Calculation of Arc-Disjoint Trees in Mixed-Graph Arbitrary Mesh Optical Networks

Costas K. Constantinou  
Department of Electrical and Computer Engineering  
University of Cyprus  
Nicosia, 1678, Cyprus  
Email: constantinou.k.costas@ucy.ac.cy

Georgios Ellinas  
Department of Electrical and Computer Engineering  
University of Cyprus  
Nicosia, 1678, Cyprus  
Email: gellinas@ucy.ac.cy

Abstract—In the paper, a heuristic algorithm that deals with the problem of finding arc-disjoint trees in mixed-graph mesh optical networks is presented. A mixed graph is defined as one that has both unidirectional and bidirectional links between its nodes. The calculation of pairs of arc-disjoint trees is applied for provisioning survivable multicast requests. The development of algorithms that work efficiently for this kind of networks is important, since in practice the network is modeled as a mixed graph. Simulations show that for the case of mixed-graph networks, the proposed multicast protection algorithm is able to calculate arc-disjoint tree pairs with lower blocking ratio and average cost compared to the existing algorithms, especially for multicast sessions with destination sets that comprise of up to half of the network nodes.

I. INTRODUCTION

Over the last years, optical fiber has prevailed as the medium of choice for high-speed communications in backbone and more recently in metropolitan area networks. Its vast amount of bandwidth and the low bit error rates compared to copper-wire transmission systems [1] are the main reasons for this. Moreover, multicasting applications such as high-definition television, video conferencing, interactive distance learning, live auctions, distributed games, and video-on-demand are now becoming more feasible because of the high-capacity and increased capabilities provided by intelligent Wavelength Division Multiplexing (WDM) optical networks [2]. Multicasting is based on the calculation of light-trees, utilizing optical splitters in the network nodes [3]. The information through an optical fiber is sent using a specific wavelength for each multicast request. If no wavelength converters exist at the network nodes, the same wavelength must be used in all fibers [1]. In the current paper it is assumed that each network node has full wavelength conversion, therefore the information can be sent in different wavelengths through the light-tree fibers.

In most cases, there are more than one light-trees that can be used, and the one with the least cost among them is preferred. The calculation of these least-cost trees for multicasting is a fundamental problem of Graph Theory and Combinatorial Optimization, and it is called the “Steiner Minimal Tree” (SMT) problem. This is an NP-complete problem [4], therefore algorithms with polynomial-time complexity that find the optimal solution in every case do not exist. The heuristics that are used extensively in the literature are Pruned Prim Heuristic (PPH) [5] and Minimum Path Heuristic (MPH) [6]. Examples of research papers that utilize these algorithms are [7, 8, 9, 10]. Other recently developed algorithms are Steiner Node Heuristic (SNH) [11, 12] and Mixed Graph Minimum Path Heuristic (MG-MPH) [13]. The reasons that these algorithms are exploited are their adequate performance, their simplicity, and the fact that they can applied to mixed-graph networks as well.

Furthermore, since an optical fiber carries vast amount of information, in the case of a single fiber failure a great amount of information can potentially be lost. In multicasting, the information loss is even greater since some fibers carry information to multiple destinations (Figure 1). Therefore, it is very important to develop efficient multicast protection techniques. These techniques are created in order to provide network survivability under single-link failures, since this is the predominant form of network failure. One method to provide protection in multicasting applications is the calculation of two arc-disjoint trees. If the first (working or primary) tree fails to transmit the information due to a single link failure, the information is transmitted through the second (protection or secondary) tree. PPH, MPH SNH and MG-MPH techniques can be utilized for the calculation of this pair of trees, and the resulting techniques are called PPH-ADT, MPH-ADT, SNH-ADT and MG-MPH-ADT respectively (ADT stands for Arc Disjoint Trees). The first three techniques give satisfactory solutions for undirected graphs, but have low performance in mixed-graph networks. The last one was especially designed to perform well for both undirected and mixed graphs.

The development of new protection algorithms that are designed to work efficiently for mixed-graph networks, is very important; even if the network is designed to be undirected, when some multicast requests arrive and hold resources of the network, the resulting network graph is mixed, therefore the calculation of the pair of working and protection trees for the new requests, will be calculated on a mixed graph. Even in the case that the two trees are calculated on an undirected graph, the arcs of the working tree are removed from the graph for the calculation of the protection tree. Therefore, the latter is calculated on the resulting mixed graph.
The creation of multicast routing algorithms that are able to find light-trees consisting of a small number of arcs is crucial. When these algorithms are utilized for the calculation of arc-disjoint trees, the primary tree is calculated, it is removed from the graph, and the secondary one is calculated on the resulting graph. Therefore, if the utilized algorithm succeeds to find trees with less arcs, the average cost (which is calculated in terms of number of arcs of the primary and secondary tree) is decreased and the blocking ratio is decreased as well; if the primary tree uses a small number of arcs, the resulting graph after the removal of them consists of more arcs and this decreases the possibility that a secondary tree will not exist.

