

Dynamic Routing and Spectrum Assignment with Non-Disruptive Defragmentation

David Coudert^{1,2} and Brigitte Jaumard³ and Fatima Zahra Moataz^{1,2} †

¹Univ. Nice Sophia Antipolis, CNRS, I3S, UMR 7271, 06900 Sophia Antipolis, France

²INRIA, France

³Concordia University, Montréal, Canada

L'augmentation du trafic Internet motive une évolution des réseaux WDM traditionnels vers les *réseaux optiques élastiques* (Elastic Optical Networks, EON). Les EONs sont conçus pour optimiser l'utilisation des ressources optiques; ils permettent l'attribution de bandes de largeurs quelconques et offrent ainsi plus de souplesse que les réseaux WDM avec leurs bandes fixes. Cependant, la gestion du spectre dans les EONs devient plus difficile du fait de la fragmentation. Dans ce papier, nous proposons deux algorithmes qui permettent de résoudre le problème de Routage et d'Allocation de Spectre d'une nouvelle requête en utilisant une technique de défragmentation non-perturbatrice.

Keywords: Algorithms, Optical Networks, Routing and Spectrum Assignment, Defragmentation

1 Introduction

Elastic Optical Networks (EONs) [GJLY12] is the new buzzword in the optical community. This new networking paradigm promises a better utilization of the spectrum in optical networks. In fact, as the optical transmission spectrum is carved into fixed-length bands in the traditional WDM networks, small bit rates are over-provisioned and very high bit rates do not fit. EONs are moving away from this fixed-grid and allow the spectrum to be divided flexibly: each request is allocated exactly the resources it needs.

The flexibility of EONs makes better use of the available spectrum. However, it also makes some of the spectrum management problems more challenging as it is the case for *fragmentation*. Fragmentation is the accumulation of small fragments of spectrum over time due to the dynamic traffic in the network (like the fragmentation of a computer hard disk). Since those fragments are non-contiguous, a new request might be blocked even if the available bandwidth in the spectrum could satisfy it. Many techniques have been proposed to address fragmentation [WM14]. The *preventive techniques* route and allocate spectrum to requests in a way to give new requests more opportunities to be accepted. The *remedial techniques*, on the other hand, offer a cure to fragmentation after it happens: when a new request cannot be provisioned under current circumstances, defragmentation is used to consolidate the small fragments and free some space. Defragmentation can disrupt the system by changing the route of some already provisioned requests or it can be *non-disruptive* as the new proposed technique, Push-Pull [CPM⁺13]. With Push-Pull, requests are shifted in the spectrum interval and a request does not change its path nor transgresses other established requests. The delay of insertion of a new request using Push-Pull indicates the duration of the shifting done to free the needed space. Mukherjee and Wang [WM13] define it as the number of spectrum slots through which the shifting is done and consider two types of parallelism to compute it as illustrated in Fig. 1.

An algorithm is proposed in [WM13] to route and allocate spectrum to a new request in EONs; it starts by pre-computing a set of paths for the request and then finds on each path a position that minimizes the delay using Push-Pull. This algorithm is not exact because it does not ensure that a path and a position are returned whenever there is a solution. Furthermore, for a given path, the algorithm does not minimize the delay of insertion over the whole space of possible positions on the path but only over a subset of that space.

†This author is supported by a grant from the "Conseil régional Provence Alpes-Côte d'Azur"

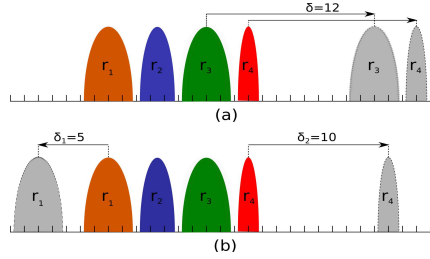


Fig. 1: Push-Pull shiftings (a) r_3 and r_2 are both shifted by 12 spectrum slots in the same direction. The delay is then $\delta = 12$. (b) r_1 is shifted by $\delta_1 = 5$ spectrum slots and r_4 is shifted in the opposite direction by $\delta_2 = 10$ spectrum slots. Since the two operations can be done in parallel the delay is $\delta = \max\{\delta_1, \delta_2\} = 10$

Contributions. We present two exact algorithms to route and allocate spectrum to a new request in an EON using only Non-Disruptive Defragmentation (Push-Pull). In the first algorithm (Section 3), we find the shortest routing path for the new request (i.e., the shortest path from source to destination where contiguous spectrum to satisfy the request can be freed) and then find the position that gives the overall minimum delay on that path. In the second algorithm (Section 4), we find at the same time a routing path and a position in the spectrum, that minimize the delay of insertion (over all other paths and positions). Both algorithms are polynomial in the size of the network, its bandwidth and the number of provisioned requests.

