Combinatorial Properties and Defragmentation Algorithms in WSW1 Switching Fabrics

Remigiusz Rajewski, Wojciech Kabaciński, and Atyaf Al-Tameemi

Abstract—A spectrum defragmentation problem in elastic optical networks was considered under the assumption that all connections can be realized in switching nodes. But this assumption is true only when the switching fabric has appropriate combinatorial properties. In this paper, we consider a defragmentation problem in one architecture of wavelength-space-wavelength switching fabrics. First, we discuss the requirements for this switching fabric, below which defragmentation does not always end with success. Then, we propose defragmentation algorithms and evaluate them by simulation. The results show that proposed algorithms can increase the number of connections realized in the switching fabric and reduce the loss probability.

Keywords—elastic optical network, switching network, rearrenging algorithm, defragmentation algorithm

I. INTRODUCTION

IN elastic optical networks (EON), the optical spectrum available in an optical fiber is divided into small slots called frequency slot units (FSUs) and an optical channel may occupy several such units [1], [2]. When an optical path uses an optical channel composed of m FSUs, the connection provided for this optical path is called an m-slot connection. One of the important device in the EON, where the bandwidth assign to an optical path depends not only on the required transmission bit rate, but also on a distance to be covered or modulation format used, is a switching node. Different structures of switching nodes in EONs were considered in [3]–[6], and short survey of switching fabrics in these nodes can be found in [7]. One of such structures are wavelength-space-wavelength (W-S-W) switching fabrics proposed in [8] and [5], and named WSW1 and WSW2.

One of the important constraints imposed on the optical path in EONs is that it must occupy adjacent FSUs (adjacency constraint). When the EON serves dynamic traffic, i.e., connections may arrive at the network and departure after some time, frequent changes in connecting paths lead to bandwidth fragmentation. As a result, in the network links we may have some FSUs available, but they not necessary must be adjacent, and new requests cannot be realized because of the insufficient number of free adjacent FSUs. The same problem may occur in interstage links of multistage switching fabrics

that serve m-slot connections. The fragmented spectrum may be defragmented using some defragmentation algorithms.

In this paper, we consider the defragmentation problem in one type of the three-stage W-S-W switching fabric called WSW1. Some combinatorial properties of this kind of switching fabric were already considered in several papers. The strict-sense nonblocking conditions have been proved in [8], while the wide-sense nonblocking conditions for different routing algorithm have been provided in [9]–[11]. Rearrangeability of WSW1 switching fabrics were considered in [12]–[15].

Defragmentation for EON is very often investigated at the network (links) level [16]–[18]. It means that frequency slot units in input and output fibers are defragmented according to the network state. Thus, for such defragmentation, it is not important what is the internal structure of a switching node. Nevertheless, defragmentation can also be done inside such a node. Such defragmentation is then done only in interstage links of a particular structure of used switching fabric. After such defragmentation, the state at the input and output fibers does not change.

The rest of this study is organized as follows. In Section II, we present the structure of the WSW1 switching fabric and notation used throughout the paper. In Section III, we provide the lower bound for rearrangeability of the considered switching fabric. This lower bound gives the number of FSUs in interstage links which are necessary, below which the switching fabric will be blocking. In Section IV, we describe defragmentation in general. In turn, in Section V, we propose defragmentation algorithms that can be used in WSW1 switching fabric. In Section VI, we present results of defragmentation obtained by simulation. The paper ends with conclusions.

II. THE SWITCHING FABRIC

The WSW1 switching fabric is presented in Fig. 1. It consists of three stages. Stages first and third contain r BV-WS (bandwidth-variable wavelength spectrum converting) switches which convert the FSUs occupied by a connection in the input fiber to another set of FSUs used in the output fiber. The second stage consists of one BV-SS (bandwidth-variable wavelength-selective space) switch, which moves connections between input and output fibers without changes in occupied FSUs. Examples of how switches BV-WS and BV-SS can be constructed are presented, for instance, in [5], [8], [11], however, this construction does not influent the results presented in this study. Each BV-WS has one input fiber and one output fiber with the spectrum divided into n or k FSUs. BV-WSs in

R. Rajewski and W. Kabaciński are with Poznan University of Technology, Poznań, Poland (e-mail: {wojciech.kabacinski, remigiusz.rajewski}@put.poznan.pl).
A. Al-Tameemi is the PhD student at Poznan University of Technology, Poznań, Poland (e-mail: atyaf.al-tameemie@doctorate.put.poznan.pl).
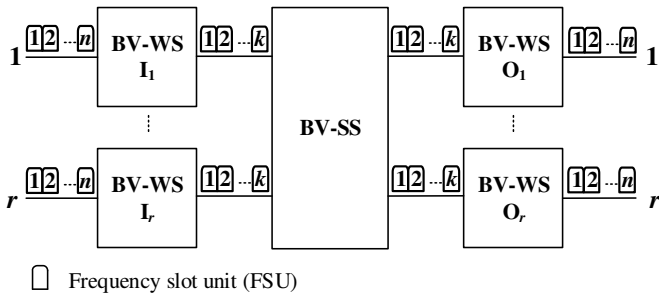
Frequency slot unit (FSU)

Fig. 1. The WSW1(r, n, k) switching fabric.

the first stage have n FSUs in the input fibers and k FSUs in the output fibers, while the opposite arrangement is used in the third stage. The BV-SS switch has r input fibers and r output fibers, i.e., its capacity is $r \times r$, with k FSUs in each fiber. This switching fabric is often denoted as WSW1(r, n, k), since parameters r, n, and k unambiguously define its structure.