The remaining of the paper is organized as follows: In Section II, some necessary graph-theory definitions are given. The general arc-disjoint-tree protection technique is presented in Section III and the existing multicast protection algorithms are presented in Section IV. Subsequently, in Section V, the proposed multicast protection algorithm is given. All of the algorithms that are presented in the paper are evaluated through simulations and the results are presented in Section VI. Analysis of the results is also given in this section. Finally, conclusions and future work are given in Section VII.

II. DEFINITIONS

Before the description and evaluation of the existing and proposed multicast protection methods, it is essential to give some important graph theory definitions. These can be found in [14,15], as well as other sources.

A graph \( G(V,E) \) is a finite, nonempty set \( V \) together with a (possibly empty) set \( E \) (disjoint from \( V \)) of two-element subsets of (distinct) elements of \( V \). Each element of \( V \) is referred to as a vertex or node. The members of set \( E \) are called edges. A graph is considered connected if there is a path between every pair of its vertices. It is called simple graph if it has no self-loops or multiple edges having the same pair of end vertices. In the case that a real number is assigned as a weight for every edge, then the graph is weighted (\( c(i,j) \) denotes the cost of edge \((i,j)\)). Unidirectional edges are defined as arcs and bidirectional as links. Therefore, every link consists of a pair of opposite arcs (arcs with opposite orientation and equal weight). A graph is considered as undirected if all edges are links, directed if all edges are arcs and mixed if it has both links and arcs. An example is given in Figure 2; in the first graph all connections are bidirectional, therefore this graph is considered to be undirected, while in the second one the connections 2–3, 2–5, 7–5, and 8–7 are unidirectional (i.e., the opposite arc does not exist) and the rest are bidirectional. This graph is mixed.

III. GENERAL ARC-DISJOINT PROTECTION TECHNIQUE

The purpose of a large number of multicast protection schemes is the calculation of two disjoint paths from the source node to each destination node of the specific multicast request. If this is realizable, the multicast request will be serviced regardless of any single link failure in the network. One way to achieve this is by calculating two disjoint trees for every multicast request [7].

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. The former is omitted in this analysis, since it needs more network resources compared to the latter. A simple example where the efficiency of ADT method is shown, is presented in Figure 3. Here, the link \( d_1-d_2 \) is utilized by both the primary tree (arc \( d_1-d_2 \)) and the secondary tree (arc \( d_2-d_3 \)). In the case that this link fails (i.e., both arcs of it fail), the information can be sent to \( d_3 \) using the primary tree \( (s-d_3) \) and to \( d_2 \) using the secondary tree \( (s-d_2) \). Therefore, although the two trees are just arc- and not link-disjoint, the network can survive from link failures as well.

The general procedure for the calculation of two arc-disjoint trees is as follows:
The Steiner Node Heuristic (SNH) algorithm is a generalization of MPH that on the average performs better. It consists of the following steps:

1) Calculate Steiner Tree $T^*(V^*, E^*)$ using MPH and find its cost, called $C$.
2) For every node $v_j \in \{V - V^*\}$ do the following:
   (a) $\{Z'\} = \{Z\} + v_j$.
   (b) Calculate Steiner Tree $T_i(V_i, E_i)$ on $G(V, E)$ for set $\{Z'\}$, using MPH, and find its cost, $c_i$.
   (c) $\{Z'\} = \{Z\}$.
3) Find $v_j$ and $T_j: c_j = \min_{i} \{c_i\}$.
4) If $c_j \geq C$, the solution is $T^*(V^*, E^*)$, STOP.
   (a) If $c_j < C$, replace $\{Z\}$ with $\{Z\} + v_j$, $C$ with $c_j$ and $T^*(V^*, E^*)$ with $T_j(V_j, E_j)$.
   (b) If set $\{V - V^*\} \neq 0$, return to Step 2. Else STOP.