2 Notations and Definitions

We will use some of the notations used in [WM13] for the sake of conformity. $G = (V, L)$ is an optical network where V is the set of nodes and L is the set of (directed) links. R is the set of provisioned requests. b_w is the available bandwidth on every link of the network. b_r is the bandwidth traffic requirement for request $r \in R$. $s(r)/e(r)$ are the starting/ending spectrum slots of r in the spectrum interval (we start from slot 0). The Δ -state (resp. ∇ -state) [WM13] is the state of the network after shifting all the requests down (resp. up) towards slot 0 (resp. slot b_w) until they are blocked. $s_\Delta(r)/e_\Delta(r)$ (resp. $s_\nabla(r)/e_\nabla(r)$) are the starting/ending spectrum slots of r in the Δ -state (resp. ∇ -state), i.e., when all existing requests are shifted to their lowest (resp. highest) spectrum position. To keep track of the dependency between the requests we build the Spectrum Dependency Graph $D = (V^R, E^R)$. It is a DAG with two special nodes *floor* and *ceiling* and where each node v_r of $V^R \setminus \{\text{floor}, \text{ceiling}\}$ is associated with a request $r \in R$. For each node v_r , there exists an arc from *ceiling* to v_r and an arc from v_r to *floor*. There exists an arc from v_r to $v_{r'}$ if r' is assigned the band with the highest top index smaller than the bottom index of r for a given $\ell \in L$. A request r_i is constrained to be below (resp. above) another request r_j if on the spectrum dependency graph, r_i is in a path from r_j to *floor* (resp. a path from *ceiling* to r_j); the position of r_i cannot be bigger (resp. smaller) than the position of r_j , under any shifting.

We call *absolute position*, a position in the spectrum range, i.e., a value in the interval $\llbracket 0, b_w - 1 \rrbracket$. We call a *relative position* (A, B) , a position between two sets of requests: allocating position (A, B) to a request r means that r is above the set A of requests, and below the set B of requests in the spectrum range. To every relative position (A, B) we associate a *complete* relative position (A_c, B_c) such that A_c contains the requests in A and all the requests constrained to be below them, and B_c contains the requests in B and all the requests constrained to be above them. We say that two relative positions (A, B) and (C, D) are *conflicting* iff $A_c \cap D_c \neq \emptyset$ or $C_c \cap B_c \neq \emptyset$. The absolute position a can be freed for request q on link ℓ if the requests on ℓ can be shifted to empty the interval $[a, a + b_q]$. A relative position (A, B) is *associated* with the absolute position a for request q if $\max\{e_\Delta(r) | r \in A\} \leq a$ and $a + b_q \leq \min\{s_\nabla(r) | r \in B\}$.

3 Dynamic RSA over the Shortest Path

3.1 Routing over the Shortest Path

We solve the problem sequentially. First, we find the shortest path from source to destination where contiguous spectrum to satisfy the new request can be freed (Problem 1) and then we find the position minimizing the insertion delay on that path (Problem 2).

Problem 1. Given a network $G = (V, L)$, a set of provisioned requests R and a new request q , find the shortest routing path for q knowing that only Non-Disruptive Defragmentation can be used.

Idea of the algorithm. There are $\text{bwt} - b_q + 1$ possible absolute positions on the spectrum to route q . For every absolute position, there are many possible relative positions on every link. If for an absolute position a , there is an st -path P whose links have non-conflicting relative positions corresponding to a , then q can be routed on P occupying position a in the spectrum range. In our algorithm, if it is possible to free $a \in \llbracket 0, \text{bwt} - b_q \rrbracket$ on link $\ell \in L$, we color ℓ with color a (a link can receive many colors). Afterwards, we find the shortest monocolored path (i.e., whose links share a color). We do not keep track of the relative positions used to free an absolute position thanks to the following lemma.

Lemma 1. If the absolute position a can be freed on a set of links S , then there are valid non-conflicting relative positions on the links of S , associated with a .

3.2 Minimum Delay Spectrum Assignment over a Path

Problem 2. Given a network $G = (V, L)$, a set of provisioned requests R and a new request q with its st -path P , find a position with minimum delay in the spectrum for q , knowing that only Non-Disruptive Defragmentation can be used.

