, and c₁ (according to their highest used protection FS indexes), and then re-establish the protection lightpaths for them. In JR-D, we release the protection lightpaths of c₁, c₂, and c₃ altogether, and then sequentially re-establish the protection lightpaths for c₃, c₂, and c₁ (according their required numbers of FSs and remaining service times from the largest to the smallest). Comparing the spectrum usage in Fig. 3, we can see that JR-D can achieve smoother FS usage status than that of SR-D after defragmentation because JR-D first jointly releases all the protection spectra before protection lightpath re-establishment, which provides more opportunities for subsequent lightpath re-establishment optimization. However, as a disadvantage, because JR-D releases all the protection lightpaths when implementing defragmentation, there is a risk that multiple of working lightpaths cannot be recovered if there is a sudden failure in the course of defragmentation. In contrast, for SR-D, there is at most one un-recovered working lightpath if such a kind of situation occurs.

4.4 Computational complexity

The pseudocode of two proposed protection lightpath-based spectrum defragmentation algorithms (i.e., SR-D and JR-D) is shown in Algorithms 1-2. In Algorithm 1, line 7 removes all unavailable SW links on each SWP, whose computational complexity is O(WM), where W is the total number of FSs in each fiber link and M is the total number of links in a network. We note that the computational complexity of lines 8-19 is dominated by the complexity of the Dijkstra’s shortest path searching algorithm in line 9. With the possibility of scanning all the SWPs, its computational complexity is O(WN²), where N is the total number of network nodes. Meanwhile, we are considering multiple types of modulation formats in the for-loop of line 4. Thus, the overall computational complexity of the SR-D algorithm is O(TW(M + N²)), where T is the total number of available modulation formats. For the JR-D algorithm, when any protection lightpath cannot be established on any SWP, we will subsequently implement SR-D. Thus, the overall computational complexity of the algorithm is O(2TW(M + N²)).
4.5 Discussion

**Triggering mechanisms:** We have considered the two defragmentation triggering mechanisms, i.e., BTD and BD. These two triggering mechanisms can work with any of the defragmentation algorithms, including SR-D and JR-D. Thus, in this study we will have four combinations of protection lightpath-based spectrum defragmentation scenarios.

**Remaining service time threshold (RSTH):** It is not meaningful to consider a service connection for defragmentation if its remaining service time has been pretty small. Thus, in the defragmentation algorithms, we can set a threshold for remaining service time of the connections that can be involved in defragmentation. This can help tradeoff the performance in terms of bandwidth blocking probability (BBP) and average number of protection lightpaths reconfigurations per successfully established service (ANR). To incorporate this measure in the algorithms, we can make a pre-judgement for each service connection in reqlist. Only if the RST of a connection is larger than RSTH, this connection can be involved in defragmentation, i.e., executing lines 3-23 in algorithm 1 or lines 4-9 in algorithm 2.

5. Performance evaluations and result analyses

In this part, we firstly present the test conditions for the simulation study. Then we compare the performance of the proposed heuristic defragmentation algorithms with the case without defragmentation. We also compare the two triggering mechanisms from the performance perspectives of BBP and ANR. Finally, we also evaluate the impact of RSTH on the performance in terms of BBP and ANR.

5.1 Test conditions

A dynamic lightpath traffic model is assumed in this study. Specifically, lightpath requests arrive according to a Poisson arrival process and their holding times follow a negative exponential distribution. A total of $10^6$ lightpath arrival requests were simulated for calculating BBP and ANR. The intensity of arrived service requests is measured in units of “Erlang.” Each of the requests can have different bandwidth requirement in units of Gb/s. We assume that the required bandwidth of each connection request is uniformly distributed within a range from 10 Gb/s to 400 Gb/s, based on which the required number of FSs can be found for a certain modulation format. In this study, all the service connections are protected. We do not consider the case of partial protection. When provisioning a lightpath service, if either working or protection lightpath cannot be successfully established, we will block the service request.

