Dedicated Protection of Multicast Sessions in Mixed-Graph Sparse-Splitting Optical Networks*

Teresa Gomes*†, Luís Raposo*, Georgios Ellinas‡

*Department of Electrical and Computer Engineering, University of Coimbra

Pólo 2, Rua Sílvio Lima 3030-290 Coimbra, Portugal

†INESC Coimbra, Rua Sílvio Lima, 3030-290 Coimbra, Portugal

‡KIOS Research Center and Department of Electrical and Computer Engineering,

University of Cyprus, Nicosia 1678, Cyprus

teresa@deec.uc.pt, lm.raposo91@gmail.com, gellinas@ucy.ac.cy

Abstract

This work addresses the problem of dedicated protection of multicast sessions in mixed-graph optical networks, where only a fraction of the nodes have optical splitting capabilities. A novel multicast routing algorithm for sparse splitting optical networks (the Modified Steiner Tree Heuristic (MSTH)) is initially presented and is subsequently utilized (together with two existing heuristics (MUS and MSH)) by an effective scheme for the calculation of a pair of disjoint trees. The key idea of this New Arc-Disjoint Trees (NADT) protection 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. Performance results demonstrate that the proposed NADT protection technique clearly outperforms the conventional Arc-Disjoint Trees (ADT) approach in terms of blocking ratio, while incurring only a negligible increase of the average cost of the derived pair of arc-disjoint trees. Furthermore, it is shown that the newly proposed algorithm, MSTH-NADT, is the one having the best performance in terms of cost and blocking, with MSH-NADT having similar, albeit slightly worse, performance. However, as MSH-NADT requires much less CPU time compared to MSTH-NADT, MSH-NADT can be considered the best compromise technique.

Keywords: survivability, multicasting; optical networks; sparse splitting; mixed-graph; heuristics

*Teresa Gomes acknowledges financial support by Fundação para a Ciência e a Tecnologia (FCT) under project grant UID/MULTI/00308/2013.

1 Introduction

Most connections carried over an optical mesh network have been and are still currently unicast connections, either high-bandwidth point-to-point connections for enterprise customers, or point-to-point trunks connecting a service provider's other networks (e.g., Ethernet). However, nowadays, as telecommunications service providers compete with traditional cable providers for the delivery of information (voice, video, etc.) over fiber-to-the-home infrastructures, they are expected to be able to support heterogeneous traffic demands, including high-bandwidth point-to-multipoint connections. As an example, the service providers receive their different programs via different avenues (content producers, cable TV channels, etc.) in a few specific locations, and aggregate them before distributing the information to their end-customers. In order to carry this aggregated content to different local distribution points (e.g., within cities), or to replace the pure broadcast capabilities of satellite networks, multicast connections would certainly seem appropriate. In addition, for the use of applications such as telepresence, video training, e-learning, and on-line teaching, as well as telemedicine and remote medical diagnosis, multicast connectivity appears to be the best solution to transport such applications.

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. These multicast applications leverage the use of common information-carrying

partial paths between the source of the information and the destinations to create a *multicast tree* that reaches all destinations, at a lower cost than an equivalent number of distinct unicast connections to these destinations.

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 [27]. 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 [19,27]. The remaining MI nodes of the network may be *Drop-and-Continue* (DaC) [45] or *Drop-or-Continue* (DoC) [34]. 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 [10]. For this problem, relevant polynomial-time multicast routing heuristics that give approximate solutions are used in practice and can be found in [9,31,37–39] 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. It is important to note here that even though failures cannot be avoided, quick detection, identification, and recovery of faults are crucial aspects in the successful deployment of telecommunication networks. A network fault that goes unattended for a long period of time can cause both tangible and intangible losses for the company that provides the service, as well as for its clients. Thus, the current trend is to try to achieve networks that are virtually uninterruptible. Currently, carriers are bound to service-level agreements (SLAs) with their customers, guaranteeing that the customer will be provided with services with a prescribed service availability (e.g., 99.999% availability - equivalent to less than 5 minutes of down time per year), with financial penalties if the availability SLA is not met. Thus, the need for fast failure recovery techniques is mainly due to the strict availability service requirements offered by the service providers to the customers (and quantified via the SLAs) in backbone mesh optical networks.

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. In essence, when protection techniques are used a pre-planned system is implemented, where a protection path is pre-computed for each potential failure (before the failure occurs) and the path uses pre-assigned resources for failure recovery. The underlying assumption in survivable networks is that enough redundancy is present in the network in the form of idle carrier facilities or spare capacity in currently used carrier facilities, to ensure recovery from failures.

Most commonly, network survivability is provided for single-link failures, since these are the predominant types of failures in optical networks, mainly due to the fact that they are a cable-based technology and the infrastructure is co-located with other utilities. Thus, fiber-cuts usually occur during construction work being done for the maintenance of other infrastructures (such as the water distribution network, the gas network, and the power distribution network), called the "backhoe" effect, or due to human error.

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 information flow [32].

Note that a bidirectional link is comprised of two arcs, one in each direction. For the case of a single-link failure, two Link-Disjoint Trees (LDT), or two Arc-Disjoint Trees (ADT) can be calculated to ensure the survivability of the network. Both of these protection techniques can protect the network from any single-link failure. However, the former is omitted in this work, since it needs more network resources compared to the latter. As it will be shown later on, in the case that a link fails (i.e., both arcs of it fail), if an arc-disjoint protection tree is found, the information can be sent to the destinations of the multicast tree using parts of the primary and the secondary trees. Therefore, although the two trees are just arc- and not link-disjoint, the network can survive from link failures as well, while requiring fewer resources compared to

the LDT technique.

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 the 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 from the primary one. The reader should note that an optimal solution to the problem (utilizing an ILP formulation) has not been considered in this work, as the networks considered are large (backbone networks of up to 100 nodes and several hundred links) and the run time of an ILP would be prohibitive. Such an ILP is left as future work, as noted in Section 8.