The first-stage switch moves a connection from the set of FSUs used in the input fiber to the set of FSUs which can be used in the interstage link to the BV-SS switch. In the second-stage switch, the connection is moved from the interstage link form the first-stage switch to the interstage link connected to the third-stage switch, however, the connection must occupy the set of the FSUs indexed with the same numbers, since the second-stage switch has no conversion capability. Finally, in the third-stage switch, the connection is moved from the set of FSUs used in the interstage links to the set of FSUs assigned in the output fiber.

We will use the following notation in:

- I_i – switch i in the first stage, $1 \leqslant i \leqslant r$;
- O_j – switch j in the third stage, $1 \leqslant j \leqslant r$;
- $(I_i,\ O_j,\ m)$ – m-slot connection from I_i to O_j;
- $(I_i[x],\ O_j[y],\ m)$ – m-slot connection from I_i to O_j which occupies FSUs from x to $x + m - 1$ in the input fiber and FSUs from y to $y + m - 1$ in the output fiber;
- I_i–O_j – a connection between I_i and O_j when the number of used FSUs is not important.

FSUs in input and output fibers are numbered from 1 to n and in the interstage links – from 1 to k. The number of FSUs m used by on connection is limited to some maximum value $m_{\max}$, $1 \leqslant m \leqslant m_{\max}$.

When a new connection request $(I_i,\ O_j,\ m)$ arrives at the switching fabric, a path searching (or routing) algorithm has to find a set of m adjacent and free FSUs in interstage links, and these must be the FSUs with the same index numbers in both links (i.e., leading from I_i and to O_j), since there is no conversion in BV-SS switch. When there are not such FSUs, $(I_i,\ O_j,\ m)$ is blocked. When in the interstage links the number of free FSUs is less than m, we cannot do anything to unblock the new request. However, when there are at least m slots free but they are not adjacent, we can try to move existing connections to other sets of FSUs to unblock the new request. This can be done by rearrangements or spectrum defragmentation. First, we will consider, how many FSUs we need at least in the interstage link (i.e., the minimum value of

k) so that such rearrangements may end with success. Then we will move to defragmentation algorithms.

Let us assume, that we have a set of compatible connections, i.e., connections occupy different sets of FSUs in input and output fibers. In other words, there is no such FSU, which is assigned to two different connections. Assume also, that connections use all FSUs in input and output fibers. Such a set of connections is denoted by $\mathbb{C}$ and will be represented by the following matrix [12]:

$$\mathbf{H}^{r \times r} = [h_{ij}], \tag{1}$$

where

$$h_{ij} = \sum_{\mathbb{X}} m, \qquad \mathbb{X} \subseteq \mathbb{C}, \tag{2}$$

and $\mathbb{X}$ denotes a set of all connections $(I_i,\ O_j,\ m)$ in $\mathbb{C}$. Therefore, h_{ij} is equal to the number of FSUs used by all connections from I_i to O_j. This matrix has the following property:

$$\sum_{i=1}^{r} h_{ij} = \sum_{j=1}^{r} h_{ij} = n. \tag{3}$$

III. REARRANGEABILITY

When we have two connections, $(I_1,\ O_1,\ m_1)$ and $(I_2,\ O_2,\ m_2)$, they are not in conflict since they have to use different interstage links. $(I_1,\ O_1,\ m_1)$ will use FSUs in the interstage links from I_1 and to O_1, while $(I_2,\ O_2,\ m_2)$ – from I_2 and to O_2. So they may use FSUs from 1 to m_1 and m_2. When, for instance $m_2 > m_1$, FSUs numbered from $m_1 + 1$ to m_2 in the link from I_1 can be used only by another connection I_1–O_j, where $j \neq 2$. When the other connections are only connections I_1–O_2, they will have to use FSUs numbered from $m_2 + 1$ and FSUs numbered from $m_1 + 1$ to m_2 will be unused. This example shows, that adjacency constraint may lead to the situation, that some of FSUs in interstage links will be unavailable for some other connections. Therefore, generally we need $k > n$. However, when $r = 2$, we can observe the following property [12]. In matrix $\mathbf{H}^{2 \times 2}$, when we take into account property 3, we have $h_{11} + h_{12} = n$, i.e., $h_{12} = n - h_{11}$, and $h_{11} + h_{21} = n$, i.e., $h_{21} = n - h_{11}$. So we get $h_{12} = h_{21}$ and $h_{11} = h_{22}$. Thus, for any $\mathbb{C}$, the matrix (1) is as follows:

$$\mathbf{H}^{2 \times 2} = \begin{bmatrix} h_{11} & n - h_{11} \\ n - h_{11} & h_{11} \end{bmatrix}. \tag{4}$$

We need h_{11} FSUs in interstage links to set up connections represented by h_{11} and h_{22} since they can be set up through the same numbered FSUs. Similarly, we need $n - h_{11}$ FSUs for connections represented by h_{12} and h_{21}. Eventually, we need $k = h_{11} + n - h_{11} = n$ FSUs in interstage links [12].

