International Journal of Innovation and Applied Studies
ISSN 2028-9324 Vol. 41 No. 4 Feb. 2024, pp. 1221-1232

http://www.ijias.issr-journals.org/

Dynamic (Re) allocation For Multicast Connexion in Elastic Optical Networks

Adepo Joel Christian[1], N'Takpe N'Guessan Christian Placide[1], Diédié Gokou Hervé Fabrice[2], and Babri Michel[3]

[1]Digital Research and Development Unit, Virtual University of Ivory Coast, Abidjan, Côte d'Ivoire

[2]Laboratory of Mathematics and Computer Science, Peleforo Gon Coulibaly University, Korhogo, Côte d'Ivoire

[3]National Institut Félix Houphouët Boigny, Yamoussoukro, Côte d'Ivoire

ABSTRACT: Elastic Optical Networks (EON) have emerged as a promising solution to meet the demands for flexible and high-capacity communication. However, due to spectral contiguity and continuity constraints, the efficient allocation of resources, such as bandwidth, in EONs remains a challenging task, especially for multicast connections. Indeed, in dynamic traffic scenarios, frequent establishment and teardown of requests lead to the formation of isolated spectrum fragments that cannot be allocated to other requests. Several reallocation approaches exist to aggregate these fragments by reconfiguring already established connections in the network to accommodate new connection requests. All these approaches aim to minimize disruptions for users by minimizing the number of reconfigured connections in the reallocation process. We propose a new approach called Dynamic Bandwidth Reallocation for Multicast Connections (RDBM), which is specifically designed for multicast connections. The RDBM algorithm primarily aims to minimize disruptions during the process of reallocating bandwidth for multicast connections while reducing the blocking probability for dynamic multicast traffic.

KEYWORDS: Elastic optical networks, routing and spectrum allocation, multicast spectrum reallocation.

1 INTRODUCTION

1.1 CONTEXT

The ever-increasing and heterogeneous bandwidth demands on the internet are partly due to the proliferation of multicast services such as video conferencing and cloud services [1], [2]. Currently, the most suitable networks to cope with the intense and diverse bandwidth demands on the internet are elastic optical networks [3]. In these networks, the spectrum is divided into multiple units of bandwidth called Frequency Slot Units (FSUs) or slots. These slots are then aggregated and allocated to users based on their bandwidth requirements, enabling flexible bandwidth allocation with high data rates (Tb/s). However, in the absence of wavelength converters, slot allocation in EONs is subject to two major constraints: contiguity and continuity. The contiguity constraint means that the slots allocated to a connection must be adjacent in the spectrum (successive slots). The continuity constraint requires that, for a given connection, the same spectrum range (the same successive indices) should be reserved on each link along the connection path, meaning from the source to the destination. Due to continuity and contiguity constraints, spectrum fragmentation is one of the major challenges in EONs. This occurs when connections are dynamically established and torn down, leading to the formation of numerous isolated slots that cannot be allocated to other requests [4]. In this context, bandwidth defragmentation is a key challenge. The goal is to reorganize the slots to reduce spectrum fragmentation and free up contiguous slots for allocating new connections by reallocating already used frequency slots to existing established connections in the network. This process enhances the utilization of available resources and improves the capacity of the network to accommodate new demands.

The spectrum defragmentation problem is addressed by reallocating frequency slots. Spectrum reallocation can be resolved using either a proactive or reactive approach [5]. The proactive approach involves the use of algorithms that are to reorganize the spectrum. In the reactive approach, algorithms are invoked when a connection request is blocked. In this case, the algorithms attempt to reconfigure existing connections to free up the necessary resources to accommodate the new connection request. With increasingly dynamic traffic and the growth of multicast services, this study focuses on the reactive reallocation approach.

1.2 OBJECTIVES

The defragmentation process requires the reconfiguration of certain connections established in the network. This reconfiguration can disrupt the destination nodes of these connections. To mitigate the impact of disruptions on all users, spectrum defragmentation algorithms aim to minimize the number of reconfigured connections. The objective of this study was to propose a reactive defragmentation approach that reconfigures multicast connections while minimizing the number of users affected by the process.

1.3 ORGANIZATION

The current section of briefly introduces the reallocation problem, along with the problem statement, objectives, motivation, and contributions. Section 2 provides a review of related work on network reconfiguration. Section 3 provides a detailed description of the problem. Section 4 is the proposed reallocation algorithm. In Section 5, the proposed algorithm is implemented, and the results are analysed and discussed. Finally, Section 6 concludes this study.

2 RELATED WORK

2.1 MULTICAST ROUTING AND SPECTRUM ASSIGNMENT

The resolution of the Multicast Routing and Spectrum Allocation (MC-RSA) problem consists of two sub-problems. The first is the routing problem, which involves finding a tree-like path for a given connection request. The second problem is the spectrum allocation along the path to meet the bandwidth demand of the connection [6], [7].