The remainder of the paper is organized as follows. Section 2 presents the state of the art for multicast routing and multicast protection in sparse splitting optical networks, while Section 3 presents a brief description of the sparse-splitting multicast routing heuristics that are combined with existing and proposed protection schemes. A new multicast routing heuristic is described in Section 4, while the computational complexity of the proposed routing algorithm and the existing ones are discussed in Section 5. The conventional and newly proposed protection techniques are discussed in Section 6, while Section 7 presents the evaluation of their performance. Finally, in Section 8 the conclusions of the paper are presented, as well as ongoing future work.

2 Multicast Routing and Protection in Sparse Splitting Optical Networks

There are a number of approaches in the literature on the problem of multicast routing in sparse-splitting optical networks. Some of the works that have received considerable attention are [2, 11, 45] where efficient solutions (namely the Member-Only (MO) heuristic [45], the On-Tree MC Node First (OTMCF) and Nearest MC Node First (NMCF) heuristics [11], and the Cost-Effective Multicasting Using Splitters (MUS) heuristic [2]) are identified for the problem of multicast routing in sparse splitting networks where the multicast-incapable nodes are either drop-or-continue (DoC) or drop-and-continue (DaC). Further, work in [3] investigates the problem of multicast routing in mixed-graph networks with sparse-splitting capabilities, demonstrating that the proposed MSH heuristic algorithm outperforms the most efficient existing approaches, in terms of average cost and blocking ratio of the arriving multicast requests.

Other multicast routing algorithms are also proposed that aim at minimizing the size and delay of the multicast trees [14,46], as well as the network resources (number of wavelengths required) [13,22], or at satisfying the underlying optical network constraints (e.g., power constraints, physical layer constraints) [1, 23, 35]. Work that appears in [25] also investigates the problem of multicast routing in conjunction with sub-wavelength traffic grooming in WDM optical networks.

Apart from multicast routing techniques, significant work has taken place on the protection of multicast connections against failures (fiber link and/or switching node failures). The works [8, 12, 30] concentrate on dedicated protection techniques, e.g., link- and arc-based protection techniques, essentially using some multicast routing techniques to obtain primary and secondary trees that are link- or arc-disjoint (also the technique used in this work). Other protection techniques are utilized in [26, 29], where some sharing of resources exists. In [15,16,18,28] the network is divided into segments and once the segments are identified, then each segment can be viewed as a separate tree and protected by any algorithm applicable to tree protection (segment-based protection techniques).

In essence, dedicated protection offers fast recovery and higher availability of services, as it does not require any set up along the backup path and there is no sharing of backup capacity, while tree sharing in multicast protection (either self-sharing (allowing sharing with other backup paths on the backup tree and with other edges

on the primary tree) or cross-sharing (allowing sharing of idle-backup edges from several multicast sessions)) saves on resources. Further, segment-based protection tries to minimize recovery time while at the same time trying to minimize the need for extra (redundant) resources.

Other constraints can also be imposed during the design of the protection techniques, such as the quality-of-transmission (QoT) and the availability of network resources (e.g., equipment cost, wavelength resources, etc.). For example, work in [20, 21] investigated the problem of protecting multicast sessions in optical networks, taking into account physical layer impairments, while work in [42] investigated the problem of sub-wavelength level protection for dynamic multicast traffic grooming, aiming at minimizing the network resources allocated for the protection of the traffic requests.

The same aforementioned types of protection techniques have also been investigated for multicast protection in sparse-splitting optical networks; however, a much smaller body of work exists for this type of network. For example, in [31] link- and node-disjoint algorithms are presented for finding the primary and secondary trees in networks with sparse splitting and sparse wavelength conversion, where some nodes have both splitting capability and wavelength conversion capabilities, whereas some other nodes have only splitting capability. These algorithms are subsequently compared in terms of the amount of bandwidth (number of wavelength channels) required for both the primary and backup trees.

Examples of works on shared protection include [37–39]. These works propose a shared protection algorithm, namely the shared sparse-splitting constrained multicast protection (SSSMP) algorithm, that allows for self-sharing (i.e., the backup paths can share wavelengths with the primary tree), as well as spare capacity sharing within different backup paths, in order to save on wavelength resources. In order to achieve wavelength sharing, a two-layered auxiliary graph model is initially developed (comprised of the physical layer that is mapping the network topology and the sharing layer that represents the primary wavelength channels that can be shared by backup paths) and subsequently utilized to obtain the backup tree that minimizes wavelength resource utilization.

Further, work in [17] investigates segment-based protection techniques against single-link failures in sparse-splitting optical networks where all the nodes have wavelength conversion capabilities. The problem is initially formulated as an ILP, followed by a heuristic algorithm called ASSP (adaptive shared segment protecting multicast tree) that initially establishes the multicast tree and subsequently constructs protection paths for the tree. A novelty of the ASSP technique is that it does not identify the segments a priori; this is done during the implementation of the algorithm, based on the multicast tree and current spare network resources. This technique is shown to perform better than link-disjoint trees (LDT) and shared disjoint paths (SDP) techniques, both in terms of blocking probability as well as cost of resources. Finally, work in [9] develops techniques for node and link protection for dynamic multicast connections in sparse-splitting networks (with DaC MI nodes) with no wavelength conversion capabilities, utilizing a p-cycle approach (called NPCC-SSC). Performance results for this technique demonstrate that it outperforms existing approaches (ESHT [44] and ESHN [43]) that also use the p-cycle approach and account for combined node and link failure recovery, in terms of blocking probability, resource utilization, and computational time.

3 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) [2] and Mixed-graph Sparse-splitting Heuristic (MSH) [3], 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.

4 A Modified Steiner Tree heuristic for Multicast Routing (MSTH)

The performance of the Steiner tree heuristic proposed in [33] is known to depend on the node selected as starting node [40].