However, when $r > 2$ the problem is more complicated. Let us consider the WSW1$(3, n, k)$. When, for instance, $h_{11} = h_{22}$, connections represented by these elements may use FSUs with the same index numbers, and such indexed FSUs can be also used only by connections represented by h_{33}. When $h_{33} < h_{11}$, some FSUs in the interstage link from I_3 will remain unused and other connections from this switch will require additional FSUs. The question is how many such
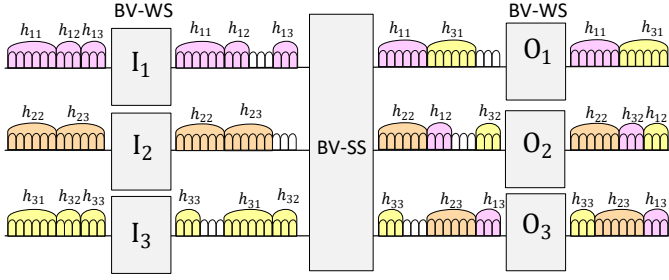
Fig. 2. The WSW1$(3, 12, 15)$ switching fabric with connections the following set of connections: $(I_1,\ O_1,\ 6)$, $(I_1,\ O_2,\ 3)$, $(I_1,\ O_3,\ 3)$, $(I_2,\ O_2,\ 6)$, $(I_2,\ O_3,\ 6)$, $(I_3,\ O_1,\ 6)$, $(I_3,\ O_2,\ 3)$, and $(I_3,\ O_3,\ 3)$; assignment 1.
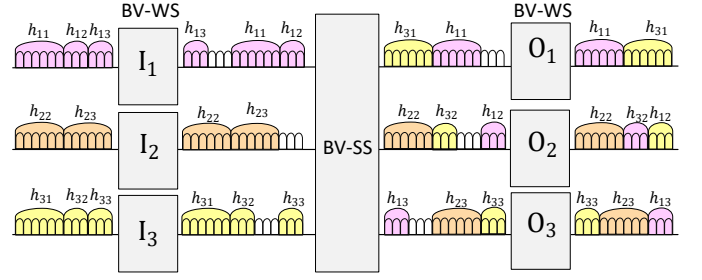
Fig. 3. The WSW1$(3, 12, 15)$ switching fabric with connections the following set of connections: $(I_1,\ O_1,\ 6)$, $(I_1,\ O_2,\ 3)$, $(I_1,\ O_3,\ 3)$, $(I_2,\ O_2,\ 6)$, $(I_2,\ O_3,\ 6)$, $(I_3,\ O_1,\ 6)$, $(I_3,\ O_2,\ 3)$, and $(I_3,\ O_3,\ 3)$; assignment 2.

additional FSUs are needed. As an example, let's look at the WSW1$(3, 12, k)$ with the set $\mathbb{C}$ represented by the following matrix:

$$\mathbf{H}^{3\times 3} = \begin{bmatrix} 6 & 3 & 3 \\ 0 & 6 & 6 \\ 6 & 3 & 3 \end{bmatrix}. \tag{5}$$

When each h_{ij} represents one connection, we have four 6-slot and four 3-slot connections. In the same numbered FSUs (say 1–6) we can set up connections $(I_1,\ O_1,\ 6)$ and $(I_2,\ O_2,\ 6)$. These FSUs can be also used by connection $(I_3,\ O_3,\ 3)$ and remaining 3 FSUs in the link from I_3 to the center-stage switch cannot be used by any other connection. Therefore, we have to add at least 3 additional FSUs in the interstage link to realize connections $(I_3,\ O_1,\ 6)$ and $(I_3,\ O_2,\ 3)$. Similarly, connections $(I_2,\ O_3,\ 6)$ and $(I_3,\ O_1,\ 6)$ can use the same numbered FSUs (say 7–12, since they must be different than that used by connections $(I_1,\ O_1,\ 6)$ and $(I_2,\ O_2,\ 6)$). These FSUs are also available for connection $(I_1,\ O_2,\ 3)$ and again, 3 FSUs will remain unused in the link from I_1, so additional 3 FSUs are needed in this link for connection $(I_1,\ O_3,\ 3)$. Connections $(I_1,\ O_3,\ 3)$ and $(I_3,\ O_2,\ 3)$ can use the same numbered FSUs in interstage links, numbered from 13–15. The total number of required FSUs in interstage links is $k = 15$. This assignment is shown in Fig. 2.

The other option for realizing this set of connections is to use FSUs 1–6 for connections $(I_2,\ O_2,\ 6)$, $(I_3,\ O_1,\ 6)$ and $(I_1,\ O_3,\ 3)$. This time 3 FSUs in the link from I_1 will remain unused. In the next six FSUs (i.e., 7–12) we can set up connections $(I_2,\ O_3,\ 6)$, $(I_1,\ O_1,\ 6)$ and $(I_3,\ O_2,\ 3)$. Again 3 FSUs will remain unused, but this time in the link from I_3. As the result, connections $(I_1,\ O_2,\ 3)$ and $(I_3,\ O_3,\ 3)$ require three additional FSUs numbered from 13 to 15, and the total number of used FSUs is also $k = 15$ (see Fig. 3)

It is also possible to change the order of assigning FSUs to connections, i.e., assigning FSUs 1-6 to connection $(I_2,\ O_3,\ 6)$, but this will not reduce the required number of FSUs in interstage links. In the considered example, n was a multiple of 4, but the result my be generalized to any value of n. There are some differences in the assignment when n is odd or even, but it was formally proved in [12] that the minimum number of required FSUs (for $n \geqslant 4$) is given by the following formula:

$$k \geqslant n + \left\lfloor \frac{n}{4} \right\rfloor. \tag{6}$$

It also gives the minimum number of FSUs which we should have if we expect that defragmentation lead to success, i.e., when the defragmentation process is finished, any new connection can be realized in the switching fabric.