Let $AR$ represent the set of all the arrived requests and $BR$ represent the set of all the blocked requests. Let $a_k$ be the bandwidth required by the $k^{th}$ arrived request in $AR$. As an important evaluation criterion, BBP is defined as

$$BBP = \frac{\sum_{k \in BR} a_k}{\sum_{k \in AR} a_k} \quad (2)$$

where the numerator sums the bandwidth of all the blocked requests and the denominator sums the bandwidth of all the arrived requests. In addition, if we change the protection route or assign different set of FSs to a protection lightpath in the course of spectrum defragmentation, a reconfiguration is required for the protection lightpath. Thus, ANR denotes the average number of lightpath reconfigurations for each successfully established service, which is defined as the ratio of the total number of lightpath reconfigurations during $10^6$ simulated arrival requests divided by the total number of successfully established services.

We consider two test networks: (i) the 14-node, 21-link NSFNET network and (ii) the 11-node, 26-link COST239 as shown in Fig. 4. The distance of each link (in km) is shown by the link. We assume that there are 320 FSs in each fiber link in both of the networks with the
bandwidth granularity of each FS being 12.5 GHz. In addition, three modulation formats (i.e., BPSK, QPSK, and 8 QAM) are assumed for both the working and protection lightpaths.

Based on SWPs, we use the Dijkstra’s shortest path algorithm to search for working and protection routes with the least cost and use the first-fit strategy to (re)assign spectrum. We also consider two spectrum defragmentation mechanisms, i.e., batch defragmentation and defragmentation upon blocking. For batch defragmentation, we set different numbers of DEF_ROUND to change the defragmentation frequency. For each triggering mechanism, we also consider the two spectrum defragmentation algorithms, i.e., SR-D and JR-D.

![Two test networks](image)

(a) 14-node, 21-link NSFNET network  
(b) 11-node, 26-link COST239 network

**Fig. 4. Two test networks**

### 5.2 Spectrum defragmentation vs. without defragmentation

We first evaluate the benefit of spectrum defragmentation compared to the case without defragmentation. Figure 5 shows the lightpath bandwidth blocking performance of the different spectrum defragmentation algorithms. We can see that no matter under which defragmentation triggering mechanism, the proposed protection lightpath-based defragmentation algorithms (i.e., SR-D and JR-D) can always significantly outperform the case without defragmentation to demonstrate much lower BBPs. In addition, comparing the two spectrum defragmentation algorithms, we see that JR-D can outperform SR-D because the former jointly releases the spectra of all the existing protection lightpaths, which creates more continuous FSs available for protection lightpath re-establishment. Moreover, in the protection lightpath re-establishment process, JR-D sorts the SBPP services according to their required FSs and remaining service times. This provides more opportunities for services with similar FSs and remaining service times to share common protection FSs. As a result, JR-D achieves a better BBP performance than SR-D.

![Bandwidth blocking performance comparison](image)

(a) COST239  
(b) NSFNET

**Fig. 5. Bandwidth blocking performance comparison between the proposed algorithms and the case without defragmentation (SR: SR-D; JR: JR-D; BTD: defragmentation upon blocking; BD: batch defragmentation)**

In addition, for the same spectrum defragmentation algorithm, either JR-D or SR-D, we see that the BD triggering mechanism outperforms the BTD mechanism for the given DEF_ROUNDs in Fig. 5. It is could be explained by the reason that the DEF_ROUND
numbers used for the simulations have been small enough to enable spectrum defragmentations more frequent than that triggered by lightpath service blocking. We can expect that, with a larger DEF_ROUND, the related performance would become worse, while with a smaller DEF_ROUND, the related performance will be further improved. For this, we will evaluate how the value of DEF_ROUND can affect the BBP performance of the BD mechanism in the next sub-section.

5.3 Batch defragmentation

In this section, we evaluate how the value of DEF_ROUND can affect the BBP performance of the BD mechanism. In Fig. 6, we can see that the BBP performance is closely related to the defragmentation frequency measured by DEF_ROUND when batch defragmentation is applied. A decreasing DEF_ROUND corresponds to more frequent defragmentations, leading to a lower BBP. However, when the DEF_ROUND reaches a certain value, further decrease of the DEF_ROUND would not significantly improve the bandwidth blocking performance. For the cases when DEF_ROUND = 10 and DEF_ROUND = 20 in COST239, and when DEF_ROUND = 5 and DEF_ROUND = 10 in NSFNET, we see that the different DEF_ROUNDs achieve close bandwidth blocking performance.