As the source node of a multicast session must always be the starting node, here it is proposed that we should select as second node, to be added to the Steiner tree, each and every one of the destination nodes. This will allow us to explore some of Steiner tree dependence on the order of added nodes.

The next terminal node to be added to the tree would be the one at minimal distance to the nodes already on the tree (as in [33]). Because we are dealing with a tree for a multicast session, the minimal distance calculation is the cost of the path that connects the node to the current tree. If the path starts at the source node then its distance is the cost of a shortest path from the source of the multicast session to the candidate node to be added. But if the tree contains MC nodes, then the path may originate from a MC node already on the tree (including the source) from which the candidate node is at minimal distance. Note that the final solution is in fact a forest of light-trees.

Hence, all nodes, after the second node, are added according to their minimum distance to the present tree. We will designate this variant of the Steiner tree algorithm in [33] as Multicast routed Steiner Tree Heuristic with fixed second Terminal node (MSTHt), and is formalized in Algorithm 1.

Algorithm Modified Steiner Tree Heuristic (MSTH) will execute MSTHt once for each destination node and select the multicast tree with the minimum cost among the set of obtained trees – see Algorithm 2.

The following notation is required for presenting Algorithm MSTHt and MSTH.

- $G = (V, A)$ – 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;
- $r = (s, D)$ – multicast session with source node s, and destination node set D;
- T – primary tree;
- T_{MC} – set of MC nodes belonging to T (including s);
- p_{xy} – sequence 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;
- c_T – cost of tree T;
- $\mathcal{MC}$ – set of all MC nodes (including s)

Algorithm 1: MSTHt: variant of algorithm [33], with two nodes in sequence s followed by z

Input: G, r, z ; // $z \in D$ is the second node
Output: T, c_T

1 **begin**
2 $T \leftarrow (\{s\}, \emptyset)$; // Initial graph of T
3 Add p_{sz} to T ; // second node added to T
4 $D' \leftarrow D \setminus \{z\}$;
5 $c_T \leftarrow c_{sz}$; // initial cost of T
6 **while** $D' \neq \emptyset$ **do**
7 $c_{xy} \leftarrow \min_{i \in T_{MC}, j \in D'} c_{ij}$; // min dist. to T
8 Add path p_{xy} to T ;
9 $D' \leftarrow D' \setminus \{y\}$;
10 $c_T \leftarrow c_T + c_{xy}$; // updates cost of T
11 **return** (T, c_T) ; // End of execution

5 Computational Complexity of MSH, MUS, and MSTH

In this section the computational complexity of the existing and proposed multicast routing techniques in sparse-splitting optical networks is presented for comparison purposes.

The Steiner tree algorithm in [33] (designated as Minimum Path Heuristic (MPH) in [24]) runs in $\mathcal{O}(n^2 k)$ time [24, 33], with $n = |V|$ and k the number of nodes

Algorithm 2: MSTH: iterative use of Algorithm 1

Input: G, r
Output: T

```
Let z : z ∈ D ;                          // z is any node from D
(T, c_T) ← MSTHt(G, r, z) ;              // Initial T
D' ← D \ {z} ;
for each node d_i ∈ D' do
    (T', c_T') ← MSTHt(G, r, d_i) ;      // new tree
    if c'_T < c_T then
        (T, c_T) ← (T', c'_t) ;          // updates (T, c_T)
return (T, c_T) ;                        // End of execution
```

to be connected. In the present problem the only branching nodes are in the set $\mathcal{MC}$. Hence one can calculate the trees of shortest paths, rooted in the elements of $\mathcal{MC}$, and store them for later use. This step has complexity $\mathcal{O}\left(n^2|\mathcal{MC}|\right)$.

Complexity of the MUS Heuristic: The MUS heuristic has two phases. The first phase consists in solving a rooted Steiner tree problem, with destination nodes in set D_{MC}, where D_{MC} is the set of destination nodes that are also MC; hence the creation of this MC tree has complexity $\mathcal{O}\left(n^2(|D_{MC}|+1)\right)$ [33], with the assumption that Dijkstra's algorithm [6] has complexity of order $\mathcal{O}(n^2)$. In the second phase, the shortest path is found between each remaining destination (that is now MI) and the nodes on the MC-tree, with the shortest one being selected and added to the tree. This procedure ends when all remaining destinations are now connected to the tree (procedure presented in [2]). Therefore the complexity of this phase is $\mathcal{O}(n^2|D\backslash D_{MC}|)$. Thus the complexity of this implementation of MUS will be of order $\mathcal{O}\left(n^2(|D|+1)\right)$. Alternatively, if the first step of MUS is to calculate the trees of shortest paths, rooted in the elements of $\mathcal{MC}$, then the complexity of this implementation of MUS will be of order $\mathcal{O}(|\mathcal{MC}|(n^2+|D|^2))$, since phases (1) and (2) will only have to select among the previously calculated paths the ones to be included in the solution.

Note that, considering n and $|\mathcal{MC}|$ are fixed, if n is sufficiently larger than $|\mathcal{MC}|$, then $|\mathcal{MC}|(n^2+|D|^2)$ is smaller than $n^2(|D|+1)$ for $|D| \geq |\mathcal{MC}|$.

Complexity of the MSH Heuristic: Suppose that the network consists of n nodes and m links and the multicast request has $|D|$ destinations. The MSH heuristic can first calculate all candidate shortest paths that can be part of the tree: from the source to the nodes in D, and from every MC node to the nodes in D; hence it applies Dijkstra's algorithm $|\mathcal{MC}|$ times. The complexity of this initial step is: $\mathcal{O}(n^2|\mathcal{MC}|)$. The remainder of the algorithm is just a repeated selection of which path, among $\mathcal{O}(|\mathcal{MC}|\cdot|D|))$ different paths, to add to the tree: this is done $\mathcal{O}(|D|^2)$ times. Hence this part of the algorithm has complexity $\mathcal{O}(|D|^3\cdot|\mathcal{MC}|)$. Therefore, the complexity of MSH is of order $\mathcal{O}\left((n^2+|D|^3)|\mathcal{MC}|\right)$.

