A New Arc-Disjoint-Trees Scheme for Survivable Multicasting in Mixed-Graph Sparse-Splitting Optical Networks

Luís Raposo*, Teresa Gomes*†, Lúcia Martins*†, Costas K. Constantinou‡ and Georgios Ellinas ‡
*Department of Electrical and Computer Engineering, University of Coimbra, 3030-290 Coimbra, Portugal
Email: lm.raposo91@gmail.com, {teresa,lucia}@deec.uc.pt
†INESC Coimbra, 3030-033 Coimbra, Portugal
‡KIOS Research Center and Department of Electrical and Computer Engineering, University of Cyprus, Nicosia 1678, Cyprus
Email: {constantinou.k.costas, gellinas}@ucy.ac.cy

***Abstract*—This work addresses the problem of survivable multicast request provisioning in mixed-graph optical networks, where only a fraction of the nodes have optical splitting capabilities. An effective scheme for the calculation of a pair of disjoint trees, namely the New Arc-Disjoint Trees (NADT) protection scheme, is presented. The key idea of this technique is to gradually construct the primary tree, verifying that after the addition of each one of the destinations of the multicast session, a secondary (arc-disjoint) tree can still be obtained. The proposed protection technique is combined with two existing heuristics for multicast routing in mixed-graph sparse-splitting networks. Performance results demonstrate that the proposed NADT protection technique clearly outperforms the conventional approach in terms of blocking ratio, and presents a negligible increase of the average cost of the derived pair of arc-disjoint trees.**

***Index Terms*—survivability; multicasting; optical networks; sparse splitting; mixed graph**

I. Introduction

Nowadays, optical networks are expected to be able to support heterogeneous traffic demands, ranging from unicast connections for the support of enterprise customers, to high-bandwidth multicast applications such as high-definition television, live auctions, distributed games, amongst others. This network capability has become feasible because of the increased capabilities provided mainly by the intelligent switching nodes present at the optical transport networks. Reconfigurable optical add-drop multiplexers and optical cross-connects are examples of such switching nodes that can be utilized to support multicast applications in the optical domain.

Multicast connectivity in the optical domain is based on the calculation and provisioning of light-trees that connect a source node with a set of destinations. In order to set-up these light-trees, the utilization of optical splitters in the network nodes is required [1]. The nodes that have splitting capability are called *Multicast-Capable* (MC) nodes, while the other ones are called *Multicast-Incapable* (MI). In practice, only a fraction of the network nodes, strategically placed at certain node locations during the network design phase, are equipped with optical splitters, aiming to provide efficient multicast connectivity while keeping the network cost low (by not utilizing MC nodes throughout the entire network), resulting in a *sparse-splitting* network [1], [2]. The remaining MI nodes of the network may be *Drop-and-Continue* (DaC) [3] or *Drop-or-Continue* (DoC) [4]. A DaC node can transmit the optical signal to the following node in its path *and* can also drop it locally as well, while a DoC node can either transmit the optical signal to the following node in its path *or* drop it locally. The current paper deals with sparse-splitting networks where the MI nodes are DoC. It is important to note that the problem of multicast routing in sparse-splitting networks is a complex problem, related to the well-known NP-complete Steiner tree problem in graphs [5]. For this problem, relevant polynomial-time multicast routing heuristics that give approximate solutions are used in practice and can be found in [6]–[10] as well as in other sources.

In addition, the vast amount of information that an optical fiber carries, as well as the amount of information loss in case of a failure on a light-tree that can affect traffic to multiple destinations, have led to the development of efficient multicast survivability techniques that can quickly restore the multicast service. One type of survivability technique is the *protection technique* that pre-computes secondary (protection) paths prior to the fault occurrence and switches to these paths once a failure has occurred. Most commonly, network survivability is provided for single-link failures, since these are the predominant types of failures in optical networks. The most straightforward way to protect a multicast tree against single-link failures is the derivation of a pair of arc-disjoint trees (primary and secondary trees) during the connection provisioning phase. After failure detection, a protection switching protocol is subsequently invoked and the traffic is switched from the primary to the secondary tree to ensure continuous

Teresa Gomes and Lúcia Martins acknowledge financial support by Fundação para a Ciência e a Tecnologia (FCT) under project grant UID/MULTI/00308/2013.

Luís Raposo is currently working as a software tester at Coriant Portugal.

information flow [11].

The main focus of this work is the protection of multicast connections in sparse-splitting optical networks, and specifically in mixed-graph sparse-splitting optical networks. Even though all network connections are considered bidirectional, when provisioning a new multicast connection, in practice, the network must be modeled as a mixed graph (having both unidirectional and bidirectional connections between its nodes), since some network arcs may be unavailable due to already existing connections on the network holding network resources. Furthermore, when two arc-disjoint trees must be found on a graph, the secondary tree, after the removal of the primary one, will be calculated on a mixed graph.