2.2 SPECTRUM REALLOCATION

The blocking of a connection request can be justified for various reasons. This can be due to a lack of available resources on the determined paths or the absence of contiguous or continuous slots. When the slots are not contiguous, spectrum reallocation involves reconfiguring existing connections by assigning them to different frequency slots to free up contiguous slots to accommodate the new connection request [8], [9]. Several reallocation techniques exist and can be grouped into four main techniques, namely "re-optimization," "make-before-break," the "push-pull" technique, and the "hop tuning" technique [10], [11], [12], [13]. In [14], [15], the authors conducted a comparative study of these techniques. Their results indicate that the "hop tuning" technique yields the best blocking probability for connection requests. In this work, we propose a reallocation algorithm based on defragmentation hoptuning techniques that not only achieves a favorable blocking probability but also minimizes the number of user disruptions. In the case of unicast connections, this means reducing the number of existing traffic flows to be reconfigured in the defragmentation process, because each reconfigured connection disrupts the flow to a single destination node [16]. However, in the case of multicast connections, disrupting a single connection affects multiple destination nodes (i.e., users). When dealing with multicast connections, minimizing the number of users affected by the reallocation process involves finding a reallocation solution in which the number of reconfigured connections minimizes the number of destination nodes.

3 MODELLING THE PROBLEM

3.1 NETWORK MODEL

Let's consider a flexible optical network with an arbitrary physical topology defined by a graph *G = (V, E),* where V is the set of network nodes, and E is the set of network links. Each node is equipped with reconfigurable optical add-drop multiplexers (ROADMs) and variable bandwidth transponders (BVTs) to interface with the upper layers of the network. In addition, each link corresponds to a bidirectional pair of optical fibers. Furthermore, the spectrum of each link, denoted by $l \in L$, is divided into *F* frequency slots, where*f* (*f* = 1,..., |F|) designates the slot with index *f*. The state of the network is described by a set *M* of established optical trees, obtained by applying a suitable multicast routing and slot allocation (MRSA) algorithm to a set of connection requests [6].Each optical tree, denoted as $i \in M$, is defined by the tuple(s_i, D_i, n_i, f_i), where s_i is the source node, $D_i$the set of de destination nodes, $n_i$the number of assigned spectrum slots and$f_i$is the first slot of spectrum assigned to *i* which ranges from f_i up to $(f_i + n_i - 1)$. A set of candidate trees for each connection request is precalculated using the Shortest Path Tree (SPT) algorithm. To avoid numerous reconfiguration steps that could impact network performance, established connections are reconfigured in parallel. This parallel reconfiguration is performed through the hop-tuning technique. Hence, for a given set of reallocation sessions, only two steps are required, namely the simultaneous allocation of new slots and the simultaneous release of the old slots that were assigned to them.

3.2 PROBLEM FORMULATION

Given:

- An elastic optical network (EON); the initial state of the network S_0;
- A set of already established connections denoted by *M*;
- The set of optical trees of M denoted by *T*;
- A blocked connection demand *k* such that $k = (s_k, D_k, n_k)$; the path of *k* is denoted by T_k.

Goal:

- Reduce the blocking probability.

Specific objectives:

- Reconfigure existing connections to free up enough slots to meet any blocked demand k;
- Minimize the number of nodes of existing connections disrupted by the reallocation process.

Constraints:

- The initial routing of established connections is not modified;
- The slots reallocated to already establish connections in the reallocation process are free;
- The reallocation process occurs without the flow interruption;
- The number of destination nodes for each established multicast connection is randomly variable.

3.3 SPECIFICATION OF PROBLEM

Existing reallocation methods prioritize minimizing the disruption of a minimal number of traffic without considering the number of destination nodes disrupted by this process. In a unicast context, these two objectives align, implying that reducing the number of reconfigured connections inherently reduces the number of destination nodes affected by flow interruption. However, for multicast connections, reducing the number of reconfigured connections does not necessarily reduce the number of destination nodes impacted. This is because the number of destination nodes varies for all multicast connections; therefore, the number of destination nodes of multicast connections and the number of connections in the network are uncorrelated. On the one hand, reconfiguring a unicast connection affects a single user (single node), while on the other hand, reconfiguring a single multicast connection can impact several users. The challenge of this work, for a blocked connection request, is to find among there allocation solutions the one that minimizes the number of disrupted users.

3.4 ILLUSTRATION WITH A REALLOCATION PROBLEM INSTANCE

Fig 1 shows an instance of the spectrum reallocation problem. The considered network consists of 5 nodes and 7 links. Connections m_1, m_2, and m_3 have 3 destinations, 2 destinations, and 1 destination, respectively. The initial routing comprises trees T_1 = (1, {2, 3, 4}), T_2 = (3, {4, 5}), and T_2 = (1, {5}). Table 1 lists the slot assignments on the links comprising the paths of the connections. Slot 1 is allocated to tree T_1 (in red color), slot 2 to tree T_2 (in yellow color), and slot 3 to tree T_3 (in green color).

We then consider a new connection request in this network, denoted as m_k = (1, {2, 5}, 2), with source 1 and destinations 4 and 5, and a bandwidth demand of 2 slots. Establishing this connection requires solving the routing and slot assignment problems. The tree obtained after solving the routing problem for this connection request is denoted by T_k (in blue dashed lines). The slot assignment problem on the T_k tree for this connection has no solution. This is because there are no two available slots that satisfy contiguous and continuous constraints on the links comprising the $T_k$tree. As a result, connection m_k is blocked. However, a reallocation of slots allocated to existing connections can potentially allow the acceptance of the new connection request m_k.