Complexity of the MSTH Heuristic: The auxiliary heuristic MSTHt, a variant of MPH with fixed second node, can have a complexity identical to MPH: $\mathcal{O}(n^2(|D|+1))$. MSTH executes MSTHt once for each destination node and selects the multicast tree with the minimum cost among the set of obtained trees, hence its complexity is $\mathcal{O}(n^2|D|^2)$. Alternatively, with MSTHt starting by calculating the trees of shortest paths rooted in the nodes of $\mathcal{MC}$, the complexity of this implementation of MSTH will be $\mathcal{O}(|D|\cdot|\mathcal{MC}|(n^2+|D|^2))$.

The reader should note that the complexities calculated above assumed Dijkstra's algorithm with complexity of order $O(n^2)$. In the case of a faster implementation of Dijkstra's algorithm (e.g., complexity of $O(m+n\log n)$ if implemented with Fibonacci heaps [5]) the complexity of all aforementioned heuristics will be adjusted accordingly.

In Section 7.2 the CPU times in Figs. 10 and 14 correspond to implementations where the first step was the calculation of the trees of shortest paths, rooted in each element of $\mathcal{MC}$.

6 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 (in fact the sum of the arcs of the light-trees that make the multicast tree). Similarly, the cost of a path on the tree is given by the sum of the weights of the arcs of the path.

6.1 Arc-Disjoint Trees Protection Scheme (Existing)

The *Arc-Disjoint Trees* (ADT) protection scheme was initially presented in [30], 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 secondary tree in the remaining graph using H.

Hereafter, the acronym $H\text{-}ADT$ 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 [3] 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 [3] 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) [11], as well as MUS [2] heuristics. In each case, the results were derived combining each multicast routing heuristic with ADT for obtaining a pair of arc-disjoint multicast trees.

As MSTH calculates several trees, each with a different second node (element of the set of destination nodes) its combination with the ADT procedure was implemented as follows. The option MSTH-ADT corresponds to calculate, for each candidate tree, a disjoint tree, starting with the same second node used to calculate the corresponding primary tree – recall that the disjoint tree is calculated in the network without the arcs of the primary candidate tree. Then the tree pair of minimum total cost is selected as the solution. Hence this corresponds, for each destination node, to using MSTHt to calculate the primary tree and corresponding secondary tree, using the same second node in both trees, and then selecting, among the resulting set of trees, the one of minimum total cost.

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). 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 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 (secondary) 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.

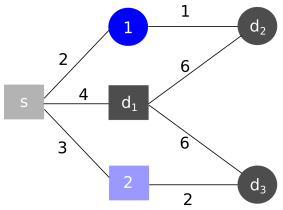

Figure 1: Example network.

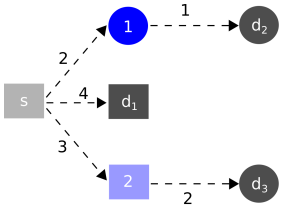

Figure 2: Tree obtained by MSH (cost=12).

6.2 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 the *New Arc-Disjoint Trees* (NADT) protection scheme, and its procedure is described in Algorithm 3. In the pseudo-code the following additional notation is used:

- T_{prot} – secondary (protection) tree;
- $T_{prot_{MC}}$ – 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;

Recall that T is the primary tree, $r = (s, D)$ is the multicast session with source node s and destination node set D, and that p_{xy} and c_{xy} are the shortest path that connects nodes x and y and its cost, respectively.

Algorithm 3: 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 Add back to G the arcs ofA_T
21 **else**
22 Add back to G the arcs in A_{exc};
23 $T_{prot} \leftarrow \emptyset$; `// No secondary tree`
24 $H(G, s, D, T)$; `// Only primary tree`
25 **return** T,T_{prot} ; `// End of execution`
26 **return** T,T_{prot} ; `// End of execution`

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 3 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 3). This new multicast 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 3). 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 3), 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, 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 [41].

The protection scheme, described by Algorithm 3, 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 3). In the latter case, it is considered that the multicast session cannot be protected and only the primary tree may be derived (lines 22-25 of Algorithm 3).

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 3). That is, when the cost of adding a minimum-cost path containing a destination is greater than β. In 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 3). 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 3). 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 [3]. This will be confirmed by the results in Section 7.

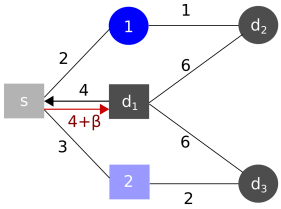

Figure 3: Modified network graph for the calculation of the primary tree with excluded arc (its weight is increased by β).

6.3 Example of NADT Protection Scheme with MSTH

In algorithm MSTH-NADT, every time a node is added to the primary tree, Algorithm 3 is used to verify if a backup tree exists. The set of terminal nodes of the backup tree are the ones already on the primary tree, and the calculation of the backup tree starts with the same node, after

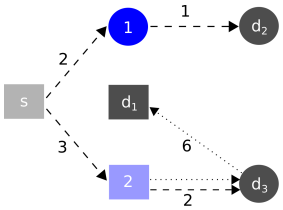

Figure 4: Primary tree obtained by MSH in the new network graph (cost=16).

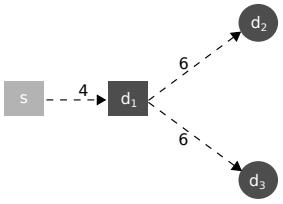

Figure 5: Secondary tree obtained by MSH, MUS and MSTHd after the calculation of the primary tree in the new network graph (cost=16).