In the current paper, a new multicast arc-disjoint protection scheme is proposed, designated New Arc-Disjoint Trees (NADT) protection scheme. The goal of this scheme is to increase the rate of success in obtaining a pair of arc-disjoint trees (without significantly degrading the average cost of the derived pairs of trees), compared to the conventional approach that usually focuses on the derivation of a primary tree, and only afterwards attempts to calculate a secondary tree that is arc-disjoint to the primary one.

The remaining of the paper is organized as follows. Section II presents a brief description of the sparse-splitting multicast routing heuristics that are combined with existing and proposed protection schemes. The conventional and newly proposed protection techniques are discussed in Section III, while Section IV presents the evaluation of their performance. Finally, in Section V the conclusions of the paper are presented, as well as ongoing future work.

II. Brief Overview of the MUS and MSH Multicast Routing Heuristics

In this work two previously proposed algorithms for multicast routing in sparse-splitting optical networks, namely Multicasting Using Splitters (MUS) [12] and Mixed-graph Sparse-splitting Heuristic (MSH) [13], are used to evaluate the performance of the proposed protection scheme. Therefore, a brief description of these heuristics is given below.

The MUS heuristic splits the destination set into MC and MI. It first adds the MC destinations sequentially, in non-decreasing order, according to the cost of the path that connects them with the current tree. Since the network has sparse splitting capability, these paths can originate either from the source node or from an MC node on the tree. After the addition of all MC destinations, the MI destinations are added in a similar fashion.

The MSH heuristic deals with the problem of multicast routing in mixed-graph sparse-splitting optical networks. As in the case of the MUS heuristic, MSH splits the destination set into MC and MI. It first adds the MC destinations sequentially, in non-decreasing order, according to the cost of the path that connects them with the current tree. After the addition of all MC destinations, the MI destinations are subsequently added in a similar fashion. The key characteristic that makes MSH more appropriate than the MUS algorithm, for mixed-graph networks is that, after the addition of a destination, it removes from the tree the already added ones and checks whether they can be connected in a more cost-efficient way through the MC nodes that belong to the path of the last added destination. In more detail, after a (MC or MI) destination y is added, the path from the source node to y is kept as the current tree and the already added destinations are removed and added again to the tree. This procedure is executed first for the already added MC destinations and then for the MI ones.

III. Calculation of a Pair of Arc-Disjoint Multicast Trees

In this section, the most common approach used to compute two arc-disjoint multicast trees is revisited. To surpass the limitations presented by this technique, a new method to obtain a pair of arc-disjoint trees is proposed; this method can also be easily adapted for the derivation of two link-disjoint multicast trees. It is assumed that the weight of the arcs is non-negative, and that the cost of a multicast tree is given by the sum of the weights of the arcs of the tree. Similarly, the cost of a path on the tree is given by the sum of the weights of the arcs of the path.

A. Arc-Disjoint Trees Protection Scheme (Existing)

The *Arc-Disjoint Trees* (ADT) protection scheme was initially presented in [14], for meshed optical networks, in order to calculate a pair of arc-disjoint trees exploiting multicast routing heuristic algorithms. The approach used by that protection scheme to derive the arc-disjoint trees is described as follows:

1) Create a primary tree using any multicast routing heuristic H.
2) Remove the arcs along the primary tree from the original network.
3) Create a backup tree in the remaining graph using H.

Hereafter, the acronym *H-ADT* stands for the combination of multicast routing heuristic H with the *ADT* protection scheme. Application of the ADT protection scheme can be found, among others, in [13] where it is combined with the MSH heuristic, resulting in the MSH-ADT algorithm for provisioning survivable multicasting in mixed-graph sparse-splitting networks. The simulation results in [13] show that the calculation of two arc-disjoint trees using the multicast routing heuristic MSH can reduce the blocking ratio and average cost of the arriving multicast requests, in comparison to the results obtained by other existing algorithms, namely the On-Tree MC Node First (OTMCF) and Nearest MC Node (NMCF) [15], as well as MUS [12] heuristics. In each case, the results were derived combining each multicast routing heuristic with ADT for obtaining a pair of arc-disjoint multicast trees.

An example of the calculation of two arc-disjoint trees, with the MSH-ADT algorithm is shown for the network illustrated in Fig. 1, where s is the source node and nodes d_1, d_2, and d_3 (nodes colored black) are the destination nodes of the multicast session. The MC nodes are square-shaped, and the MI nodes are considered to be DoC (Drop-or-Continue) *i.e.*, they can either transmit the optical signal to the following node *or* drop it locally [3], [4], [16]. The primary tree obtained by the MSH heuristic can be seen in Fig. 2. Following the steps of the secondary tree construction with the ADT scheme, the arcs belonging to the primary tree are removed from the original network. In this new network graph it is impossible to calculate a secondary tree (since the source node does not have any outgoing arcs), despite the fact that there are enough arcs in the network to construct an arc-disjoint secondary (backup) tree for a different primary tree.