The reallocation solution for this request involves identifying a block of two slots on the links of the T_k tree for which all the connections that intersect it can be reconfigured. Reconfiguring these connections will free up the necessary slots to establish the new connection request. The candidate slot blocks that, when reallocated, free up the resources for the new connection request m_k (m_k = (1, {2, 5}, 2)) are {1; 2} and {2; 3}. For the candidate slot block {1,2}, a reallocation solution involves reallocating slot 4 to trees T_1 and T_2 to free up the slot block {1,2} and allocate it to T_k. For the candidate slot block {2,3}, a reallocation solution involves reallocating slot 1 to tree T_2 and slot 4 to tree T_3 to free up the slot block {2,3} and allocate it to T_k. Considering that there are two reconfigured connections in each of these blocks, a reallocation algorithm used for unicast connections can randomly select one of these blocks. If the slot block

{2; 3} is chosen, the process will reconfigure two connections and impact 5 destinations, as shown in Table 2. If the slot block {1; 2} is chosen, the process will reconfigure two connections and impact 3 destinations, as shown in Table 3. Slot reallocation algorithms used for unicast connections are, therefore, not suitable for multicast connections. Next, we propose a slot reallocation algorithm that minimizes the number of impacted destinations during the process

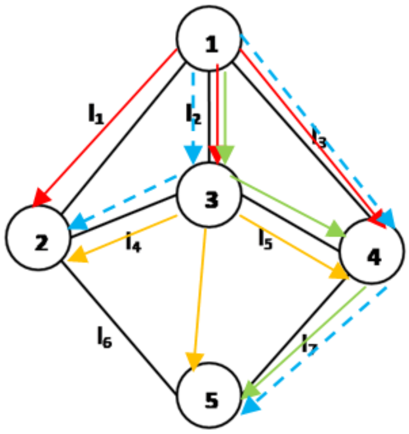

Fig. 1. An instance of the spectrum reallocation problem

Table 1. Network state with block of slots

Liens	1	2	3	4
l_1				
l_2				
l_3				
l_4				
l_5				
l_6				
l_7				
Number of connections reconfigured per block	2			
		2		
			1	
Number of affected destination nodes (users) per block	5			
		3		
			1	

Table 2. Reallocation with minimum existing traffic reconfigured

Liens	1	2	3	4
l_1				
l_2				
l_3				
l_4				
l_5				
l_6				
l_7				
Number of connections reconfigured per block		2		
Number of affected destination nodes (users) per block		5		

Table 3. Number of users (nodes) aware reallocation

Liens	1	2	3	4
l_1				
l_2				
l_3				
l_4				
l_5				
l_6				
l_7				
Number of connections reconfigured per block	2			
Number of affected destination nodes (users) per block	3			

4 Proposed Methods

4.1 Slots Reallocation Algorithm With Minimum Nodes Reconfigured

To minimize the number of reconfigured users, we select, among the reallocation solutions, the one for which the candidate block minimizes the number of recipient nodes according to Reallocation with Minimum Nodes Disrupted (see Algorithm I) denoted by RMNDA. RMNDA takes as input the graph G= (E, V), the set of already established multicast connection denoted by M, T_k, which is the tree selected for the blocked request k = (s_k, D_k, n_k), and a binary variable Sol, which takes True if a solution is found. The first step ofRMNDA consists in creating a list B of all candidate blocks on T_k, each consisting of n_k contiguous slots, where n_k is the number of slots required by the request (refer to line 1 of Algorithm I). For each block b_i in B, RMNDA computes the list L_i of all already established connections that intersect the T_k tree on block b_i (refer from line 2 to line 5 of Algorithm I). At the end of this step, each candidate block b_i is associated with the group L_i of established connections. In the second step, RMNDA calculates the total sum of destinations nodes for the connections in each group L_i. The connection groups are then sorted in ascending order based on the total number of destination nodes (refer to line 7 of Algorithm I). Additionally, the blocks are sorted in the same order as their corresponding L_i (refer to line 8 of Algorithm I). At the end of this step, the blocks belonging to Bare arranged in ascending order according to their number of destination nodes.In the third step, RMNDA traverses the blocks established in line 8 to solve the slot reallocation problem for these blocks. The first block b_i for which slot reallocation is possible becomes the solution to the problem. This block is the elected block. Then, all the slots of the connections in the L_i group associated with $b_i$are reallocated. The slots of b_i used by these connections are released simultaneously using the hop-tuning technique (refer from line 10 to line 16 of Algorithm II). These slots are used to establish the new connection. RMNDA stops when a block is elected or when all the candidate blocks in B have been processed.