Explanation of SNH: The key function of the algorithm is the discovery of those nodes that the addition of them in set $Z$ gives a less-cost tree. In the first step, the Steiner Tree using MPH is calculated (called “TREE” (TR)). Then, in Step 2, for every node in graph but not in TREE, MPH is utilized again for the calculation of the corresponding tree, if the specific node was temporarily in set $Z$. After the comparison of these trees, the one with the least cost (called “New Tree” (NT)) is kept. If $\text{cost}_{NT} \geq \text{cost}_{TR}$, the algorithm terminates and tree TR is the final solution, whereas if $\text{cost}_{NT} < \text{cost}_{TR}$, tree TR is replaced with NT, the corresponding node is added permanently in set $Z$ and Step 2 is repeated. The algorithm terminates, with tree TR as the final solution, either if, after the procedure of Step 2, NT has more than or equal cost to TR, or if all nodes of the graph belong to TR. The following simple example shown in Figure 4 makes the algorithm more understandable and shows its enhanced cost performance.

While the MPH algorithm finds tree $\{S - d_i, S - d_j\}$ with cost 200, the SNH technique, with the addition of node $n$ to the destination set, is able to find a tree with less cost, i.e., tree $\{S - n, n - d_1, n - d_2\}$, with cost 180.

Mixed Graph Minimum Path Heuristic (MG-MPH) algorithm is a generalization of MPH that performs better in mixed graphs and has the same performance with the latter in undirected graphs. It consists of the following steps:

1) Tree $T_i$ consists only of the source $s$. Set $V_i = \{s\}$.
2) a) Determine a node $u$ in $\{D - V_i\}$, that is closest to $T_i$ (which is connected through a directed path originating from $T_i$ and ending at $u$). Construct a tree $T_{i+1}$ by adding the minimum cost path joining $u$ and $T_i$. Add the nodes of this path in set $\{V_i\}$. $i = i + 1$.
adding the minimum cost path joining \( u \) and \( T_i \). Add the nodes of this path in set \( \{V_1\} \).

b) Keep only the path from source to node \( u \) as the current tree and delete all other arcs.

c) Add again to the current tree all the destinations that were part of the tree before \( u \), using the MPH algorithm.

d) \( i = i + 1 \).

3) Repeat Step 2. STOP when all nodes in \( D \) are connected to the tree.

The algorithm functions as follows: After the addition of destination \( u \) (let \( U \) be the set of the nodes that are added to the tree for the connection of destination \( u \)), only the path from source \( s \) to \( u \) is kept as the current tree. All destinations that were part of the tree before the addition of \( u \) and do not belong to path \( s \rightarrow u \), are added again using the MPH algorithm. Therefore, they will be connected either through the path that were already connected, or through a path that originates from a node in set \( U \). The best path between the two will be selected.

The proposed algorithm is applied in Figure 5 (\( s \) is the source node and \( d_1, d_2 \) are the destinations). PPH, MPH and SNH would find the tree of step \( ii \) (\text{cost} = 7), while MG-MPH algorithm succeeds to find the minimum cost tree (step \( iv \), \text{cost} = 6).

The aforementioned algorithms are applied to the general arc-disjoint technique for provisioning survivable multicast requests. It is obvious that SNH and MG-MPH algorithms perform better compared to PPH and MPH. SNH is more efficient because it succeeds in finding the critical nodes, the addition of which in the destination set will give a Steiner tree of less cost. On the other hand, MG-MPH is a generalization of MPH that performs efficiently in mixed-graph networks. Therefore, in the current paper, a new multicast routing algorithm is proposed, that combines the advantages of both SNH and MG-MPH. It is in essence the mixed-graph generalization of SNH; it has the same performance with the latter for undirected graphs and improved performance for mixed ones. The new algorithm is called Mixed Graph Steiner Node Heuristic or MG-SNH and it is presented in the following section.

V. PROPOSED MULTICAST PROTECTION ALGORITHM (MG-SNH-ADT)

The Mixed Graph Steiner Node Heuristic (MG-SNH) consists of the following steps:

1) Calculate Steiner Tree \( T^*(V^*, E^*) \) using MG-MPH and find its cost, called \( C \).

2) For every node \( v_i \in \{V - V^*\} \) do the following:
   a) \{\( Z' \} = \{Z\} + v_i.
   b) Calculate Steiner Tree \( T_i(V_i, E_i) \) on \( G(V, E) \) for set \( \{Z'\} \), using MG-MPH, and find its cost, \( c_i \).
   c) \{\( Z' \} = \{Z\} \).

3) Find \( v_j \) and \( T_j : c_j = min_i \{c_i\} \).

4) If \( c_j \geq C \), the solution is \( T^*(V^*, E^*) \), STOP.
   a) If \( c_j < C \), replace \( \{Z\} \) with \{\( Z\} + v_j, \) \( C \) with \( c_j \) and \( T^*(V^*, E^*) \) with \( T_j(V_i, E_i) \).
   b) If set \( \{V - V^*\} \neq 0 \), return to Step 2. Else STOP.