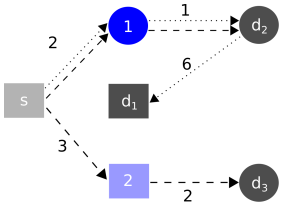

Figure 6: Primary tree obtained by MUS in the new network graph (cost=17).

the source node, used to calculate the corresponding primary tree under construction; from that point onwards MSHT-NADT uses MSTHt to verify if a secondary tree can be obtained.

Using the same network as in the previous examples, we will get the following solutions depending on the second node (the first node added to the tree after the source node):

if d_1 is the second node then the primary tree will coincide with the one depicted in Fig. 6, and the secondary tree will be identical to the one shown in Fig. 5.

if d_2 or d_3 is the second node then the primary tree will coincide with the one depicted in Fig. 4, and the

secondary tree will be identical to the one shown in Fig. 5.

7 Performance Evaluation

In this study we consider 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 DoC.

To study the performance of both protection schemes, combined with existing multicast routing heuristics (MSH and MUS) and the proposed heuristic MSTH, 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 to find a pair of arc-disjoint trees for a session in a given network, without being constrained by capacity.

Hence network links have unlimited capacity, and the blocking probability derives from the inability of the algorithm 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.

7.1 Simulation Set Up

Two undirected-graph networks were randomly created with the Doar-Leslie model [7] 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 are 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, five nodes were selected to be MC; these were chosen using the *kmaxD* method as described in [36] (i.e., the MC nodes were placed at the nodes that have the largest degree). Note that, similar to [3], 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 one 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 standard deviation of the ten samples around that mean is shown as error bars (barely visible in some figures, as for example in Fig. 7 or Fig. 11).

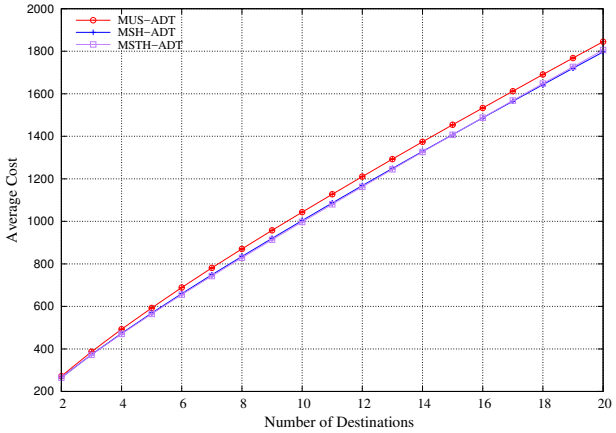

Figure 7: Average cost of the arc-disjoint trees vs. number of destinations, for the network with 40 nodes, using ADT.

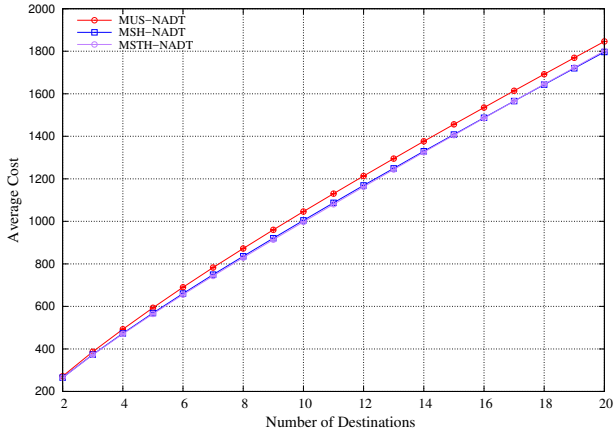

Figure 8: Average cost of the arc-disjoint trees vs. number of destinations, for the network with 40 nodes, using NADT.

7.2 Results for Multicast Routing with Protection

The results of the simulations, for the network with 40 nodes, are given in Figs. 7-10 while Figs. 11-14 illustrate the results for the 50-node network.

MUS-ADT is the heuristic with higher multicast tree cost, while MSTH-ADT and MSH-ADT present similar costs, with a slight advantage for MSTH-ADT – see Fig. 7 and Fig. 11. However, the blocking ratio of MSTH-ADT is significantly smaller than the blocking ratio of both MUS-ADT and MSH-ADT, especially for $|D| \geq 10$, in both networks (see Fig. 9 and Fig. 13).

In terms of blocking ratio, the NADT scheme outperforms the ADT scheme for the three 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 were rejected (less than 0.001% of all the multicast requests for both MUS-NADT and MSH-NADT, although not visible in Fig. 13, due to the non existence of a secondary tree. MSTH-NADT presented zero blocking also in the case of the 50-node network. The better performance of NADT with respect to ADT results from the fact that the proposed scheme, during the derivation of the primary tree, takes into account that a secondary tree, arc-disjoint from 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.

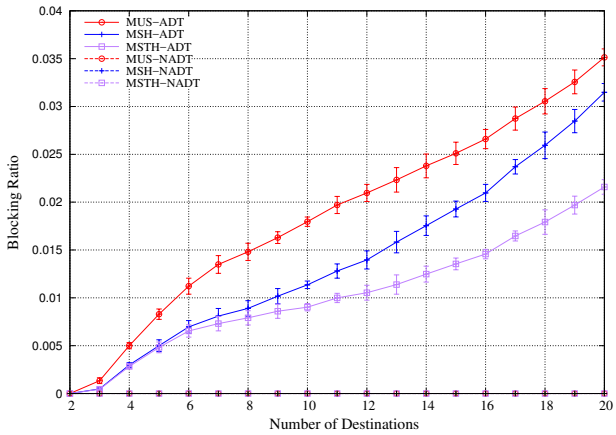

Figure 9: Blocking ratio vs. number of destinations, for the network with 40 nodes.

As for the results obtained for the average cost of the pair of arc-disjoint trees, the NADT scheme presents very similar cost to the ADT scheme, as can be seen in Fig. 7 versus Fig. 8 for the 40-node network and in Fig 11 versus Fig. 12 for the 50-node network.