IV. DEFRAGMENTATION

Because the defragmentation in a switching node can be done only in the interstage links, the only place in the WSW1 structure where defragmentation is realized are input and output links of the BV-SS switch. Each defragmentation method has to answer following questions: *When to defragment?*, *For what to defragment?*, and *How to defragment?*.

A. When to defragment?

Such defragmentation can be done in different moments:

1) The first moment when the defragmentation can be done is the moment when m-slot connection is disconnected. It means that m free FSUs are appearing, and they can be used together with other free FSUs to establish future connections.

2) The second moment when the defragmentation can be done is the moment when a new m-slot connection appears, and in the switching node, there are enough free FSUs (at least m). However, they are not enough adjacent FSUs to establish such a new connection.

B. For what to defragment?

Defragmentation allows to establish a new connection when there are enough free FSUs in some internal link in a switching node; however, they are not adjacent ones. Several defragmentation methods manage free FSUs in different ways [19], [20]. Thus, choosing the right defragmentation method allows sorting free FSUs in the optical spectrum in a required way to establish a new connection.

C. How to defragment?

There are several defragmentation methods. All of them are used to manage the optical spectrum in links between EON nodes (defragmentation at the network level). In this paper, we use only two of them to defragment optical spectrum inside the WSW1 node architecture: *push-and-pull* (PP) [19] and *hop-tuning* (HT) [20]. We also merged PP and HT

methods into a hybrid method (HTPP), and we proposed three new defragmentation methods: modified *push-and-pull* method (PP2), modified *hop-tuning* method (HT2), and altered hybrid method (HTPP2).

V. THE DEFRAGMENTATION METHODS

In this section, we described a few defragmentation methods which were investigated and implemented in our research.

A. The Push-and-Pull Method (PP)

In the PP defragmentation method, already established connections are pushed in the optical spectrum only into one direction (in the right or the left). Any existing connection can be pushed in a range of free adjacent FSUs until it did not become an adjacent connection to any already established connection. In case, when there are no free adjacent FSUs for a pushing connection, this connection can not be pushed farther.

A simple example showing how the PP method operates is presented in Fig. 4. There are following connections in the BV-SS switch before defragmentation (see Fig. 4(a)): $(I_1[2]; O_3[2]; 1)$ (green color), $(I_1[3]; O_1[3]; 2)$ (red color), $(I_1[9]; O_2[9]; 3)$ (blue color), $(I_2[4]; O_3[4]; 2)$ (orange color), $(I_3[3]; O_2[3]; 4)$ (light brown color), and $(I_3[10]; O_3[10]; 3)$ (pink color). During the defragmentation process some connections were moved and we have the following state (see Fig. 4(b)): $(I_1[1]; O_3[1]; 1)$ (green color), $(I_1[2]; O_1[2]; 2)$ (red color), $(I_1[7]; O_2[7]; 3)$ (blue color), $(I_2[2]; O_3[2]; 2)$ (orange color), $(I_3[3]; O_2[3]; 4)$ (light brow color), and $(I_3[7]; O_3[7]; 3)$ (pink color).

B. The Hop-Tuning Method (HT)

In the HT defragmentation method, already established connections are moved in the optical spectrum into free FSUs, not necessarily adjacent to the moving connection. In case, when there is no enough free FSUs for the moving connection, this connection cannot be moved. In contrast to the PP method, the HT method allows moving even a few connections simultaneously. It is a huge advantage because the time needed for the defragmentation process is shorter, and it is very often below 1μs. Of course, in the HT method, there is no need to use additional transmitters at the input side.

A simple example showing how the HT method works is presented in Fig. 5. Before defragmentation, we have following connections established in the BV-SS switch (see Fig. 5(a)): $(I_1[2]; O_1[2]; 2)$ (green color), $(I_1[8]; O_2[8]; 2)$ (blue color), $(I_1[10]; O_2[10]; 3)$ (red color), $(I_2[3]; O_2[3]; 2)$ (gray color), $(I_2[10]; O_3[10]; 3)$ (orange color), $(I_3[9]; O_3[9]; 1)$ (pink color), and $(I_3[10]; O_1[10]; 3)$ (yellow color). After defragmentation in the BV-SS switch some connections were moved and there is the following state (see Fig. 5(b)): $(I_1[2]; O_1[2]; 2)$ (green color), $(I_1[8]; O_2[8]; 2)$ (blue color), $(I_1[5]; O_2[5]; 3)$ (red color), $(I_2[3]; O_2[3]; 2)$ (gray color), $(I_2[5]; O_3[5]; 3)$ (orange color), $(I_3[1]; O_3[1]; 1)$ (pink color), and $(I_3[10]; O_1[10]; 3)$ (yellow color).
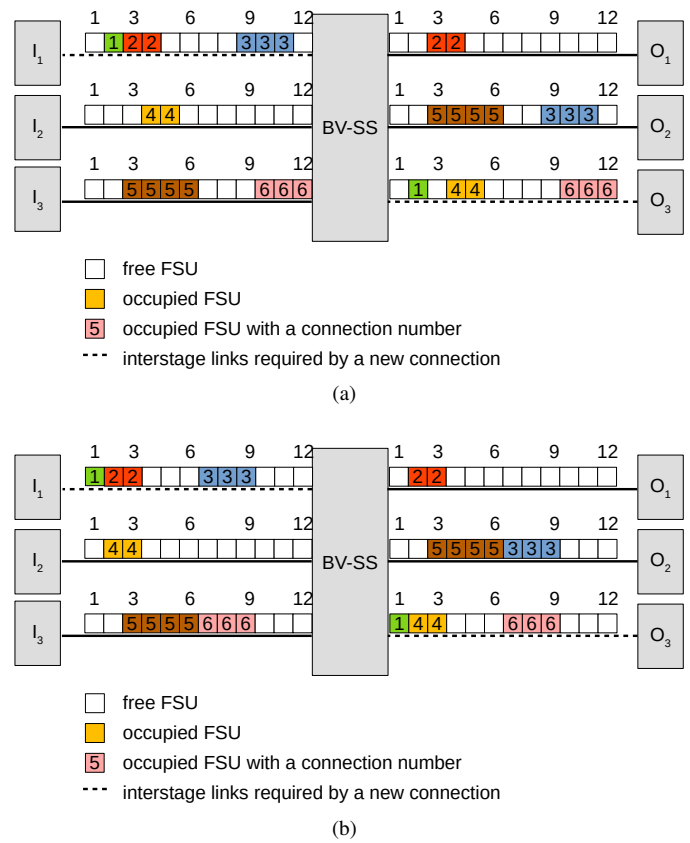