Through this example, the main limitation of heuristics using the existing ADT protection scheme was exposed, which is the fact that the construction of the primary tree does not take into account the need for obtaining an arc-disjoint secondary tree. This approach, when calculating the primary tree, focuses on minimizing the cost of the multicast tree. Hence, paths containing the destinations nodes are added to the primary tree under construction, without taking into account whether an arc-disjoint tree can be subsequently obtained. To address this limitation, a new method to calculate two arc-disjoint trees is proposed, as described next.

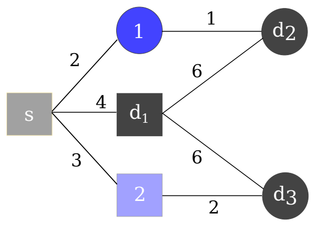

Fig. 1: Network graph.

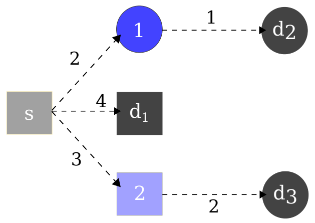

Fig. 2: Tree obtained by MSH (cost=12).

B. New Arc-Disjoint Trees Protection Scheme (Proposed)

The proposed protection scheme for calculating a pair of arc-disjoint trees, utilizing any multicast routing heuristic, is called *New Arc-Disjoint Trees* (NADT) protection scheme, and its procedure is described in Algorithm 1. In the pseudo-code the following notations are used:

- $G = (V, A)$ – network graph consisting of a set V of network nodes (optical switching nodes) and a set A of network arcs (optical fibers) whose elements are ordered pairs of distinct nodes, with a positive cost (weight) assigned to each arc;
- T – primary tree;
- T_{prot} – secondary (protection) tree;
- $r = (s, D)$ – multicast session with source node s, and destination node set D
- $T_{prot_{MC}}$ – set of MC nodes belonging to T_{prot}; this set includes the source node s as well, since in the present work it is assumed that each node is equipped with a bank of tunable transmitters and receivers allowing the source of the multicast session, even in the case of an MI node, to transmit the information through multiple fibers;
- D_{prot} – set of destinations already added to T;
- β – a sufficiently large positive constant; an arc is considered to be removed from G if its weight is greater than β;
- A_{exc} – set of arcs that are excluded from G (every arc in A_{exc} has its cost increased by β);
- $a_{x'y'}$ – arc that connects nodes x' and y', $a_{x'y'} \in A$;
- H – the multicast routing heuristic utilized in the NADT protection scheme.
- $H(G, s, D, T)$ – multicast routing heuristic H executed in graph G, for multicast session with source node s and destination set D, resulting in tree T;
- $H_i(G, s, D, T)$ – iteration of the H heuristic where a destination d_i ($d_i \in D$) is added to T, thus updating T;
- $H'(G, s, D_{prot}, T_{prot})$ – H heuristic terminating as soon as the cost of T_{prot} would become greater than β;
- p_{xy} – set of arcs and nodes of the shortest path that connects nodes x and y (it includes x and y);
- c_{xy} – cost of the path p_{xy} that connects nodes x and y;

Explanation of the NADT Protection Scheme

The NADT protection scheme starts with the normal execution of the multicast routing heuristic H, where a node d_i, belonging to the destination set of the multicast session, is added to a primary tree, initialized with the source node. Every time a destination node is added to the primary tree (represented in line 5 of Algorithm 1 by H_i), the arcs of the partial primary tree are removed from the network graph and an arc-disjoint tree is derived, utilizing heuristic H', for a new multicast session (line 13 of Algorithm 1). This new multicast

```
Algorithm 1: H-NADT
  Input: $G$, $r$, $H$
  Output: $T$, $T_{prot}$
1  begin
2      $T \leftarrow (\{s\}, \emptyset)$ ;            // Initial graph of $T$
3      $D_{prot} \leftarrow \emptyset$;
4      $A_{exc} \leftarrow \emptyset$;
5      for each node $d_i \in D$ added to $T$ during the
       execution of $H_i\,(G, s, D, T)$ do
6          Remove from $G$ the arcs belonging to $A_{exc}$;
7          if $T$ is admissible;                 // cost of $T < \beta$
8          then
9              $A_T \leftarrow$ the set of arcs of $T$;          // saves
10             Remove from $G$ the arcs of $A_T$;
11             Add back to $G$ the arcs in $A_{exc}$;
12             $D_{prot} \leftarrow D_{prot} \cup \{d_i\}$;
13             $H'\,(G, s, D_{prot}, T_{prot})$ ;          // new $T_{prot}$
14             if $\exists d \in D_{prot} : c_{sd} > \beta$ then
                   // $T_{prot}$ does not contain $D_{prot}$
15                 $c_{xy} \leftarrow \max_{i \in T_{prot_{MC}}, j \in D_{prot}} c_{ij}$;
                   // $c_{xy}$ is cost the of path $p_{xy}$
16                 Identify the first arc $a_{x'y'}$ of $p_{xy}$ shared
                   by $p_{xy}$ and $T$;
17                 $A_{exc} \leftarrow A_{exc} \cup \{a_{x'y'}\}$;    // excludes
18                 $T \leftarrow (\{s\}, \emptyset)$;              // restarts $T$
19                 $D_{prot} \leftarrow \emptyset$;
20             end
21             Add back to $G$ the arcs of $A_T$
22         else
23             Add back to $G$ the arcs in $A_{exc}$;
24             $T_{prot} \leftarrow \emptyset$;           // No secondary tree
25             $H\,(G, s, D, T)$ ;    // Only primary tree
26             return $T$,$T_{prot}$ ;    // End of execution
27         end
28     end
29     return $T$,$T_{prot}$ ;              // End of execution
30 end
```