There is a very slight increase in cost in the case of MUS-NADT with respect to MUS-ADT, 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-

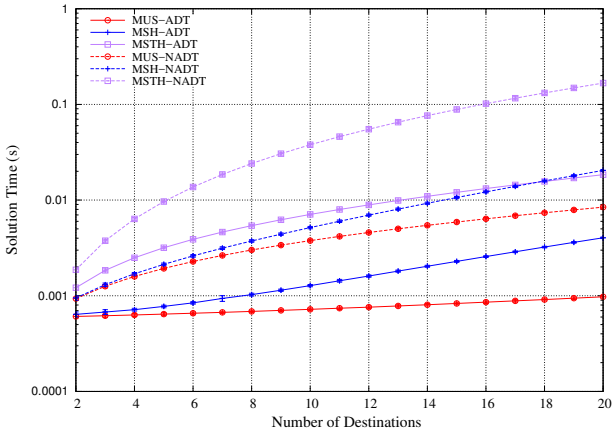

Figure 10: CPU time vs. number of destinations, for the network with 40 nodes.

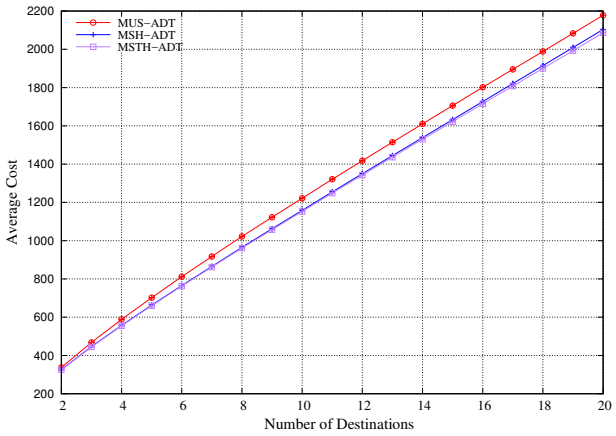

Figure 11: Average cost of the arc-disjoint trees vs. number of destinations, for the network with 50 nodes, using ADT.

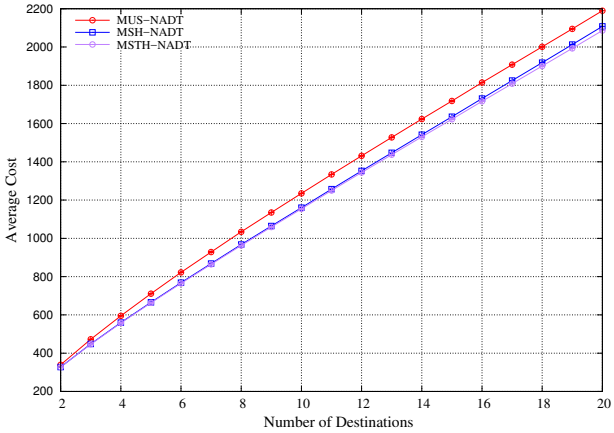

Figure 12: Average cost of the arc-disjoint trees vs. number of destinations, for the network with 50 nodes, using NADT.

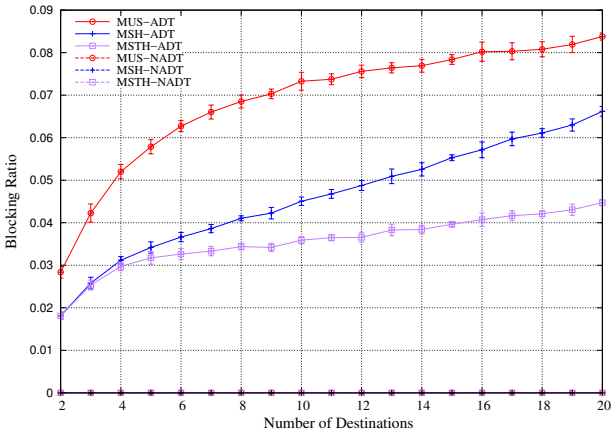

Figure 13: Blocking ratio vs. number of destinations, for the network with 50 nodes.

NADT and MUS-NADT follows the pattern in [3], where MSH-ADT has advantage over MUS-ADT. 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.

The sample average and standard deviation of the execution time of the simulations for the multicast trees with protection were also calculated. The execution times were obtained using a desktop with an Intel(R) Xeon(R) CPU, X5660 @ 2.80GHz, with 48 GB RAM. In Fig. 10 and Fig. 14, for the 40- and 50-node networks, respectively, the CPU time per number of destination nodes can be found. As would be expected, the ADT scheme is the one that requires less CPU time and MSTH-NADT is the one that requires more CPU time. The NADT

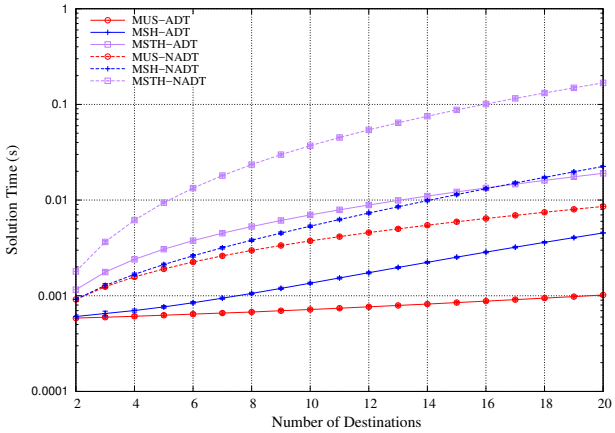

Figure 14: CPU time vs. number of destinations, for the network with 50 nodes.

scheme needs more CPU time than ADT, due to backup tree calculation per added node in the active tree, and also due to the active tree reconstruction every time the tree resulting from adding an additional destination cannot be protected. Hence the CPU time of the NADT scheme depends strongly on the complexity of the underlying heuristic.