Fig. 4. The *push-and-pull* (PP) defragmentation method: (a) the state before defragmentation, (b) the state after defragmentation.
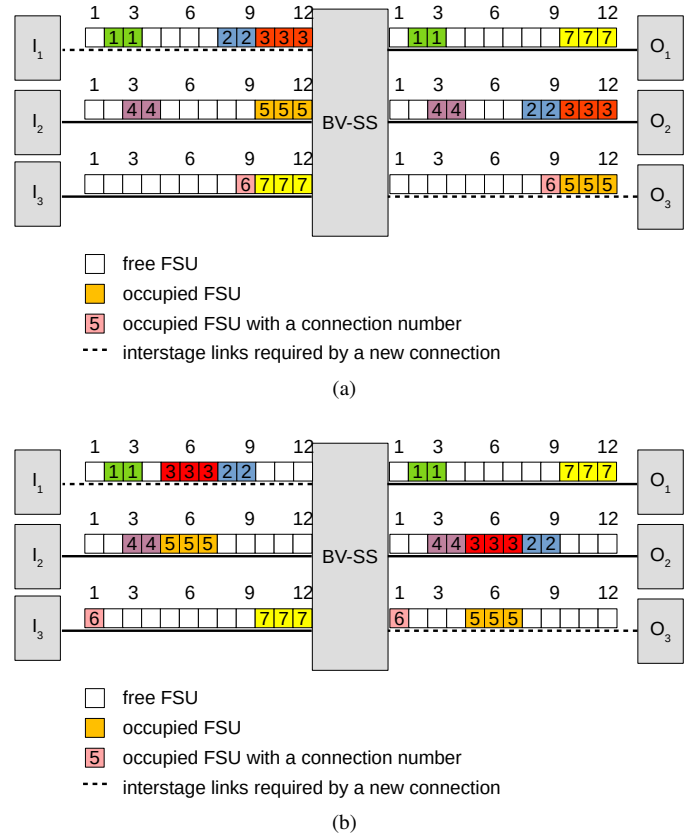

Fig. 5. The *hop-tuning* (HT) defragmentation method: (a) the state before defragmetnation, (b) the state after defragmentation.

C. The Hybrid Hop-Tuning and Push-and-Pull Method (HTPP)

In the hybrid HTPP defragmentation method the HT method is used first followed by the PP defragmentation method. In the PP method, some connections cannot be pushed farther because they depend on the situation on different interstage links. The HTPP allows decreasing the weakens of the PP method, i.e., in the PP method, there are some "holes" of the free optical spectrum which cannot be filled in some cases, and the HT method allows to fill these "holes" as well.

D. The Modified Push-and-Pull Method (PP2)

The PP2 defragmentation method works like the PP method, except:

- pushing of already established connection can be done in both left and right directions;
- the defragmentation is interrupted when a new connection can be established.

This means that, in some links, not always all established connections are pushed. If after pushing only one already established connection a new connection can established, this defragmentation method is finished and the new connection is set up. However, when pushing one already established connection is not sufficient, the second already set up connection is pushed, and so on, until no more connections can be pushed. If, until that moment, it was no possible to establish a new connection, it means that defragmentation finished with no success even when there were enough free FSUs. In contrast to the PP method, the advantage of the PP2 defragmentation method in most cases is a new connection is waiting a shorter time till it will be established. In other words, in more cases the PP2 method is quicker than the PP method because in most cases less already established connections are pushed, in the worst-case, the number of pushed connections is the same as in the PP method. The disadvantage of the PP2 defragmentation method is that in the future, the optical spectrum at a given link can have more spectrum "holes". This, in turn, can cause future problems with setting up future connections (see results which we achieved in Section VI).

A simple example showing how the PP2 method is presented in Fig. 6 where dashed FSUs denote a new 5-slot connection (which cannot be established using the PP defragmentation method). Before defragmentation, the situation in the BV-SS switch is the same as for the PP method (see Fig. 4(a)). In turn, after the defragmentation in the BV-SS switch already established connection were pushed and the state is as follows (see Fig. 6): $(I_1[1]; O_3[1]; 1)$ (green color), $(I_1[2]; O_1[2]; 2)$ (red color), $(I_1[9]; O_2[9]; 3)$ (blue color), $(I_2[2]; O_3[2]; 2)$ (orange color), $(I_3[3]; O_2[3]; 4)$ (light brown color), and $(I_3[10]; O_3[10]; 3)$ (pink color).