4.2 Dynamic Multicast Routing And Spectrum Assignment Algorithm With Reallocation

We present our approach (see Algorithm II) denoted by MURHto solvethe dynamic multicast routing and spectrum allocation problem also called or reallocation problem. For a new connection request, the routing problem is solved using Dijkstra's algorithm [17]. MURH determines *k* trees with the smallest number of links. The second subproblem is that of slot allocation, which is resolved using the First-Fit algorithm. This algorithm allocates the first available contiguous slots on each link of the connection tree. If the allocation fails, then existing connections in the network are reconfigured using the proposed reallocation algorithm denoted by RMNDA (refer to line 24 of Algorithm II). If a block is elected (Sol = True), then the current selected tree is established, and the connection request is accepted into the network. If the reallocation algorithm fails, then the connection request is blocked (refer from line 25 to line 32 of Algorithm II).

Aglorithm I: Reallocation with Minimum Nodes Disrupted Algorithm(RMNDA)

Input: G = (E, V), M, Tk, (sk, Dk, nk), Sol = False
Output : Reallocation decision
1 Create List B of all candidate blocks for T_k
2 **While**B$\neq \emptyset$**do**
3 Select a block $b_i \in B$ // Choisir un bloc
4 Create List Li of all connections intercepting T_k on bi
5 $B = B \setminus \{b_i\}$
6 End
7 Sort L_i in ascending order based on the total number of destination nodes in the trees of L_i
8 Sort b_i in B in the same order as L_i
9 While (B $\neq \emptyset$ ou Sol = false) do
10 Select b_i in the order obtained in 8
11 **If** a reallocation solution exists for b_i**then**
12 Reallocate the corresponding slots in parallel for each of the trees in L_i.
Free up the old slots used by the trees in L_i.
13 **Sol** = true
14 Else
15 $B = B \setminus \{b_i\}$
16 End
End

Algorithm II: Minimum User Reallocation Hoptuning (MURH)

Input : Physical Topology Graph G = (V, E), Multicast Request (s_k, D_k, n_k), A: List of k-trees for (s_k, D_k, n_k), Sol=False
Output : Connection request decision
1 For a request (s_k, D_k, n_k)
2 While (n < k) do
3 While $D_k \neq \emptyset$**do**
4 Select a destination $d \in D_t$
5 If there is a path from s_k to d **then**
6 Find the shortest path P_d from s_k to d (Dijkstra)
7 $E_n = E_n \cup P_d$
8 $D_k = D_k - \{d\}$
9 End
10 End
11 Construct the tree T_n with the links from E_n
12 $A = A \cup T_n$
13 n = n + 1
14 End
15 While e < k do
16 $T_k = T_e$ // Select $T_e$as the tree of the request
17 **If** The resources to establish T_k are available **then**
18 Allocate resources to T_k
19 Accept the session (s_k, D_k, n_k)
20 **Else**
21 Run algorithm I
22 **If (Sol** == True)**then**
23 Go to 18
24 End
25 End
26 e = e + 1
27 End
28 Reject the session (s_k, D_k, n_k)
End

5 SIMULATION AND RESULTS ANALYSIS

5.1 DYNAMIC MULTICAST ROUTING AND SPECTRUM ASSIGNMENT ALGORITHM WITH REALLOCATION

The proposed approach MURH was compared to two other multicast connection routing and slot allocation approaches. The first algorithm is classic routing and slot allocation algorithm without spectrum reallocation, MRSA (Multicast Routing and Slot Allocation) [6]. The second algorithm is the routing and slot allocation algorithm with minimal reallocation of connections, MCRH (Minimum Connection Reallocation Hoptuning) [16]. Standard network topologies with different characteristics were used, including the physical network topologies USA-Backbone, NSFNet, and cost 239. The simulations were conducted using the simulator called FlexgridSim [17]. The traffic demand type considered here is multicast, which involves one source and multiple destinations. Connection requests were generated randomly following a uniform distribution, and each source node and the set of destination nodes for the request belong to set V. Each fiber had a capacity of 150 FS (Frequency Slots), and each slot had a width of 12.5 GHz. The bandwidth requirements for each multicast traffic request were evenly distributed between 1 and 8 slots, and optical tree requests were established using the first-fit approach. The inter-arrival time between connection requests and the connection holding time were generated according to an exponential distribution [6]. For each network, between 100 and 1000 multicast connections were randomly generated, with the number of destinations ranging from 2 to |V|-2. All the nodes in the network have multicast capabilities. The simulation scenario is as follows: when a connection request arrives, we used a offline manner to calculate the number of different trees from the source to various destinations using the Dijkstra algorithm. For our simulations, we used the first two calculated trees as candidates for routing the request to limit computation time (k=2). For each traffic load (Erlang) in each network, and for a number of 1000 connection requests, 10 simulations were conducted. The evaluation metrics include the connection request blocking rate, the rate of reconfigured destinations, and the fragmentation rate. The average link fragmentation rate in the network is calculated for every 100 processed connection requests. These three metrics are detailed in the following section.

5.2 PERFORMANCE METRICS

Three metrics are evaluated here, namely, the blocking probability, the number of users reconfigured and the fragmentation rate.

- The blocking probability (BP) is defined as the ratio of the total number of blocked connection requests to the total number of requested connections. Let B be the number of blocked requests, R be the number of connection requests, and BP be the blocking probability. Equation (1) gives the expression of BP

$$BP = \frac{B}{R} \quad (1)$$

- The fragmentation rate assesses the proper distribution of slots in the network. The metric used here is Shannon's entropy [18]

$$FP = \sum_{i \in I} \frac{f_i}{S} \ln\left(\frac{S}{f_i}\right) \quad (2)$$

Where S is the number of slots per network link, and f_i is a contiguous block of i slots.