session differs from the original one only in the destination set; its elements are now the destinations already added to the primary tree (line 12 of Algorithm 1). The existence of a secondary tree, for the new multicast session, means that the current primary tree can be protected, thus the remaining destinations of the original multicast session can continue to be added to the primary tree. If during the construction of the secondary tree the multicast routing heuristic fails to add one or more destinations to this tree, a new procedure is executed. In this new procedure, one arc of the primary tree is identified as the reason for which the destination node(s) in question cannot be added to the secondary tree. The identified arc is added to the set of *excluded arcs*, A_{exc} (line 17 of Algorithm 1), and the construction of the arc-disjoint trees is restarted. This procedure is repeated each time the multicast routing heuristic fails to calculate a secondary tree for a given primary tree under construction. Note that the primary tree is only considered to be admissible (see line 7) when it does not use any of the arcs belonging to the set A_{exc} (*i.e.,* when the tree has cost less than β).

Briefly, the primary tree is calculated on a new network graph, where one or more arcs belonging to the set A_{exc}, identified as the ones that prevent the creation of a secondary tree, are successively excluded. This procedure was inspired by the Trap Avoidance algorithm for the calculation of shared risk link group disjoint path-pairs proposed in [17].

The protection scheme, described by Algorithm 1, is completed when either a pair of arc-disjoint trees is obtained for the original multicast session, or the multicast routing heuristic is unable to obtain a primary tree without using any of the excluded arcs (line 7 of Algorithm 1). In the latter case, it is considered that the multicast session cannot be protected and only the primary tree may be derived (lines 23-26 of Algorithm 1).

NADT will be most efficient combined with multicast heuristics where the destination nodes are iteratively added. For other heuristics, when the secondary tree with D_{prot} equal to D has cost larger than β, the iterative procedure of identifying the arc to be removed can still be performed.

Removing Arcs and Identifying the Arcs to be Excluded

The elimination of an arc belonging to the primary tree in the original network graph is done by replacing its weight with a new value, which will be the sum of the arc's original weight with a sufficiently large constant (β). The existence of an admissible secondary tree is confirmed by the absence of arcs with cost greater than β in the final (*i.e.*, containing all destination nodes) protection multicast tree.

The process of identifying an arc to be excluded from the calculation of a primary tree starts when the multicast routing heuristic fails to add one or more destinations to the secondary tree (line 13 of Algorithm 1). That is, when the cost of adding a minimum-cost path containing a destination is greater than β. In the occurrence of this event, the algorithm selects the path of highest cost (always greater than β) from the source node, or from an MC node, to a destination node (line 15 of Algorithm 1). The identified arc to be excluded from the derivation of the primary tree, during the remaining iterations of the algorithm, is the first arc of the path common to the path and the primary tree (line 16 of Algorithm 1). The reason for choosing the largest cost path, amongst all inadmissible paths, is an attempt to reduce the cost of the new primary tree to be calculated.

Example of the NADT Protection Scheme with MSH

Regarding the calculation of the arc-disjoint trees in the network graph illustrated in Fig. 1, for multicast session

$r = (s, \{d_1, d_2, d_3\})$, it was clear that the ADT protection scheme is unable to obtain a pair of arc-disjoint trees. The same problem is now addressed using the MSH-NADT algorithm. Following the procedure of the MSH multicast routing heuristic, and recalling its steps, firstly the MC destinations are added to a tree initialized with the source node s. After the addition of the only MC destination d_1, the heuristic succeeds in constructing a secondary tree for the partial primary tree, leading the MSH heuristic (used internally by MSH-NADT) to the next step, which is the connection of the MI destinations to the primary tree.

