Survivable Multicasting in Sparse-Splitting Optical Networks

Costas K. Constantinou
KIOS Research Center
University of Cyprus
Nicosia, 1678, Cyprus
Email: constantinou.k.costas@ucy.ac.cy

Konstantinos Manousakis
KIOS Research Center
University of Cyprus
Nicosia, 1678, Cyprus
Email: manousakis.konstantinos@ucy.ac.cy

Georgios Ellinas
KIOS Research Center and
ECE Department
University of Cyprus
Nicosia, 1678, Cyprus
Email: gellinas@ucy.ac.cy

***Abstract*—In this work, the problem of provisioning survivable multicast connections in optical networks is investigated, under the assumption that not all network nodes are multicast capable (MC). A MC capable node is a node that can forward the incoming signal to multiple output ports. An integer linear programming (ILP) formulation is presented, as well as a heuristic algorithm to address the problem. Simulations show that the proposed heuristic gives an average cost of the derived multicasting subgraphs that is very close to the optimal one obtained by the ILP.**

***Index Terms*—Survivable multicasting, protection, optical networks, sparse splitting.**

I. Introduction

Wavelength Division Multiplexing (WDM) optical networks enable different connections to be established concurrently through a common set of fibers. WDM networks provide high-capacity communication through lightpath establishment. Lightpaths are all-optical channels that may span multiple consecutive fibers. In the absence of wavelength converters in the network, a lightpath occupies the same wavelength on all links from the source to the destination node (*wavelength continuity constraint*). High-bandwidth demand 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 increased capabilities provided mainly by the intelligent optical cross-connects (OXCs) that are utilized in these kinds of WDM optical backbone networks [1].

All these aforementioned applications are feasible when multicast communication is used that is based on the calculation of light-trees instead of lightpaths. In order to set-up these light-trees in optical networks to provide multicast services from a single source to multiple destinations, the utilization of optical splitters in the network nodes is required [2]. An optical-splitter is a passive device that splits the input signal into multiple identical output signals [3]. The nodes that have splitting capability are called *Multicast-Capable* (MC) nodes. If not, they are called Multicast Incapable (MI). In the case that the optical splitter splits the incoming signal into n outputs ports, each one of these signals has $(1/n)^{th}$ of the input power. Thus, an optical network with a large number of multicast-capable nodes may cause the signal to experience a significant power loss, potentially limiting its reach. To combat this effect, a large number of optical amplifiers will be required in an all-optical network, further adding noise in the system and requiring a worst-case network engineering and design [4]. To limit the impact of optical splitters in the network, they can be placed at only some of the network nodes (multicast-capable (MC) nodes), resulting in a *sparse-splitting* network [2, 3]. The remaining multicast-incapable (MI) nodes of the network may be *Drop-and-Continue* (DaC) [5] or *Drop-or-Continue* (DoC) [6]. 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 networks where the MI nodes have the DaC capability.

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 the traffic to multiple destinations, have led to the necessity for the development of efficient multicast survivability (or "protection") techniques. Protection techniques entail the pre-computation of the secondary (protection) path prior to the fault occurrence. These techniques are developed in order to provide at a minimum network survivability under *single-link* failures, since this is the predominant form of network failure. The problem of multicast routing in sparse-splitting networks is NP-hard, since the NP-hard Steiner problem in graphs [7] is a special case of it. Therefore, the problem of multicast protection is NP-hard as well. For this problem, polynomial-time heuristics that give approximate solutions are used in practice [8]-[12].

In the current paper a new Integer Linear Programming (ILP) formulation, as well as a new heuristic are presented, for provisioning survivable multicast requests under single-link failures in sparse-splitting optical networks. The heuristic (called *Dual-route Sparse-splitting Heuristic (DSH)*), succeeds to give results close to the optimal ones, obtained by ILP, as simulations on the well-known USNET network have shown. The performance criterion was the average cost of the derived multicasting subgraphs. The remaining of the paper is organized as follows: Section II presents the formulation of the problem investigated. The proposed ILP formulation and the heuristic algorithm are presented in Sections III and

IV respectively. The performance comparison between the optimal solution obtained by ILP and the one derived by the heuristic is presented in Section V. Finally, in Section VI, the conclusions of the paper are presented, as well as ongoing future work.

II. Problem Formulation

The current work deals with the problem of survivable multicast routing in sparse-splitting optical networks and particularly, the survivability of the multicast requests under single link failure scenarios. The work focuses on the case of single multicast requests, rather than multiple dynamically arriving requests (i.e., requests that arrive in the network and stay for some time, holding resources of it). This case is investigated so as to ascertain the cost of the derived multicasting subgraphs rather than the blocking rate of the network.

Throughout the paper it is assumed that the network is not equipped with wavelength converters, therefore the desired multicasting subgraph must use only one wavelength, the same on each of its arcs. It is also assumed that the source is equipped with multiple transmitters and that the network nodes are MC or MI. Moreovere, MI nodes have the DaC capability. The network is equipped with two fibers on each *link*, with opposite orientation. Each one of the two fibers is called an *arc* throughout the paper. Since the purpose of the paper is the survivable routing of multicast requests, the derived subgraph in this case is not a *tree* as defined in graph theory, since a tree is defined as an acyclic connected graph, whereas the existence of cycles on the derived multicasting subgraph is permitted here due to the fact that two link-disjoint paths exist on the derived subgraph for each destination of the multicast request.