As in [WM13], we denote by $CS(P)$ the set of provisioned requests that use paths sharing some links with P . Let $n = |CS(P)|$. Every position of the new request q corresponds to a partition $A \cup \bar{A}$ of $CS(P)$ (requests above and below q). The floors of a position $y = (A, \bar{A})$ before and after defragmentation are defined respectively as: $f(y) = \max\{e(x) : x \in A\}$ and $f^*(y) = \max\{e_\Delta(x) : x \in A\}$. The ceilings of y are determined by \bar{A} : $c(y) = \min\{s(x) : x \in \bar{A}\}$ and $c^*(y) = \min\{s_\nabla(x) : x \in \bar{A}\}$. The sizes of y before and after defragmentation are given by $b(y) = c(y) - f(y)$ and $b^*(y) = c^*(y) - f^*(y)$. To check if we can provision request q on a position y , it is enough to check if $b^*(y) \geq b_q$ and the delay of insertion in position y will be:

$$\text{Delay}(y) = b_q - b(y) - \min\{f(y) - f^*(y), c^*(y) - c(y), (b_q - b(y))/2\} \quad (1)$$

Let $CS(P)$ be sorted as $\langle r_1, r_2, \dots, r_n \rangle$ in the ascending order of the requests spectrum occupancy in the Δ -state, i.e., $e_\Delta(r_1) \leq \dots \leq e_\Delta(r_n)$. We define the decision-positions of q over P as the $n + 1$ positions marked by \dagger in $\langle \dagger r_1 \dagger r_2 \dagger \dots \dagger r_n \dagger \rangle$ and denoted by y_0, y_1, \dots, y_n . Mukherjee and Wang [WM13] have proven that a request q can be provisioned over a path P if and only if there is $i \in \{0, n\}$ such that $b^*(y_i) \geq b_q$. Using this fact, they have designed an algorithm that finds a position for q on P by checking only the $n + 1$ decision-positions. Indeed, whenever it is possible to route q over P , their algorithm chooses among the $n + 1$ decision-positions the one that minimizes the delay of insertion. However, the chosen position is not necessarily the one that minimizes the delay over all possible positions on P , and examples can be designed where it is not. Using the following lemma, we have modified their algorithm to find a position that minimizes the delay on P over all possible positions.

Lemma 2. For every decision-position $y = A \cup \bar{A}$ such that $\langle z_1, z_2, \dots, z_k \rangle$ is A sorted in the descending order of the request spectrum occupancy $e(z)$, we define the following positions $y^\ell = (\{A \setminus \{z_1, z_2, \dots, z_\ell\}\}, \{\bar{A} \cup \{z_1, z_2, \dots, z_\ell\}\})$, $\ell \in \{1, \dots, k\}$, and $y^0 = y$. Any position on path P with delay d can be transformed into a position y^l , $l \in \{0, \dots, k\}$, with delay $d' \leq d$ for some decision-position y .

Idea of the proof. Let $x = B \cup \bar{B}$ be any valid position with delay d for request q on path P . In the proof of Theorem 1 of [WM13], x is transformed into one of the decision-positions by shifting down some of the requests of \bar{B} . This shifting may affect the delay d (see (1)). The idea then is to shift up back some of the requests that might increase the delay.

Algorithm 1 uses Lemma 2 to solve Problem 2.

4 Dynamic RSA with Minimum Delay

Problem 3. Given a network $G = (V, L)$, a set of provisioned requests R and a new request q , find for q a routing path and a position in the spectrum, that minimize the delay of insertion, knowing that only Non-Disruptive Defragmentation can be used.

The delay of freeing position a on link ℓ for request q with relative position (A, B) is given by the formula: $\text{Delay}_\ell(A, B) = \max\{0, e(A) - a, a + b_q - s(B)\}$ where $e(A) = \max\{e(x) : x \in A\}$ and $s(B) = \min\{s(x) : x \in B\}$. The delay of insertion in position a on a path P is $\text{Delay}_P = \max_{\ell \in P} \text{Delay}_\ell(A_\ell, B_\ell)$ where (A_ℓ, B_ℓ) is the relative position used to free a on link ℓ . The proposed algorithm will be based on the following 2 lemmas.