The connection of d_2 to the primary tree, and the successful construction of the secondary tree for the current destinations in the primary tree (d_1 and d_2), precede the addition of node d_3 to the primary tree, whose final result is illustrated in Fig. 2. As already seen, this primary tree cannot be protected. To overcome this difficulty, NADT identifies the arc connecting nodes s and d_1, which will be excluded from the network graph, before deriving the next candidate primary tree. Note that MSH starts by adding to the tree all MC destinations which, in the present example, is only the node d_1. The modified network graph is illustrated in Fig. 3. The construction of the pair of arc-disjoint trees is then restarted and the execution of the technique leads to the multicast trees illustrated in Figures 4 and 5, which represent, respectively, the primary and secondary trees for the multicast session. In Fig. 4 the different lines represent different wavelengths.

Example of the NADT protection scheme with MUS

For the same previously considered network (Fig. 1) and multicast section $r = (s, \{d_1, d_2, d_3\})$, MUS-NADT obtains the same candidate primary tree as MSH-NADT (Fig. 2). Furthermore, the same arc (connecting nodes s and d_1) will be excluded (Fig. 3) from the original network graph, since the first set of nodes added by the multicast heuristic is the set of MC destinations. The procedure is restarted and the new primary tree is successfully calculated in a network where the previously mentioned arc was removed. This primary tree can be seen in Fig. 6, where (as in Fig. 4) different lines represent different wavelengths. The secondary tree will be the same as in the MSH-NADT case (Fig. 5), but the total cost of MSH-NADT will be 32 and of MUS-NADT will be 33.

The two presented examples indicate that, regarding the total cost, MSH-NADT and MUS-NADT will tend to present the same relative performance as MSH-ADT and MUS-ADT [13]. This will be confirmed by the results in Section IV.

IV. Performance Evaluation

In this section, the proposed NADT protection scheme is evaluated and compared with the existing ADT scheme, for mixed-graph, sparse-splitting networks with sparse wavelength conversion (*i.e.*, only the MC nodes are equipped with wavelength converters). All the MI nodes are considered to be

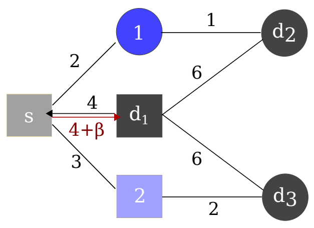

Fig. 3: Modified network graph for the calculation of the primary tree with excluded arc (its weight is increased by β).

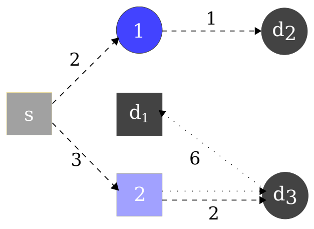

Fig. 4: Primary tree obtained by MSH in the new network graph (cost=16).

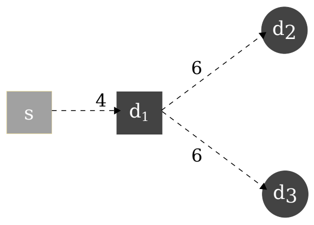

Fig. 5: Secondary tree obtained by MSH and MUS after the calculation of the primary tree in the new network graph (cost=16).

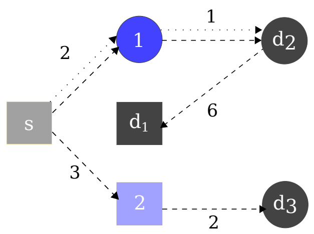

Fig. 6: Primary tree obtained by MUS in the new network graph (cost=17).

DoC. To study the performance of both protection schemes, combined with existing multicast routing heuristics (MSH and MUS), the results were obtained through simulations and compared in terms of: i) average cost of the pair of arc-disjoint trees; ii) blocking ratio (*i.e.*, the number of requests that were not established due to the fact that a pair of arc-disjoint trees could not be derived, over the total number of arrival requests).

The objective of this work was to show the ability of the proposed strategy in finding a pair of arc-disjoint trees for a demand in a given network, without being constrained by capacity. Hence network links have unlimited capacity, and the blocking probability derives from the algorithm's inability to obtain a pair of arc-disjoint trees. The cost of each tree depends on the number of arcs and on the different number of wavelengths used in each arc of the network, which depends on the MC nodes present in the tree.

A. Simulation Set Up

Two undirected-graph networks were randomly created with the Doar-Leslie model [18] using the GT-ITM Georgia Tech Internetwork Topology Models software. The first one consists of 40 nodes and 217 bidirectional connections, while the second one consists of 50 nodes and 177 bidirectional connections. The cost of each link is in the interval [1,100] (for both networks) and the mean cost and respective standard deviation is 42 and 22.2 for the first network, and 40.5 and 20.8 for the second. To convert the undirected-graph networks to mixed-graph networks, half of the bidirectional connections were transformed to unidirectional, thus leading to a *Percentage of Directionality* (PoD) (the ratio of the unidirectional connections over the total number of network connections) equal to 50%. The candidate edges to be transformed were selected randomly between only those whose endpoints had out-degree and in-degree greater than 2, ensuring (possible) protection for every node. In the resulting mixed-graph network, 5 nodes were selected to be MC; these were chosen using the *kmaxD* method as described in [19] (i.e., the MC nodes were placed at the nodes that have the largest degree). Note that, similarly to [13], this work deals only with the survivable routing problem, assuming that the MC nodes were already placed, and therefore does not address the problem of MC node placement.