The network graph is notated as $G(V,E)$ (or G for simplicity), where V ($|V|=n$) and E ($|E|=m$) are the sets consisting of the network nodes and edges respectively. The source of the multicast request is notated as s, the destination set as $D=\{d_1,d_2,...,d_k\}$, and the desired subgraph as $SG(V_{SG},E_{SG})$ (or SG for simplicity) where V_{SG} and E_{SG} are the sets consisting of the subgraph nodes and edges respectively. The distance among a set of nodes A and a node v is defined as the cost of the shortest path with the minimum cost among all shortest paths between the nodes in A, and v. The corresponding path is notated as $SP[A][v]$ and its cost (i.e., its distance as defined above) is notated as $|SP[A][v]|$. The $outdegree[v]$ of a node v is defined as the number of arcs that originate from v, on SG.

The output of the ILP as well as of the proposed heuristic is a directed subgraph rooted at s which includes two directed, link-disjoint paths from s to each $d_i, 1 \leq i \leq k$. The goal of the proposed heuristic is the derivation of these subgraphs with a cost close to the cost of the ones derived by the ILP.

III. Integer Linear Programming (ILP) Formulation

The proposed ILP formulation for multicast routing and wavelength assignment is presented aiming at minimizing the total cost of establishing and protecting a multicast session in sparse-splitting optical networks. Specifically, the ILP is given as input a specific multicast instance; that is, a network topology, a placement of optical splitters (MC nodes), and a multicast session (the source node and the set of destination nodes). The ILP computes lightpath pairs from source to every destination with minimum aggregate cost such that a multicast connection request is established while protecting it from any single link failure.

In order to protect a multicast session, the path-pair protection approach [13] is used in which a path-pair (disjoint primary and backup paths) is computed for each destination and a path-pair is allowed to share edges with other path-pairs. Unlike other approaches, such as finding link-disjoint and arc-disjoint trees, this approach guarantees a solution where previous approaches fail and also finds an efficient solution requiring less network resources.

The following parameters and variables are used to formulate the sparse-splitting protection multicast problem.

Parameters:

- Network graph: $G=(V,E)$
- Set of multicast-incapable nodes: $N \subseteq V$
- Cost of arc mn: c_{mn}
- Number of destination nodes for the multicast session: k
- Multicast session: $S=\{s,d_1,...,d_k\}$, where s is the source node and $d_i, i=1,\ldots,k$ are the destination nodes of the multicast session

Variables:

- P^d_{mn}: Boolean variable representing the primary path from source node s to destination node d, equal to 1 if the path from source node s to destination node d occupies the link between nodes m and n, 0 otherwise.
- B^d_{mn}: Boolean variable representing the backup path from source node s to destination node d, equal to 1 if the path from source node s to destination node d occupies the link between nodes m and n, 0 otherwise.
- T_{mn}: Boolean variable equal to 1 if the arc between nodes m and n is used in either establishing or in protecting the multicast session, 0 otherwise (arc-variable).

Objective:

$$\textit{Minimize}: \sum_{mn} T_{mn} \cdot c_{mn}$$

Subject to the following constraints:

- Source node has an outgoing flow of one unit to the primary path and zero incoming flow:

$$\sum_{n} P^d_{sn} - \sum_{n} P^d_{ns} = 1, \forall d \in S \qquad (1)$$

- Destination node has an incoming flow of one unit to the primary path and zero outgoing flow:

$$\sum_{n} P_{nd}^{d} - \sum_{n} P_{dn}^{d} = 1, \forall d \in S \quad (2)$$

- Every intermediate node has the same incoming and outgoing flow for the primary path:

$$\sum_{n} P_{nm}^{d} = \sum_{n} P_{mn}^{d}, \forall d \in S, \forall m \neq s, d \quad (3)$$

- Constraints (1)-(3) are repeated for the backup paths:

$$\sum_{n} B_{sn}^{d} - \sum_{n} B_{ns}^{d} = 1, \forall d \in S \quad (4)$$

$$\sum_{n} B_{nd}^{d} - \sum_{n} B_{dn}^{d} = 1, \forall d \in S \quad (5)$$

$$\sum_{n} B_{nm}^{d} = \sum_{n} B_{mn}^{d}, \forall d \in S, \forall m \neq s, d \quad (6)$$

- The primary and backup paths do not share an arc:

$$P_{mn}^{d} + B_{mn}^{d} \leq 1, \forall d \in S, \forall m, n \in V \quad (7)$$

- Enable the arc variables that are used from the primary or the backup path:

$$\sum_{d} P_{mn}^{d} + \sum_{d} B_{mn}^{d} \leq 2 \cdot k \cdot T_{mn}, \forall m, n \in V \quad (8)$$