Overall, algorithm MSTH-NADT is the one with globally better performance regarding cost and blocking, however it requires much more CPU time than MSH-NADT, so MSH-NADT may be considered as a compromise solution.

8 Conclusion

The problem of dedicated protection of multicast sessions in mixed-graph optical networks, where only a fraction of the nodes have optical splitting capabilities, was addressed. Conventional approaches have a two-step approach: firstly a primary tree of minimum cost is derived, and secondly an attempt is made to calculate a secondary tree that is arc-disjoint from the primary one. This can result in a false trap problem, when the minimum cost primary tree can not be protected, while a pair of disjoint trees does exist.

A new multicast routing scheme was initially proposed, which is clearly the most effective, of the three evaluated heuristics, when used with the ADT procedure.

Subsequently, a new algorithm for the calculation of a pair of disjoint trees, designated New Arc-Disjoint Trees Protection Scheme (NADT), was described. The basic idea of this algorithm is to keep checking if the tree under construction allows an arc-disjoint tree to be obtained. If no such tree can be calculated, a conflicting (shared) arc is identified for removal from the network before attempting once more to build a primary tree. This process is repeated until a tree with protection is obtained or it is no longer possible to calculate a primary tree.

The proposed protection scheme was combined with three heuristics for multicast routing (two existing and one new) in sparse-splitting networks. Simulation results 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 very slight increase of the average cost of the derived pair of trees. Further, it was shown that the newly proposed algorithm, MSTH-NADT, is the one having the best performance in terms of cost and blocking, with MSH-NADT having similar, albeit slightly worse, performance. However, as MSH-NADT requires much less CPU time compared to MSTH-NADT, MSH-NADT can be considered as the most suitable technique when all the performance metrics are taken into consideration.

Future work will focus on the application of the proposed NADT protection scheme in mixed-graph sparse-splitting networks with DaC nodes [4], as well as on the embedding of NADT into routing heuristics for networks with capacity constraints. Finally, an ILP will be formulated for obtaining the optimal solutions (for small networks of a few nodes), which will be subsequently compared with the solutions obtained by the heuristics, so as to ascertain the accuracy of the proposed techniques.

References

[1] M.A. Ait-Ouahmed and F. Zhou, A tabu search optimization for multicast provisioning in mixed-line-rate optical networks, Int Workshop Advances in Commun Networking: 20th EUNICE/IFIP EG 6.2, 6.6, *Lecture Notes in Computer Science*, Vol. 8846, Springer International Publishing, 2014, pp. 14–25.

[2] S.H. Cho, T.J. Lee, M.Y. Chung, and H. Choo, Minimum cost multicast routing based on high utilization MC nodes suited to sparse-splitting optical networks, Int Conference Comput Sci Its Appl - ICCSA, *Lecture Notes in Computer Science*, Vol. 3981, Springer Berlin Heidelberg, 2006, pp. 288–297.

[3] C. Constantinou and G. Ellinas, Survivable multicast

routing in mixed-graph sparse-splitting optical networks, 5th Int Congress Ultra Modern Telecommunications Control Syst Workshops (ICUMT), Sept. 2013, pp. 68–73.

[4] C.K. Constantinou, K. Manousakis, and G. Ellinas, Multicast routing algorithms for sparse splitting optical networks, Comput Commun 77 (March 2016), 100–113.

[5] T. Cormen, C. Stein, R. Rivest, and C. Leiserson, Introduction to algorithms, McGraw-Hill Higher Education, 2nd edition, 2001.

[6] E. Dijkstra, A note on two problems in connexion with graphs, Numer Mathematik 1 (1959), 269–271.

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

[8] A. Fei, J. Cui, M. Gerla, and D. Cavendish, A "dual-tree" scheme for fault-tolerant multicast, ICC 2001. IEEE Int Conference Communications. Conference Record (Cat. No.01CH37240), Vol. 3, 2001, pp. 690–694.

[9] A. Frikha, S. Lahoud, B. Cousin, and M. Molnar, Reliable multicast sessions provisioning in sparse light-splitting DWDM networks using p-cycles, 2012 IEEE Int Workshop Tech Committee Commun Qual Reliab (CQR), May 2012, pp. 1–6.

[10] M.R. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness, Freeman, 1979.

[11] C. Hsieh and W. Liao, All-optical multicast routing in sparse splitting WDM networks, IEEE J Selected Areas in Commun 25 (2007), 51–62.

[12] A. Khalil, A. Hadjiantonis, G. Ellinas, and M.A. Ali, Pre-planned multicast protection approaches in wdm mesh networks, 31st Eur Conference Optical Commun, ECOC 2005, Vol. 1, Sept 2005, pp. 25–26.

[13] D.D. Le, F. Zhou, and M. Molnar, Light-hierarchy for provisioning multiple multicast requests in sparse splitting WDM networks, 2015 Int Conference Comput, Networking Commun (ICNC), Feb 2015, pp. 847–852.

[14] D.D. Le, M. Molnár, and J. Palaysi, An improved multicast routing algorithm in sparse splitting WDM networks, 2013 Int Conference Comput, Manage Telecommunications (ComManTel), Jan 2013, pp. 99–104.

[15] L. Liao, L. Li, and S. Wang, Dynamic segment shared protection for multicast traffic in meshed wavelength-division-multiplexing optical networks, OSA J Optical Networking 5 (2006), 1084–1092.

[16] L. Liao, L. Li, and S. Wang, Multicast protection scheme in survivable WDM optical networks, Elsevier J Network Comput Appl 31 (2008), 303–316.

[17] C. Lu, G. Luo, S. Wang, and L. Li, A novel shared segment protection algorithm for multicast sessions in mesh WDM networks, ETRI J 28 (2006), 329–336.