Every 100 processed connection requests and for each Erlang load, the overall network fragmentation rate was calculated by averaging the fragmentation rate of the network's links.

- The numbers of destination nodes reconfigured names numbers of users reconfigured in each graph

$$Nb_{User} = \sum \gamma_i x_i \quad (3)$$

Where γ_i is a binary variable equal to 0 if a destination node x_i is reconfigured and 0 otherwise.

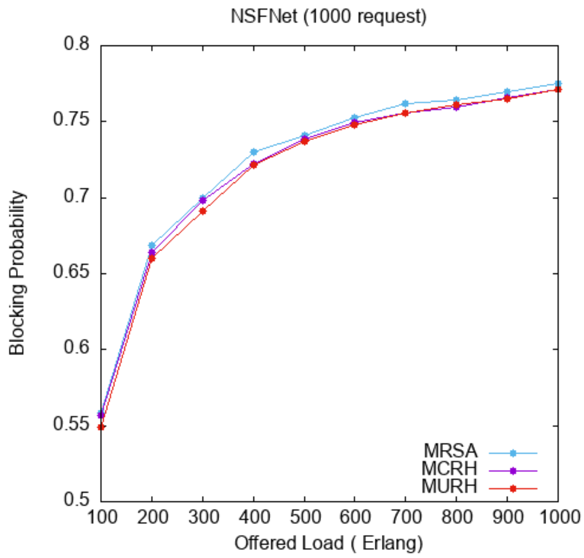

Fig. 2. Blocking probability in NSFNet network

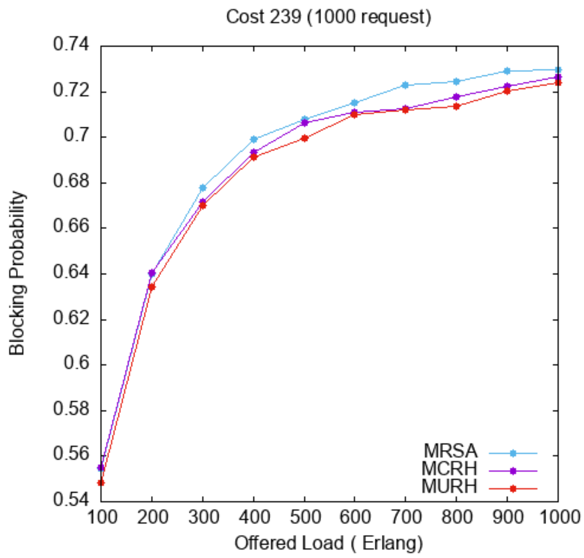

Fig. 3. Blocking probability in COST 239 network

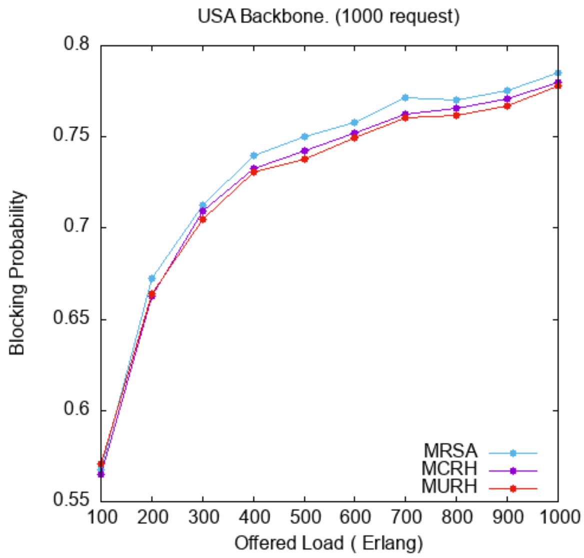

Fig. 4. Blocking probability USA backbone network

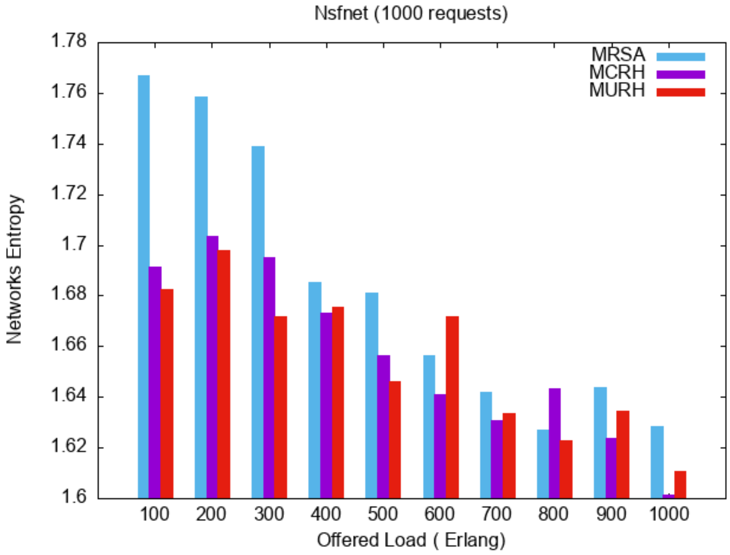

Fig. 5. Average of fragmentation rate in NSFNet

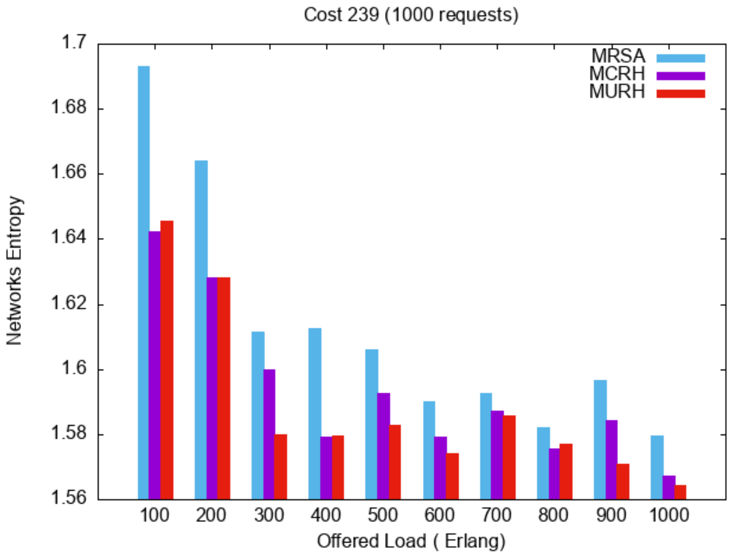

Fig. 6. Average of fragmentation rate in COST 239

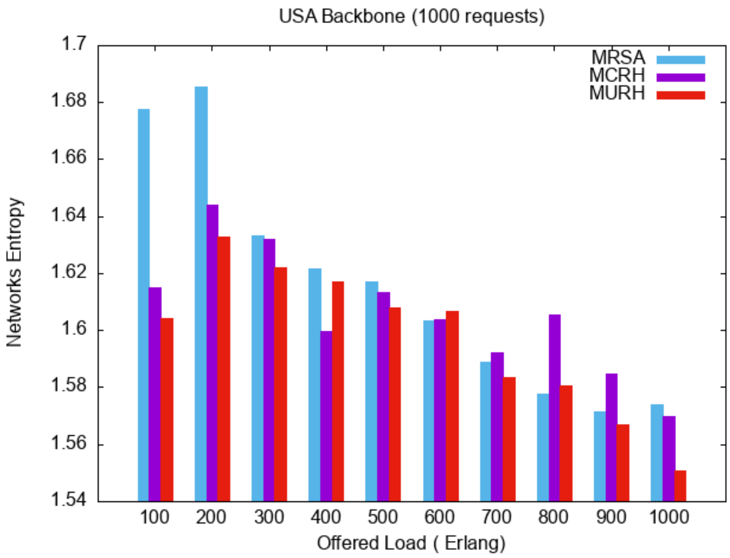

Fig. 7. Average of fragmentation rate in USA Backbone

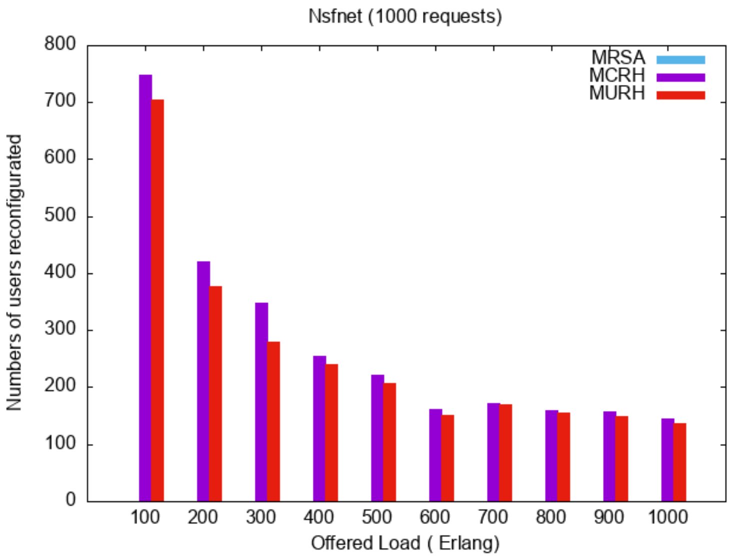

Fig. 8. Numbers of nodes reconfigured in NSFNet

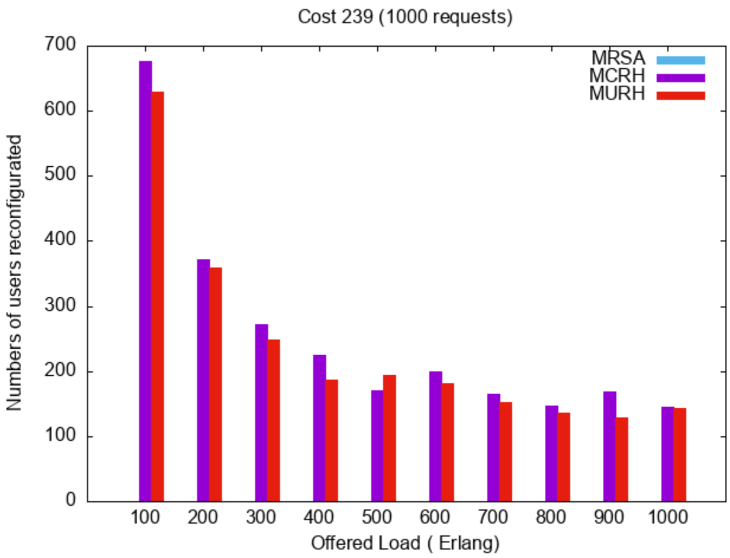

Fig. 9. Numbers of nodes reconfigured in COST 239

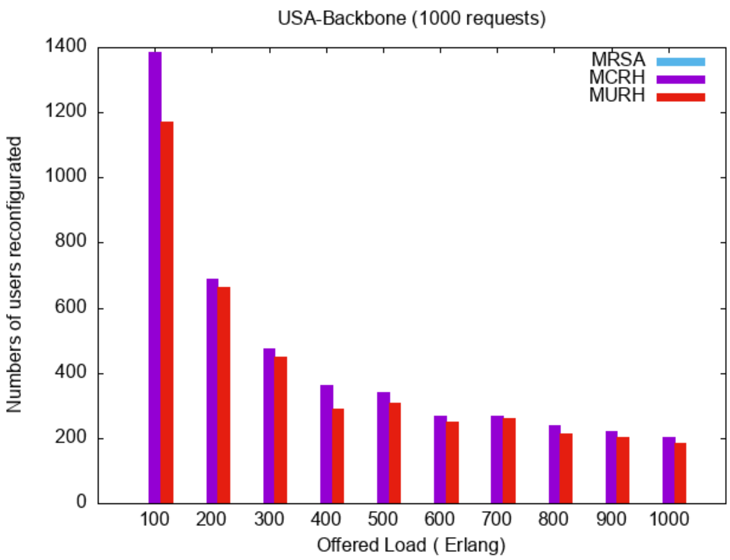

Fig. 10. Numbers of nodes reconfigured in USA Backbone

5.3 Analysis And Discussion

The results of the various simulations are shown in Figures 2 to 10. Figures 2, 3, and 4 show the results of the blocking probability for the MRSA, MCRH, and MURH algorithms, respectively in NSFNet network, COST 239 network and USA backbone network. It is evident that the MRSA