Algorithm 1 Finding Position with Minimum Delay on Path P **Require:** Network $G = (V, L)$, a set of provisioned requests R and a new request q with a path P **Ensure:** The position with minimum delay for q on P

- 1: $pos := \emptyset$ and $delay := \infty$
- 2: Find $CS(P)$ the set of requests conflicting with q on P and sort it in the ascending order of e_{Δ} . The sorted list is $\langle r_1, r_2, \dots, r_n \rangle$. The corresponding decision-positions are y_0, y_1, \dots, y_n and $y_i = (A_i, \bar{A}_i)$
- 3: **for all** $i \in \{0, \dots, n\}$ **do**
- 4: **if** $b^*(y_i) \geq b_q$ **then**
- 5: Sort the requests in A_i in the descending order of $e(x)$. The sorted list is $\langle z_1, z_2, \dots, z_k \rangle$ and $y_i^{\ell} = (\{A_i \setminus \{z_1, z_2, \dots, z_{\ell}\}\} \cup \{\bar{A}_i, \{z_1, z_2, \dots, z_{\ell}\}\})$ for $\ell \in \{1, \dots, k\}$ and $y_i^0 = y_i$
- 6: **for all** $\ell \in \{0, \dots, k\}$ **do if** $Delay(y_{\ell}^i) < delay$ **then** $pos := y_{\ell}^i$ and $delay := Delay(y_{\ell}^i)$

Lemma 4. For an absolute position a and a link ℓ , there are at most two relative positions freeing a on ℓ with minimum delay and if there are two such positions they are of the form (A, B) and $(A \cup \{x\}, B \setminus \{x\})$, where x is a request using ℓ .

If a can be freed on ℓ with minimum delay using relative positions (A, B) and $(A \cup \{x\}, B \setminus \{x\})$, we choose (A, B) to be called *the* relative position freeing a with minimum delay.

Lemma 5. For an absolute position a and two links ℓ and ℓ' , if (A, B) and (C, D) are *the* relative positions that free a with minimum delay on ℓ and ℓ' , respectively, then, (A, B) and (C, D) are not conflicting.

Algorithm 2 Dynamic RSA with Minimum Delay**Require:** Network $G = (V, L)$, a set of provisioned requests R and a new request q .**Ensure:** A path P and a position pos with minimum delay for request q .

- 1: $P := \emptyset$, $pos := \emptyset$ and $delay := \infty$
- 2: **for** $\ell \in L$ **do** $Delay_{\ell} := \infty$; sort requests using ℓ in the increasing order of their spectrum occupancy; the sorted list $\langle r_1^{\ell}, \dots, r_{d_{\ell}}^{\ell} \rangle$ and the corresponding relative positions $\langle p_0^{\ell}, \dots, p_{d_{\ell}}^{\ell} \rangle$.
- 3: **for all** $a \in \llbracket 0, \text{bwt} - b_q \rrbracket$ and $\ell \in L$ and $i \in \llbracket 0, d_{\ell} \rrbracket$ **do**
- 4: **if** $[a, a + b_q] \subset [e_{\Delta}(r_i), s_{\nabla}(r_{i+1})]$ **then**
- 5: **if** ℓ is not colored with a **then** Color link ℓ with color a
- 6: **if** $Delay_{\ell} > Delay_{\ell}(p_i^{\ell})$ **then** $Delay_{\ell} := Delay_{\ell}(p_i^{\ell})$
- 7: Find shortest path P_a colored with a that minimizes $Delay_{P_a} = \max\{Delay_{\ell} : \ell \in P_a\}$
- 8: **if** $Delay_{P_a} < delay$ **then** $P := P_a$, $delay := Delay_{P_a}$ and $pos := a$

5 Conclusion

As future work, we intend to do simulations with our algorithms to measure their performance. We would like also to examine possible trade-offs between the length of the routing path and the delay of insertion.

References

- [CPM⁺13] F. Cugini, F. Paolucci, G. Meloni, G. Berrettini, M. Secondini, F. Fresi, N. Sambo, L. Poti, and P. Castoldi. Push-pull defragmentation without traffic disruption in flexible grid optical networks. *Journal of Lightwave Technology*, 31(1):125–133, Jan 2013.
- [GJLY12] O. Gerstel, M. Jinno, A. Lord, and S. J B Yoo. Elastic optical networking: a new dawn for the optical layer? *Communications Magazine, IEEE*, 50(2):s12–s20, 2012.
- [WM13] R. Wang and B. Mukherjee. Provisioning in elastic optical networks with non-disruptive defragmentation. *Lightwave Technology, Journal of*, 31(15):2491–2500, 2013.
- [WM14] R. Wang and B. Mukherjee. Spectrum management in heterogeneous bandwidth optical networks. *Optical Switching and Networking*, 11, Part A(0):83 – 91, 2014.