E. The Modified Hop-Tuning Method (HT2)

The HT2 defragmentation method works the same as the HT method; however, it is interrupted when a new connection can be established, i.e., not all already established connections in some link are moved. If moving one of the already established
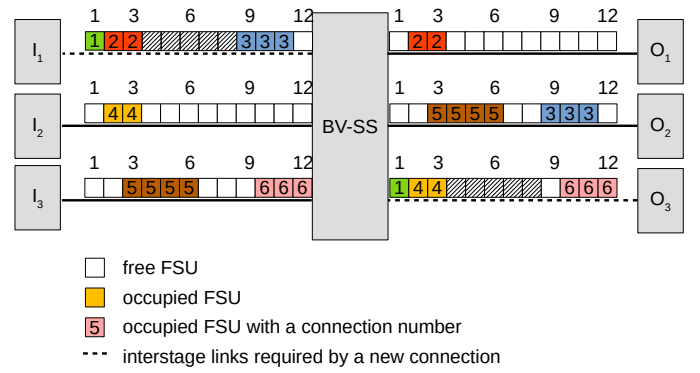

Fig. 6. The modified *push-and-pull* (PP2) defragmentation method: state after defragmentation.

connections allows setting up a new connection, this method is finished, and the new connection is set up. If moving one of the already established connections is not enough, the second already set up connection is moving, and so on until no more already established connections can be moved. If, until that moment, it was not possible to establish a new connection, it means that defragmentation finished with no success even when there were enough free FSUs in some link. In contrast to the HT method, the advantage of the HT2 defragmentation method is a new connection is waiting a shorter time till it will be established. In other words, in more cases, the HT2 method is quicker than the HT method because, in most cases, less already established connections are moved, and in the worst-case, the same number of such connections as in the HT method. The disadvantage of the HT2 method is that in the future, the optical spectrum at a given link can have more spectrum "holes". This, in turn, can cause future problems with setting up future connections (see results which we achieved in Section VI).

F. The Modified Hybrid Hop-Tuning and Push-and-Pull Method (HTPP2)

The HTPP2 defragmentation method works similar to the HTPP method. However, this time, defragmentation is interrupted when a new connection can be established, i.e., not all already established connections in some links are moved.

VI. SIMULATION EXPERIMENTS

During simulation experiments, we compared a few defragmentation methods: PP, HT, HTPP, PP2, HT2, and HTPP2. These methods were implemented in the EON node of the WSW1 structure (see Fig. 1). These methods were implemented and investigated in a specially prepared simulator in the OMNEST simulation environment [21]. We used a few random generators available in OMNEST:

- to generate the size of a new connection (the number of FSUs used by each connection);
- to draw a time when each connection arrivals;
- to draw the output to which each connection will be directed.

A final result for each method is the average value from 10 simulation runs, each of 1000000 events (the externally
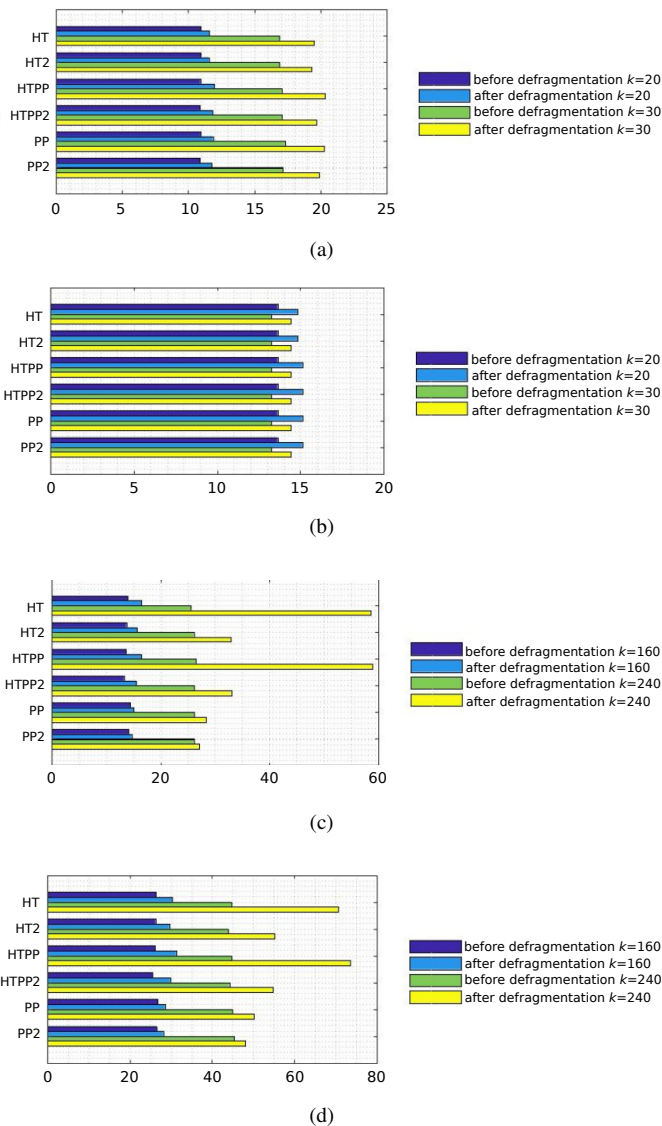
Fig. 7. The maximal average number of free adjacent FSUs in interstage links for: (a) $n = 20$, $m_{max} = 10$, (b) $n = 20$, $m_{max} = 20$, (c) $n = 160$, $m_{max} = 10$, (d) $n = 160$, $m_{max} = 20$.

blocked connections were not counted). Results for different methods were compared to each other, and are shown in charts of Fig. 7 and Fig. 8.