The algorithm uses MG-MPH as its basis, and applies a recursive procedure to locate the nodes that must be added to the destination set to give a tree with less cost.

The MG-SNH-ADT multicast protection method then consists of the following steps:

1) Calculate the primary light-tree using the MG-SNH algorithm.

2) Remove the arcs of the primary tree from the network graph.

3) Calculate the secondary tree on the resulting graph using the MG-SNH algorithm.

VI. PERFORMANCE EVALUATION

The performance of the MG-SNH-ADT protection method is evaluated and compared with the existing relevant algorithms (MPH-ADT, SNH-ADT, MG-MPH-ADT). PPH-ADT is omitted since it performs much worse compared to the other three techniques [11, 12, 13]. The evaluation is performed via simulations on a mixed graph that was randomly created. The graph was created to be strongly connected. It has 40 nodes, 160 arcs (unidirectional connections) and 40 links (bidirectional connections). This graph should be simple, therefore all 200 edges connect different pairs of nodes. The Percentage of Directionality (PoD) (i.e., the ratio arcs/edges) was set to \( PoD = 160/200 = 0.8 = 80\% \). The simulation was repeated for various possible multicast group sizes (\( D \)), from \( D = 4 \) to \( D = 36 \), with a step equal to 4. The experiment was executed 5000 times for every multicast group size, while the source and destinations of the multicast connections were distributed uniformly across the network. 5000 pairs of arc-disjoint trees were calculated with each one of the existing and proposed multicast protection methods. If an arc-disjoint pair of trees was able to be calculated, the multicast request was considered as realizable. In the opposite case it was considered as blocked. The blocking ratio was calculated as the ratio of
blocked/total multicast requests. The cost criterion for the routing algorithms was the sum of the arcs that were used for the primary and secondary trees. The blocking ratio and the average cost (i.e., the sum of the arcs of the two disjoint trees) were calculated for each protection method. The results are presented in Figures 6-9. The cases of small multicast group size (up to 16 destinations) and large (20 destinations and more) were presented in different graphs for clarity purposes.

**Analysis of the results**

**Small multicast group size (size of destination set up to half the size of the network):** It is shown that the MG-SNH-ADT multicast protection method outperforms the rest in terms of blocking ratio (BP) and average cost, for small multicast sessions. The first reason is that this method is able to find the nodes that give light-trees with less arcs if they are added in the destination set. The second reason is that the proposed method was especially designed for mixed graphs, therefore it has enhanced performance in this category of networks. Since MG-SNH-ADT finds trees with less arcs, it is obvious that this minimizes the average cost (average cost is calculated in terms of number of arcs of the primary and secondary trees). It minimizes the blocking ratio as well; if the primary tree uses a small number of arcs, the resulting graph after the removal of them consists of more arcs and this decreases the possibility that a secondary tree will not exist.

**Large multicast group size:** For the case of large multicast sessions, MPH-ADT and SNH-ADT have the same blocking ratio. The blocking ratio of MG-MPH-ADT and MG-SNH-ADT is equal, much lower compared to the other
two techniques. The reason is that if the destination set has many network nodes, the possibility of the existence of non-destination nodes that will give trees with less arcs if added in the destination set, is negligible. Therefore, the only characteristic that can give better results is the mixed-graph specialization of an algorithm. The four methods also have practically the same average cost for large multicast sessions.

VII. CONCLUSIONS

A new multicast routing algorithm that performs efficiently for mixed-graph networks, called Mixed Graph Steiner Node Heuristic (MG-SNH) is presented in this paper. The proposed algorithm is applied for provisioning survivable multicast requests via the calculation of arc-disjoint tree pairs. The resulting technique is MG-SNH-ADT. It was specially designed to work efficiently for mixed graphs and simulation results verify this, showing that for this category of networks, MG-SNH-ADT outperforms the existing relevant methods (MPH-ADT, SNH-ADT, MG-MPH-ADT) for small multicast sessions (size of the destination set up to half the number of the network nodes), in terms of blocking ratio while having the same average cost.

Future work will concentrate on simulations on more networks with different size, density and topology, as well as on the comparison of the simulation results with the exact results that will be obtained using Integer Linear Programming for networks of small size. The case of dynamic multicast request arrivals will also be investigated. For the latter case the new method will be combined with cross-sharing techniques [16] for improved performance.

Acknowledgment: This work was co-funded by the European Regional Development Fund and the Republic of Cyprus through the Research Promotion Foundation (Project New Infrastructure/ Strategic/0308/26).

REFERENCES