Although the underlying graph is a mixed graph, the algorithms for tree generation consider the bidirectional connections to be represented as a pair of symmetrical arcs – note that the number of occupied wavelengths in those topologically symmetrical arcs can differ. Moreover an arc can be in the primary tree and its symmetrical in the secondary tree. Therefore, the algorithms consider a directed graph representation, with a given percentage of unidirectional connections (pair of nodes linked by a single directed arc).

The results were obtained under the following conditions:

- All the nodes of the network were used as source nodes and for each source node the multicast group size, D (number of destinations), ranged between 2 and 20. MC nodes are not excluded from the set of destination nodes.
- The number of runs per network was 380000 (40 source nodes $\times$ 500 sessions $\times$ 19 multicast group sizes for the network with 40 nodes and $50 \times 400 \times 19$ for the network with 50 nodes).
- For a given source, the same destination group was never repeated. The source node was excluded from the elements in the destination group.
- The average cost of the derived arc-disjoint trees was obtained over all multicast group sizes (regardless of the source node).
- The blocking ratio was also obtained for all the multicast group sizes regardless of the source node.
- The final results were obtained by executing 10 simulations for each network.

  The mean value of those ten simulations is presented in the graphs and the variance of the ten samples around that mean is shown as error bars (barely visible in Figs. 7 and 9).

B. Results and Analysis

The results of the simulations, for the network with 40 nodes, are given in Figs. 7 and 8, while Figs. 9 and 10 illustrate the results for the 50-node network. In terms of blocking ratio, the NADT scheme outperforms the ADT scheme for both multicast routing heuristics. For the network with 40 nodes, the multicast requests presented zero blocking and for the network with 50 nodes, only a very small number of requests are rejected (less than 0.001% of all the multicast requests for both MUS-NAT and MSH-NADT, although not visible in Fig. 10) due to the non existence of a secondary tree. This is due to the fact that the proposed scheme, during the derivation of the primary tree, takes into account that a secondary tree, arc-disjoint to the primary one, must be derived as well, whereas the ADT scheme ignores this, focusing only on the derivation of the low-cost primary trees.

As for the results obtained for the average cost of the pair of arc-disjoint trees, the new scheme presents a small increase, almost undetectable, of the average cost of the derived pair, as can be seen in Figs. 7 and 9, where the cost obtained by MSH-NADT and MUS-NADT is compared with MSH-ADT and MUS-ADT, respectively. This very slight increase is also due to the larger number of established multicast requests, which led to a higher number of calculated arc-disjoint trees.

Note that, regarding the cost, the relative performance of MSH-NADT and MUS-NADT follows the pattern in [13], where MSH-ADT has advantage over MUS-ADT. Furthermore, when MUS-ADT is able to find a pair of disjoint trees, the same solution will be obtained by MUS-NADT; hence the slightly larger average cost (only visible in Fig. 9) of the solutions of MUS-NADT results from the new solutions which MUS-ADT was unable to calculate. However, in MSH-ADT and MSH-NADT this may not be the case, because MSH rebuilds its tree several times, and in MSH-NADT such procedure may also depend on NADT. With MSH-NADT, as the proposed scheme is primarily focused on finding a pair of arc-disjoint trees (and only secondly on minimizing the primary tree cost) it is expected that the cost of the derived pair will be slightly higher compared to the conventional approach. However, the significant gain in terms of blocking ratio achieved via the newly proposed technique significantly outweighs this small average cost increase for the pair of arc-disjoint trees.

Regarding execution times, the conducted experiences have shown that, per run, the duration of MSH-NADT surpasses MUS-NADT by a factor of approximately 2.33. The average execution time and the 95 % confidence interval of the simulations (380000 runs) in the 50 node networks with MSH-NADT and MUS-NADT was respectively 188.98 ± 4.61 and 81.00 ± 2.58. This factor can be explained by the tree reconstruction every time a destination is added, resulting in the aforementioned time increase. The executions times were not calculated per run since this value can fluctuate greatly according to the number of multicast group sizes (a run with $D = 20$ will always have a bigger execution time than, for instance, $D = 5$). The execution times were obtained in a laptop with an Intel(R) Core(TM) i7-2670QM CPU @ 2.20GHz with a 4GB memory RAM.