In Fig. 7, the average number of free adjacent FSUs in interstage links is shown for cases before and after defragmentation, for $n = 20$ and $n = 160$, and for a given maximum size of connections $m_{max} = 10$ and $m_{max} = 20$, respectively. We also assumed that the number of FSUs for interstage links was $k = n$ and $k = 1.5n$. For all simulated defragmentation methods, the maximal number of free FSUs in interstage links was the same. As a result, independently of the used defragmentation method, it is always possible to increase the number of free adjacent FSUs that increase the possibility of establishing a new connection.

In Fig. 8, the number of defragmentations finished with success is shown. The success means that a new connection was established after defragmentation (it was not rejected due

to the internal blocking). The HT2 defragmentation method allows for establishing more new connections than the HT defragmentation method. It means that the HT2 method is decreasing the internal blocking. The PP2 defragmentation method is, in turn, worse than the PP method because in a current moment, chances to establish a new connection grow, however, chances to establish future connections drop, i.e., the optical spectrum has more "holes". Due to that fact, new HTPP and HTPP2 methods were proposed. Both HTPP and HTPP2 methods allow for better optical spectrum usage than PP and HT defragmentation methods. As a result, new methods (HTPP and HTPP2) and the HT2 method allow to improve usage of the optical spectrum in interstage links in WSW1 node in contrast to other investigated methods. In turn, it allows us to increase the number of served connections and to decrease the internal blocking.

VII. Conclusion

In this study, a few defragmentation methods were used to manage an optical spectrum in interstage links in the WSW1 node of EON. Investigation was done not in the network level but only at the optical node level. We prepared in the OMNEST simulation environment a unique simulation program which allowed us to compare all investigated defragmentation methods. As a result, we showed which modified defragmentation methods are better than theirs primary forms and which methods allow us to decrease the internal blocking (more new connection can be served/established) in the WSW1 elastic optical node. We also discuss the minimum number of FSUs in interstage links and explained that when $k < 1.25n$ there is not any defragmentation method which will always end with success.

References

[1] M. Masahiko, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S.Matsuoka, "Spectrum-Efficient and Scalable Elastic Optical Path Network: Architecture, Benefits, and Enabling Technologies", *IEEE Communications Magazine* 47 (11), 66–73 (2009), DOI: 10.1109/MCOM.2009.5307468.

[2] V. López and L. Velasco, "Elastic Optical Networks. Architectures, Technologies, and Control", Springer International Publishing, Switzerland, (2016).

[3] Y. Chen, J. Li, P. Zhu, L. Niu, Y. Xu, X. Xie, Y. He, and Z. Chen, "Demonstration of Petabit scalable optical switching with subband-accessibility for elastic optical networks", in *OptoElectronics and Communication Conference and Australian Conference on Optical Fibre Technology (OECC/ACOFT)*, Melbourne, Australia, 6–10 July (2014).

[4] G. Danilewicz, W. Kabaciński, and R. Rajewski, "Strict-sense nonblocking space-wavelength-space switching fabrics for elastic optical network nodes", *IEEE/OSA Journal of Optical Communications and Networking* 8 (10), 745–756 (2016), DOI: 10.1364/JOCN.8.000745.

[5] W. Kabaciński, M. Michalski, and R. Rajewski, "Strict-sense nonblocking W-S-W node Architectures for elastic optical networks", *Journal of Lightwave Technology* 34 (13), 3155–3162 (2016), DOI: 10.1109/JLT.2016.2560624.

[6] P. Zhang, J. Li, B. Guo, Y. He, Z. Chen and H. Wu, "Comparison of node architectures for elastic optical networks with waveband conversion", *China Communications*, 10 (8), 77–87 (2013), DOI: 10.1109/CC.2013.6633747.

[7] B.C. Chatterjee, N. Sarma, and E. Oki, "Routing and spectrum allocation in elastic optical networks: A tutorial", *IEEE Communications Surveys and Tutorials* 17 (3), 1776–1800 (2015), DOI: 10.1109/COMST.2015.2431731.
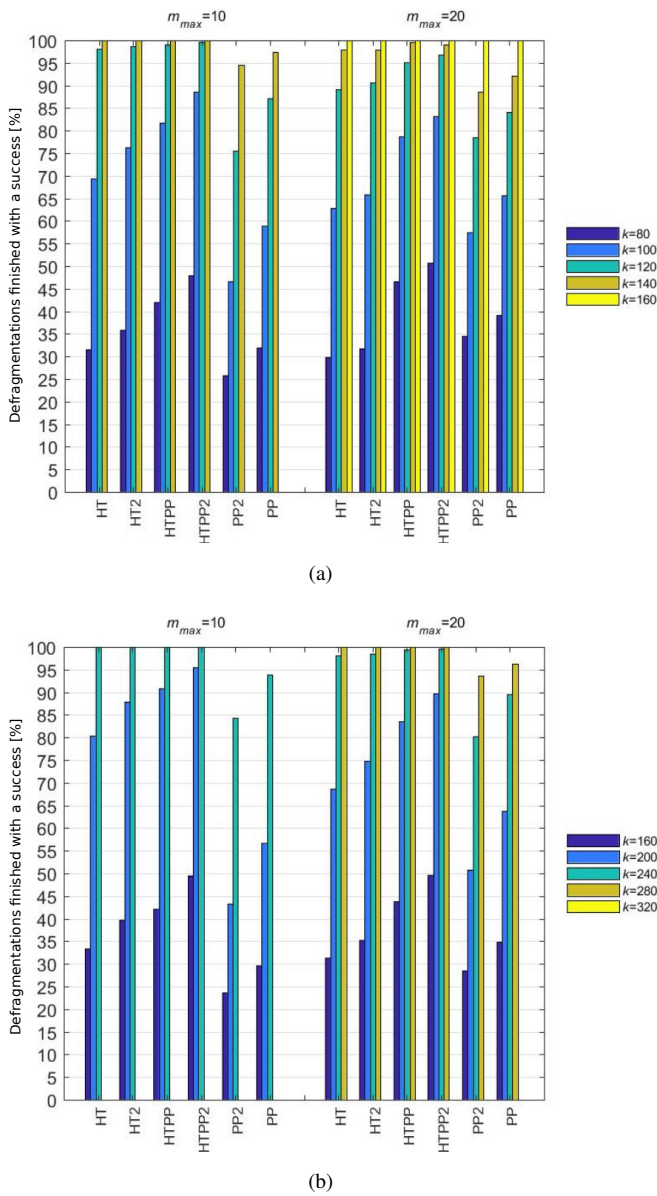
(a)