![Fig. 6. Bandwidth blocking performance comparison between different DEF_ROUNDs](image)

On the other hand, more frequent defragmentations correspond to more protection lightpath reconfigurations. We also evaluate how the number of protection lightpath reconfigurations change with a decreasing DEF_ROUND, i.e., more frequent spectrum defragmentations. As shown in Fig. 7, we can see that the ANR becomes larger with more frequent spectrum defragmentations. In particular, it is interesting to observe the two cases (i.e., DEF_ROUND = 10 and DEF_ROUND = 20 for COST239 or DEF_ROUND = 5 and DEF_ROUND = 10 for NSFNET) yield very large differences between the average number of reconfigurations, but they perform closely in bandwidth blocking performance. This observation implies that it is not worth to implement very frequent spectrum defragmentations to achieve a good BBP. Rather, we should balance the BBP performance improvement and the number of lightpath reconfigurations for the control of network control overhead. In the
following study, we choose DEF ROUND = 20 and 10 respectively for COST239 and NSFNET, for a reasonably good BBP and the control of defragmentation frequency.

5.4 Batch defragmentation vs. defragmentation upon blocking

We also compare the performance of the two spectrum triggering mechanisms, i.e., BD and BTD. Figure 8 shows BBP and ANR with an increasing traffic load per node pair. For performance evaluation purposes, the JR-D algorithm was applied for both of the defragmentation triggering mechanisms.

We can see that when traffic load per node pair is low (i.e., TL = 5.5 or TL = 5.8 Erlang for COST239 and TL = 1.25 or TL = 1.33 Erlang for NSFNET), BD achieves better BBP than BTD. This is because the DEF ROUND numbers used for the simulation studies are small enough to enable spectrum defragmentations more frequent than that triggered by lightpath service blocking. On the other hand, under the BD mechanism, the ANR is much larger than that of BTD.

When the traffic load per node pair becomes higher, the bandwidth blocking performances of BTD and BD become very closely or even the latter achieves better performance. This is because a higher lightpath blocking probability corresponds to more frequent spectrum defragmentations for the BTD mechanism, which helps enhance the bandwidth blocking performance. Moreover, as an important advantage, compared to the BD, the BTD shows fewer lightpath reconfigurations per successfully established service. For BD, its defragmentation frequency is less dependent on the traffic load. However, BTD performs defragmentation upon a blocked lightpath request. Thus, when traffic load per node pair becomes larger, more lightpath requests will be blocked and there are more frequent defragmentations. In addition, we can see that when BTD is applied, although the average number of reconfigurations increases with the increase of traffic load per node pair, it is always lower than that of BD. Thus, considering required lightpath reconfigurations, it would be preferred for a network to apply the BTD mechanism when the traffic load is high.

![Fig. 8. BBP and ANR performance comparison between two triggering mechanisms (i.e., defragmentation upon blocking and batch defragmentation)](image)

5.5 Impact of remaining service time threshold

We also evaluated how the performance in terms of BBP and ANR will tradeoff with different RSTH values. The two proposed algorithms (i.e., JR-D and SR-D) were evaluated under the strategy of BTD.

In a general sense, a larger RSTH would correspond to a smaller ANR, but a higher BBP. Table 2 shows the result of COST239 under the SR-D algorithm, which trades off between BBP and ANR for different RSTH values. As expected, we can see that with an increasing RSTH, ANR reduces while BBP increases. This is because a larger RSTH requires fewer connections to be involved defragmentation, thereby resulting in a smaller ANR and meanwhile a higher BBP because fewer connections correspond to a lower chance for spectrum re-optimization. In addition, at a high traffic load (e.g., TL = 6.7 Erlang), it seems that the reduction of RSTH does not show a strong impact on BBP. This is because at a high traffic load, much spectrum resources have been occupied. With the SR-D algorithm, in which
each time only one connection is reconfigured, while all the other connections are untouched though they may be reconfigured subsequently, the chance for the reconfigured connection to find a better spectrum allocation is low given very strict free resources at a high traffic load.