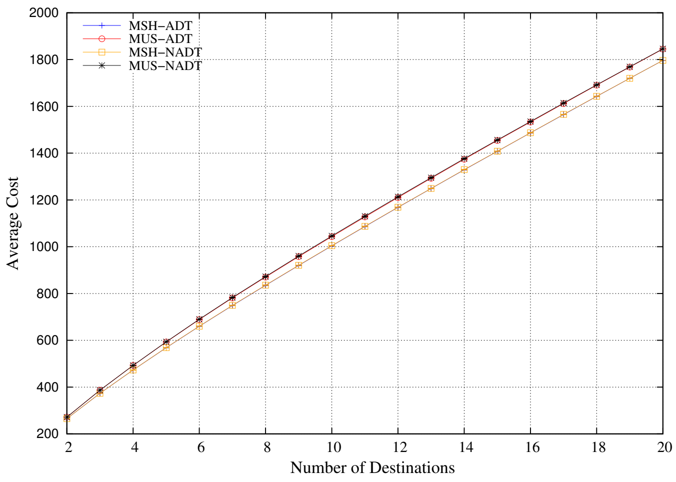

Fig. 7: Average cost of the arc-disjoint trees vs number of destinations, for the network with 40 nodes.

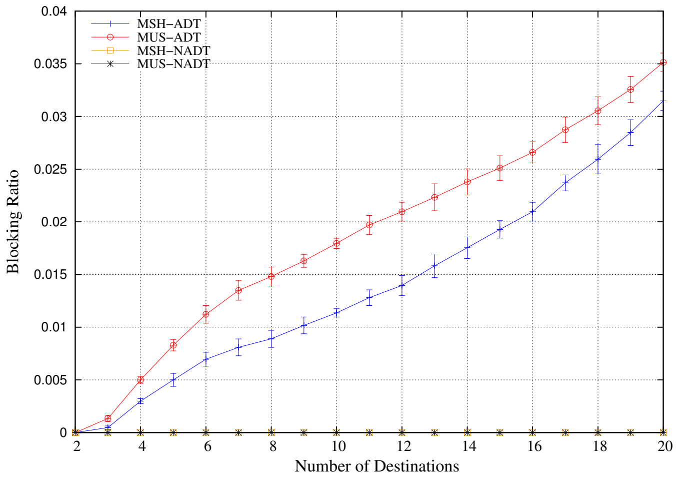

Fig. 8: Blocking ratio vs number of destinations, for the network with 40 nodes.

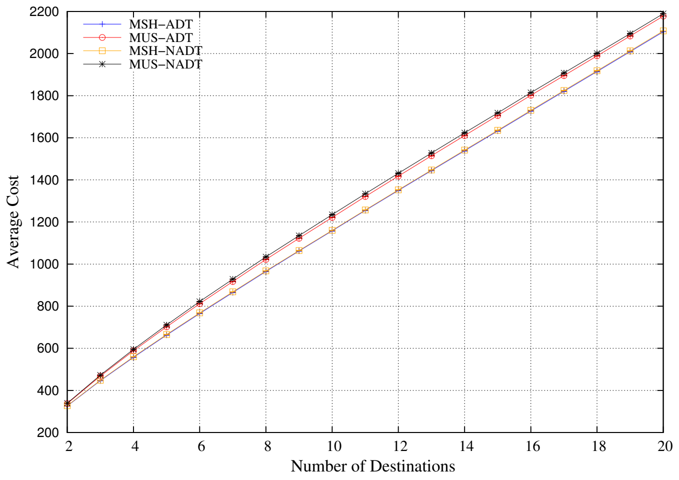

Fig. 9: Average cost of the arc-disjoint trees vs number of destinations, for the network with 50 nodes.

V. Conclusion

The proposed work dealt with the subject of provisioning survivable multicast requests in mixed-graph optical networks with sparse-splitting capabilities. Conventional approaches focus on the derivation of a primary tree of minimum cost, and only afterwards attempt to calculate a secondary tree that is arc-disjoint to the primary one. This can result in a false trap problem, since it is possible that a secondary tree that is arc-disjoint to the primary one may not be found.

An effective scheme for the calculation of a pair of disjoint trees, namely the New Arc-Disjoint Trees Protection Scheme (NADT), has been presented. This technique focuses on the gradual construction of the primary tree, verifying that after the addition of each one of the destinations of the multicast session, a protection tree can still be obtained. The proposed protection scheme was combined with two existing heuristics for multicast routing in sparse-splitting networks. Simulations have shown that the proposed NADT protection scheme clearly outperforms the relevant existing ADT technique in terms of blocking ratio, while leading only to a slight increase of the average cost of the derived pair of trees.

Ongoing work focuses on the application of the proposed NADT protection scheme in mixed-graph sparse-splitting networks with DaC nodes [16], as well as on the embedding of NADT into routing heuristics for networks with capacity constraints. Furthermore, the performance evaluation of the adaptation of the underlying idea of NADT for the derivation of two link-disjoint multicast trees in optical networks with sparse-splitting capabilities is also being considered.