algorithm consistently results in the highest blocking probability, regardless of the network. This can be explained by the fact that MRSA does not perform spectrum reallocation compared to the MCRH and MURH approaches. In the NSFNet network, the MCRH and MURH algorithms have very similar blocking probabilities. This is due to the characteristics of the NSFNet network, which has fewer nodes. In such conditions, reallocation solutions with a minimum number of reconfigured connections come closer to those with a minimum number of affected destinations. Under the Cost 239 and USA Backbone networks, the difference between the two approaches is more apparent, especially as the network load increases, with an advantage for our MURH approach, which achieves a slightly lower blocking probability than the MCRH algorithm. This is because when reconfiguring fewer destinations, it translates to reallocating slots along fewer paths, as each reconfigured destination node corresponds to a path from the source of the reconfigured connection to the reconfigured destination node. The MURH solutions create less disruption in the alignment of slots in the network than the MCRH algorithm. This is confirmed by the fragmentation rate recorded in Figures 5, 6, and 7, where it is observed that the MURH algorithm achieves an average reduction in the fragmentation rate compared to MCRH of 0.08% in the NFSNet network, 1.5% in the Cost 239 network, and 0.55% in the USA Backbone network. Regarding the metric of the number of reconfigured destinations, there is a clear advantage of the MURH algorithm compared to the MCRH algorithm, regardless of the network or network load considered. Since no reallocation operations are performed in the MRSA algorithm, no destination is affected during its execution. In contrast, when comparing MURH and MCRH, MURH reduces the number of affected users by 7% in the NSFNet network compared to the MCRH algorithm. Similarly, MURH reduces the number of affected users by 8% compared to the MCRH algorithm in the Cost 239 network. Finally, in the USA Backbone network, this difference reaches up to a 10.5% reduction in the number of destinations.

6 Conclusion

In this work, we addressed the problem of multicast connection allocation. This problem consists of satisfying an initially blocked multicast connection request and reduces the blocking probability.

To overcome this problem, an allocation algorithm was proposed and implemented in a dynamic multicast traffic scenario on the NSFNET, USA Backbone, and Cost 239 topologies. It was demonstrated that the proposed algorithm was capable of satisfying incoming new demands that were initially blocked by reconfiguring a minimal number of existing destinations in the network. A comparison using the metrics of blocking probability, fragmentation rate, and the number of reconfigured destinations was performed with the MRSA and MCRH algorithms. It was concluded that our approach, named MUCRH, reconfigures the smallest number of existing destinations in the network while maintaining or even improving blocking probability performance for higher network loads.