[18] H. Luo, L. Li, and H. Yu, Algorithm for protecting light-trees in survivable mesh wavelength-division-multiplexing networks, OSA J Optical Networking 5 (2006), 1071 – 1083.

[19] B. Mukherjee, Optical communication networks, McGraw-Hill, 1997.

[20] T. Panayiotou, G. Ellinas, and N. Antoniades, Segment-based protection of multicast connections in metropolitan area optical networks with quality-of-transmission considerations, IEEE/OSA J Optical Commun Networking 4 (2012), 692–702.

[21] T. Panayiotou, G. Ellinas, and N. Antoniades, p-Cycle-based protection of multicast connections in metropolitan area optical networks with physical layer impairments constraints, Optical Switching Networking 19 (2016), 66–77.

[22] J. Park and J. Kim, QoS-driven multicast routing in sparse-splitting optical networks, Photonic Network Commun 25 (2013), 178–188.

[23] J. Park, H. Lim, and J. Kim, Virtual-node-based multicast routing and wavelength assignment in sparse-splitting optical networks, Photonic Network Commun 19 (2010), 182–191.

[24] J. Plesnik, Heuristics for the Steiner problem in graphs, Discr Appl Math 37/38 (1992), 451–463.

[25] A.K. Pradhan and T. De, Multicast traffic grooming in sparse splitting WDM mesh networks, Advances in Comput, Commun, Control: Third Int Conference, ICAC3, Springer Berlin Heidelberg, 2013, Communications in Computer and Information Science, pp. 444–458.

[26] T. Rahman and G. Ellinas, Protection of multicast sessions in WDM mesh optical networks, OFC/NFOEC Tech Digest. Optical Fiber Commun Conference, Vol. 2, March 2005, paper OTuK5, pp. 3.

[27] L. Sahasrabuddhe and B. Mukherjee, Light-trees: Optical multicasting for improved performance in wavelength-routed networks, IEEE Commun Magazine 2 (1999), 67–73.

[28] N. Singhal and B. Mukherjee, Algorithms for provisioning survivable multicast sessions against link failures in mesh networks, 5th Int Workshop Distrib Comput - IWDC, 2003, *Lecture Notes in Computer Science*, Vol. 2918, pp. 361–371.

[29] N. Singhal, C. Ou, and B. Mukherjee, Cross-sharing vs. self-sharing trees for protecting multicast sessions in mesh networks, Comput Networks 50 (2006), 200 – 206.

[30] N. Singhal, L. Sahasrabuddhe, and B. Mukherjee, Provisioning of survivable multicast sessions against single link failures in optical WDM mesh networks, IEEE/OSA J Lightwave Technology 21 (Nov. 2003), 2587–2594.

[31] N. Sreenath and T.S. Prasad, Protecting multicast sessions from link and node failures in sparse-splitting WDM networks, 7th Int Workshop Distrib Comput (IWDC), *Lecture Notes in Computer Science*, Vol. 3741, 2005, pp. 195–200.

[32] T.E. Stern, G. Ellinas, and K. Bala, Multiwavelength optical networks: Architectures, design, and control, 2nd ed., Cambridge University Press, 2008.

[33] H. Takahashi and A. Matsuyama, An approximate solution for the Steiner problem in graphs, Mathematica Japonica 24 (1980), 573–577.

[34] W.Y. Tseng and S.Y. Kuo, All-optical multicasting on wavelength-routed WDM networks with partial replication, Proc 15th Int Conference Informat Networking, 2001, pp. 813–818.

[35] J. Wang, J. Yuan, X. Zhou, and X. Yu, A novel multicast routing algorithm in sparse splitting WDM network with power attenuation constraint, Photonic Network Commun 19 (2010), 134–143.

[36] S. Wang, Allocation of light splitters in all-optical WDM networks with sparse light splitting capabilities, Telecommunication Syst 52 (2013), 261–270.

[37] X. Wang, S. Wang, and L. Li, Provisioning of survivable multicast sessions in sparse light splitting WDM networks, 2008 IEEE Int Conference Commun, May 2008, pp. 5286–5291.

[38] X. Wang, S. Wang, L. Li, and S. Xu, Achieving shared multicast protection in WDM networks with sparse-light-splitting constraint, J. Opt. Netw. 7 (Jan 2008), 1–14.

[39] X. Wang, S. Wang, L. Li, and S. Xu, Protecting light forest in survivable WDM mesh networks with sparse light splitting, Int. J. Electron. Commun. 63 (2009), 1043–1053.

[40] P. Winter and J. MacGregor Smith, Path-distance heuristics for the Steiner problem in undirected networks, Algorithmica 7 (1992), 309–327.

[41] D. Xu, Y. Xiong, C. Qiao, and G. Li, Trap avoidance and protection schemes in networks with shared risk link groups, IEEE/OSA J Lightwave Technology 21 (Nov. 2003), 2683–2693.

[42] X. Yu, G. Xiao, and T.H. Cheng, An efficient mechanism for dynamic survivable multicast traffic grooming, Optical Fiber Technology 23 (2015), 1–12.

[43] F. Zhang and W.D. Zhong, Performance evaluation of optical multicast protection approaches for combined node and link failure recovery, J Lightwave Technology 27 (Sept 2009), 4017–4025.

[44] F. Zhang and W.D. Zhong, ρp-Cycle based tree protection of optical multicast traffic for combined link and node failure recovery in WDM mesh networks, IEEE Commun Lett 13 (January 2009), 40–42.

[45] X. Zhang, J. Wei, and C. Qiao, Constrained multicast routing in WDM networks with sparse light splitting, IEEE/OSA J Lightwave Technology 18 (2000), 1917–1927.

[46] F. Zhou, M. Molnar, and B. Cousin, Distance priority based multicast routing in WDM networks considering sparse light splitting, 11th IEEE Singapore Int Conference Commun Syst, Nov 2008, pp. 709–714.