References

[1] L. Sahasrabuddhe and B. Mukherjee, “Light-trees: Optical multicasting for improved performance in wavelength-routed networks,” *IEEE Com-*

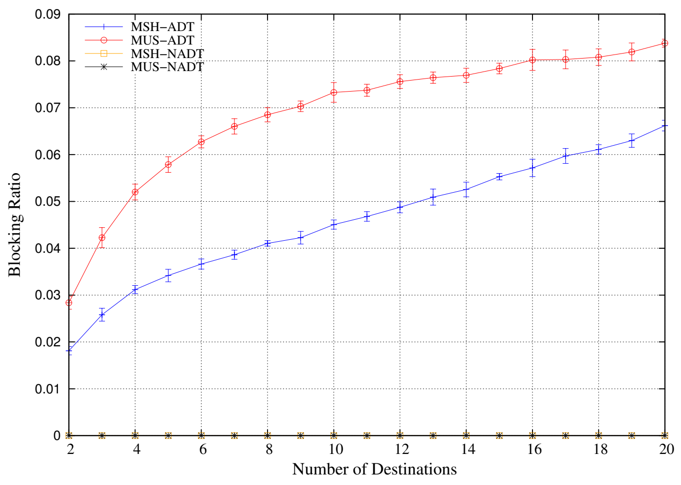

Fig. 10: Blocking ratio vs number of destinations, for the network with 50 nodes.

munications Magazine, vol. 2, no. 37, pp. 67–73, 1999.

[2] B. Mukherjee, *Optical Communication Networks*. New York, NY, USA: Mc-Graw-Hill, 1997.

[3] X. Zhang, J. Wei, and C. Qiao, "Constrained multicast routing in WDM networks with sparse light splitting," *IEEE/OSA Journal of Lightwave Technology*, vol. 18, no. 12, pp. 1917–1927, 2000.

[4] W. Tseng and S. Kuo, "All-optical multicasting on wavelength-routed WDM networks with partial replication," *International Conference on Information Networking (ICOIN)*, 2001.

[5] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

[6] L. L. X. Wang, S. Wang and S. Xu, "Achieving shared multicast protection in WDM networks with sparse-light-splitting constraint," *OSA Journal of Optical Networking*, vol. 7, no. 1, pp. 1–14, 2008.

[7] L. L. X. Wang, S. Wang and S. Xu, "Protecting light forest in survivable WDM mesh networks with sparse light splitting," *Int. J. Electron. Commun.*, vol. 63, no. 12, pp. 1043–1053, 2009.

[8] N. Sreenath and T. Prasad, "Protecting multicast sessions from link and node failures in sparse-splitting WDM networks," *International Workshop on Distributed Computing (IWDC)*, 2005.

[9] S. W. X. Wang and L. Li, "Provisioning of survivable multicast sessions in sparse light splitting WDM networks," *International Conference on Communications (ICC)*, 2008.

[10] B. C. A. Frikha, S. Lahoud and M. Molnar, "Reliable multicast sessions provisioning in sparse light-splitting DWDM networks using p-cycles," *International Communications Quality and Reliability Workshop (CQR)*, 2012.

[11] T. E. Stern, G. Ellinas, and K. Bala, *Multiwavelength Optical Networks: Architectures, Design, and Control*. New York, NY, USA: 2nd ed., Cambridge University Press, 2008.

[12] S. Cho, T. Lee, M. Chung, and H. Choo, "Minimum cost multicast routing based on high utilization MC nodes suited to sparse-splitting optical networks," *International Conference on Computational Science and Its Applications (ICCSA)*, 2006.

[13] C. Constantinou and G. Ellinas, "Survivable multicast routing in mixed-graph sparse-splitting optical networks," in *5th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 68–73, Sept. 2013.

[14] N. Singhal, L. Sahasrabuddhe, and B. Mukherjee, "Provisioning of survivable multicast sessions against single link failures in optical WDM mesh networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 21, pp. 2587–2594, Nov. 2003.

[15] C. Hsieh and W. Liao, "All-optical multicast routing in sparse splitting WDM networks," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 6, pp. 51–62, 2007.

[16] C. K. Constantinou, K. Manousakis, and G. Ellinas, "Multicast routing algorithms for sparse splitting optical networks," *Computer Communications*, vol. 77, pp. 100–113, March 2016.

[17] D. Xu, Y. Xiong, C. Qiao, and G. Li, "Trap avoidance and protection schemes in networks with shared risk link groups," *IEEE/OSA Journal of Lightwave Technology*, vol. 21, pp. 2683–2693, Nov. 2003.

[18] M. Doar and I. Leslie, "How bad is naive multicast routing?," in *Proc. IEEE INFOCOM*, pp. 82–89, 1993.

[19] S. Wang, "Allocation of light splitters in all-optical WDM networks with sparse light splitting capabilities," *Telecommunication Systems,*, vol. 52, no. 1, pp. 261–270, 2013.