(b)

Fig. 8. The number of defragmentations finished with a success (a new connection was established) for $m_{max} = 10$ and $m_{max} = 20$ for: (a) $n = 80$, (b) $n = 160$.

[8] W. Kabaciński, M. Michalski, and M. Abdulsahib, "The strict-sense nonblocking elastic optical switch", in *IEEE 15th International Conference on High Performance Switching and Routing (HSPR)*, Budapest, Hungary (2015), DOI: 10.1109/HPSR.2015.7483108.

[9] M. Abdulsahib and W. Kabaciński, "Wide-Sense Nonblocking Converting-Space-Converting Switching Node Architecture under Xs-VarSLOT Algorithm", in *IEEE 20th International Conference on High Performance Switching and Routing (HPSR)*, Xi'An, China, 26-29 May (2019), DOI: 10.1109/HPSR.2019.8808115.

[10] W. Kabaciński, M. Abdulsahib, and M. Michalski, "Wide-sense nonblocking W-S-W node architectures for elastic optical networks", *IEICE Transactions Communications* E102–B (5), 978–991 (2019), DOI: 10.1587/transcom.2018EUP0004.

[11] M. Abdulsahib, M. Michalski, and W. Kabaciński, "Optimization of wide-sense nonblocking elastic optical switches", *Optical and Switching Networking* 33, 85–94, (2019), DOI: 10.1016/j.osn.2018.01.003.

[12] W. Kabaciński, A. Al-Tameemi, and R. Rajewski, "Necessary and sufficient conditions for the rearrangeability of WSW1 switching fabrics", *IEEE Access* 7, 18622–18633 (2019), DOI: 10.1109/ACCESS.2019.2896283.

[13] W. Kabaciński, R. Rajewski, and A. Al-Tameemi, "Simultaneous connections routing in W-S-W elastic optical switches with limited number of connection rates", in *21th International Conference on Optical Networks Design and Modeling (ONDM)*, Budapest, Hungary (2017), DOI: 10.23919/ONDM.2017.7958549.

[14] W. Kabaciński and A. Al-Tameemi, "Control algorithms for simultaneous connections routing in flexible optical switching networks", in *International Conference on Photonics in Switching and Computing (PSC)*, Limassol, Cyprus (2018), DOI: 10.1109/PS.2018.8751273.

[15] B.-C. Lin, "Rearrangeable W-S-W elastic optical networks generated by graph approaches", *IEEE/OSA Journal of Optical Communications Networking* 10 (8), 675–685 (2018), DOI: 10.1364/JOCN.10.000675.

[16] M. Zhang, W. Shi, L. Gong, W. Lu, Z. Zhu, "Bandwidth Defragmentation in Dynamic Elastic Optical Networks with Minimum Traffic Disruptions", in *IEEE International Conference on Communications (ICC)*, Budapest, Hungary, 3894–3898 (2013), DOI: 10.1109/ICC.2013.6655165.

[17] C. You, M. Zhang, Z. Zhu, "Reduce Spectrum Defragmentation Latency in EONs with Effective Parallelization of Connection Reconfigurations", in *Optical Fiber Communications Conference and Exhibition (OFC)*, San Francisco, CA, USA, 1–3 (2014), DOI: 10.1364/OFC.2014.W4A.6.

[18] S. Ba, B. C. Chatterjee, E. Oki, "Defragmentation Scheme Based on Exchanging Primary and Backup Paths in 1+1 Path Protected Elastic Optical Networks", *IEEE/ACM Transactions on Networking* 25 (3), 1717–1731 (2017), DOI: 10.1109/TNET.2017.2650212.

[19] F. Cugini, et al, "Push-Pull Defragmentation without Traffic Disruption in Flexible Grid Optical Networks", *Journal of Lightwave Technology* 31 (1), 125–133 (2013), DOI: 10.1109/JLT.2012.2225600.

[20] M. Zhang, Y. Yin, R. Proietti, Z. Zhu, S. J. B. Yoo, "Spectrum defragmentation algorithms for elastic optical networks using hitless spectrum retuning techniques", in *Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC)*, Anaheim, CA, USA 1–3 , 17–21 March (2013).

[21] The OMNEST official website, last access: March 20, 2019, https://omnest.com.