However, the proposed approach does not take into account energy consumption, which is a major consideration in today's choice of reallocation solutions. A future work could involve integrating energy efficiency into this approach using artificial intelligence methods

References

[1] W. Liu, P. Wang, Y. Meng, C. Zhao, and Z. Zhang, «Cloud spot instance price prediction using kNN regression,» Human-centric Computing and Information Sciences, vol. 10, article no. 34, 2020. https://doi.org/10.1186/s13673-020-00239-5.

[2] V. Martin, J. P. Brito, C. Escribano, M. Menchetti, C. White, A. Lord, et al., «Quantum technologies in the telecommunications industry,» EPJ Quantum Technology, vol. 8, article no. 19, 2021. https://doi.org/10.1140/epjqt/s40507-021-00108-9.

[3] Sharma, Deepak & Kumar, Suresh. (2017). An Overview of Elastic Optical Networks and its Enabling Technologies. International Journal of Engineering and Technology. 9. 1643-1649. 10.21817/ijet/2017/v9i3/170903022.

[4] Sergio Fernández-Martínez, Benjamín Barán, Diego P. Pinto-Roa, Spectrum defragmentation algorithms in elastic opticalnetworks, ISSN 1573-4277, https://doi.org/10.1016/j.osn.2019.04.001.

[5] B. C. Chatterjee, S. Ba and E. Oki, «Fragmentation Problems and Management Approaches in Elastic Optical Networks: A Survey,» in IEEE Communications Surveys & Tutorials, vol. 20, no. 1, pp. 183-210, Firstquarter 2018, doi: 10.1109/COMST.2017.2769102.

[6] M. Moharrami, A. Fallahpour, H. Beyranvand and J. A. Salehi, «Resource Allocation and Multicast Routing in Elastic Optical Networks,» in IEEE Transactions on Communications, vol. 65, no. 5, pp. 2101-2113, May 2017.
doi: 10.1109/TCOMM.2017.2667664.

[7] Walkowiak, K., Goscien, R., Klinkowski, M., & Wozniak, M. (2014). Optimization of Multicast Traffic in Elastic Optical Networks With Distance-Adaptive Transmission. IEEE Communications Letters, 18 (12), 2117-2120. doi: 10.1109/lcomm.2014.2367511.

[8] Velasco, L., Vela, A. P., Morales, F., & Ruiz, M. (2017). Designing, Operating, and Reoptimizing Elastic Optical Networks. Journal of Lightwave Technology, 35 (3), 513-526. doi: 10.1109/jlt.2016.2593986.

[9] Alberto Castro, Luis Velasco, Marc Ruiz, Mirosław Klinkowski, Juan Pedro Fernández-Palacios, Davide Careglio, Dynamic routing and spectrum (re) allocation in future flexgrid optical networks, Computer Networks, Volume 56, Issue 12, 2012, Pages 2869-2883, ISSN 1389-1286, https://doi.org/10.1016/j.comnet.2012.05.001

[10] Rajewski, Remigiusz (2018). Defragmentation in W-S-W Elastic Optical Networks. Journal ofTelecommunications and Information Technology. 1. 18-23. 10.26636/jtit.2018.123317.

[11] H. Duong, B. Jaumard, D. Coudert and R. Armolavicius, «Efficient Make Before Break CapacityDefragmentation,» 2018 IEEE 19th International Conference on High Performance Switching and Routing (HPSR), 2018, pp. 1-6, doi: 10.1109/HPSR.2018.8850754.

[12] N'takpe «Protected Light-Tree Reconfiguration Without Flow Interruption in Elastic Optical Networks» DOI: 10.22247/ijcna/2021/209185, International Journal of Computer Networks and Application 2021.

[13] P. Soumplis, K. Christodoulopoulos and E. Varvarigos, «Dynamic connection establishment and network re-optimization in flexible optical networks,» 2014 International Conference on Optical Network Design and Modeling, 2014, pp. 61-66.

[14] S. Fernández-Martínez, B. Barán, and D. P. Pinto-Roa, «Spectrum defragmentation algorithms in elastic optical networks,» Opt. Switch.Netw., vol. 34, no. July, pp. 10–22, 2019.

[15] Moniz, Daniela & Eira, Antonio & de Sousa, Amaro & Pires, João (2017). On the Comparative Efficiency of Non-Disruptive Defragmentation Techniques in Flexible-Grid Optical Networks. Optical Switching andNetworking. 25. 10.1016/j.osn.2017.04.001.

[16] S. K. Posam, S. K. Bhyri, R. Gowrishankar, V. N. Challa and S. S. S. Sanagapati, «Reactive Hitless Hop tuning based defragmentation algorithm for enhanced spectrum efficiency in Elastic optical networks,» 2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), 2020, pp. 1-6.

[17] N'takpe «Protected Light-Tree Reconfiguration Without Flow Interruption in Elastic Optical Networks» DOI: 10.22247/ijcna/2021/209185, International Journal of Computer Networks and Application 2021.

[18] Wright, M. C. Parker, and A. Lord, «Simulation results of Shannon entropy based flexgrid routing and spectrum assignment on a real network topology,» in ECOC, London, UK, Sept. 2013, paper We.2.E.4