Similar performance evaluation was carried out for the JR-D algorithm. Table 3 shows the results in terms of BBP and ANR for different RSTH values. We notice that there is a similar trend between BBP/ANR and RSTH, i.e., an increasing RSTH, there are a higher BBP and a smaller ANR. However, different from the results of SR-D, we notice that BBP improvement seems more significant than that of SR-D at a high traffic load (e.g., TL = 6.7 Erlang) when reducing the RSTH value. This is because the JR-D algorithm implements joint optimization that release all the connections involved in defragmentation and then re-optimize them in spectrum allocations. As a result, there is a better chance of spectrum resource optimization, thereby performance improvement with a reduced RSTH. Similar results were obtained for the other test networks, which are however not presented for space saving.

Table 2. BBP and ANR under different RSTH (BTD + SR-D, COST239).

<table>
<thead>
<tr>
<th>TL = 5.5</th>
<th>TL = 6.1</th>
<th>TL = 6.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBP</td>
<td>ANR</td>
<td>BBP</td>
</tr>
<tr>
<td>BTD SR (RSTH = 0.1)</td>
<td>0.00184</td>
<td>0.24</td>
</tr>
<tr>
<td>BTD SR (RSTH = 0.2)</td>
<td>0.00212</td>
<td>0.22</td>
</tr>
<tr>
<td>BTD SR (RSTH = 0.3)</td>
<td>0.0023</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Table 3. BBP and ANR under different RSTH (BTD + JR-D, COST239).

<table>
<thead>
<tr>
<th>TL = 5.5</th>
<th>TL = 6.1</th>
<th>TL = 6.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBP</td>
<td>ANR</td>
<td>BBP</td>
</tr>
<tr>
<td>BTD JR (RSTH = 0.1)</td>
<td>0.00071</td>
<td>0.54</td>
</tr>
<tr>
<td>BTD JR (RSTH = 0.3)</td>
<td>0.00119</td>
<td>0.48</td>
</tr>
<tr>
<td>BTD JR (RSTH = 0.5)</td>
<td>0.00171</td>
<td>0.43</td>
</tr>
</tbody>
</table>

6. Conclusions

We proposed protection lightpath-based hitless spectrum defragmentation for distance adaptive elastic optical networks with shared backup path protection (SBPP). Two spectrum defragmentation mechanisms, i.e., defragmentation upon blocking and batch defragmentation, were considered. Also, two spectrum defragmentation algorithms, namely, SR-D and JR-D, were proposed. Through simulation studies, we find that the proposed protection lightpath-based hitless spectrum defragmentation scheme can significantly improve the bandwidth blocking performance compared to the case without defragmentation, no matter which triggering mechanism and defragmentation algorithm are applied. In addition, the blocking performance improvement trades off with the number of lightpath reconfigurations. We can also see that JR-D can achieve better bandwidth blocking performance than SR-D because it jointly releases all the existing protection lightpaths to achieve smoother spectrum usage status for arriving lightpath requests. In addition, when DEF_ROUND reaches a certain value, further decrease of DEF_ROUND would obviously increase number of reconfigurations, but not significantly improve the bandwidth blocking performance. Considering bandwidth blocking performance improvement and required number of lightpath reconfigurations, our results shows that defragmentation upon blocking is efficient than batch defragmentation when the network traffic load is high. Finally, we also consider remaining service time, which can reduce lightpath reconfigurations trading off increasing bandwidth blocking probability.

Acknowledgments

This work was jointly supported by NSFC (61322109, 61172057, 61307082), NSF of Jiangsu Prov. (BK20130003, BK2012179), Science and Technology Support Plan of Jiangsu Province (BE2014855), and partly supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (2015R1C1A1A02036536).