- An arc variable must be enabled only if this arc belongs to a primary or to a backup path:

$$\sum_{d} P_{mn}^{d} + \sum_{d} B_{mn}^{d} \geq T_{mn}, \forall m, n \in V \quad (9)$$

- For MI nodes, the incoming flow is greater than or equal to the outgoing flow for the primary paths:

$$\sum_{n} T_{nm} \geq \sum_{n} T_{mn}, \forall m \in N, m \neq s \quad (10)$$

The above formulation creates a primary path from the source to every destination node and a corresponding link-disjoint backup path. The objective function accounts for the total cost of establishing and protecting a multicast session. Constraints (1)-(3) correspond to the flow conservation constraints for the primary paths. Specifically, constraint (1) ensures that the source node has an outgoing flow of one unit to the primary path and has zero incoming flow. Constraint (2) ensures that the destination node has no outgoing flow and has an incoming flow of one unit from the primary path. Constraint (3) ensures flow conservation for intermediate nodes of one unit if it belongs to a primary path and zero incoming and outgoing flow if it does not belong to a primary path. Constraints (4)-(6) correspond to the same constraints as (1)-(3) for the backup paths. Constraint (7) guarantees that the primary and the backup paths do not share an arc. Constraints (8) and (9) are used to define the connection between variables P_{mn}^{d}, B_{mn}^{d}, and T_{mn}. Specifically, constraint (8) enables the arc variables that are used by the primary or the backup paths. Parameter k is used to count only once the usage of a specific arc by a primary or a backup path. Constraint (9) prohibits an arc variable to be enabled, unless a primary or a backup path uses this arc. Constraint (10) prohibits the splitting in a node with no splitting capabilities (MI node) for the primary and backup paths. This means that the outgoing traffic in an intermediate MI node should not be greater than the incoming traffic in order to prevent the splitting of the incoming signal. Signal splitting is only possible at MC nodes.

IV. Proposed Heuristic Algorithm

The proposed heuristic for provisioning survivable multicast requests in sparse-splitting optical networks, is called *Dual-route Sparse-splitting Heuristic (DSH)*. The output of the algorithm is a subgraph consisting of two link-disjoint paths $p_{d_i}^{1}$ (primary path) and $p_{d_i}^{2}$ (secondary path) for each destination d_i, $1 \leq i \leq k$, of the multicast request. A total number of $2k$ paths are calculated. Relevant protection methods for the case of full-splitting networks either calculate all the primary paths in the first step and all the secondary ones in the second (e.g., algorithm H-SDS in [13]), or calculate the primary and secondary paths for a destination and then proceed to the next destination (e.g., algorithm OPP-SDP in [13]).

In the proposed algorithm, the $2k$ paths are calculated by finding the path (either $p_{d_i}^{1}$ or $p_{d_i}^{2}$ for a destination that $p_{d_i}^{1}$ has already been found) that is closest to the current subgraph at each iteration. The algorithm terminates when $p_{d_i}^{1}$ and $p_{d_i}^{2}$ are obtained for every destination (the desired multicasting subgraph is derived), or when no other paths can be found (the desired multicasting subgraph is not derived).

For the derivation of a proper multicasting subgraph, the following constraints must be applied for the calculation of these paths: (a) Primary path: it can originate either from the source, or from an MC node in the current subgraph, or from an MI node in the current subgraph with $outdegree = 0$ and (b) Secondary path: it can originate either from the source, or from an MC node in the current subgraph, or from an MI node in the current subgraph with $outdegree = 0$. Furthermore, the secondary path must be link disjoint from the corresponding primary path. In order to achieve this, an *excluded set* must be created for the derivation of a secondary path. Let d_i be the corresponding destination. This (excluded) set consists of the nodes of $p_{d_i}^{1}$, starting from d_i, going backwards on $p_{d_i}^{1}$ and stopping when a node v is found, as a part of $p_{d_i}^{1}$, where there already exist two paths from the source node to node v. Node v is not part of the excluded set. The secondary path cannot originate from any node in the excluded set. Additionally, nodes that have their primary path originating from nodes in the excluded set (and for which a secondary path has not as of yet been found), must be added in the excluded set as well. The reason is that their primary paths are not link-disjoint from $p_{d_i}^{1}$.

After the derivation of a path, the corresponding nodes and arcs are added in the current subgraph SG and these arcs are removed from graph G, since the subgraph must not use more than one wavelength in each arc, as stated in Section

II. Let the constraints set up for the primary paths notated by $CONset_1$ and for the secondary paths by $CONset_2$. The function $Dijkstra(n_i, G)$ used in DSH, is the application of Dijkstra's algorithm [14] on graph $G(V, E)$, having n_i as the source (i.e., the calculation of the shortest paths originating from node n_i and ending at each other node in V).

The formal description of DSH is as follows:

1) $SG = \{s\}$, $A = \emptyset$, $false = 0$, $index[d_i] = 0$ for each $d_i, 1 \leq i \leq k$.
2) **While** $(A! = D$ and $false = 0)\{$
   a) For each node $n_i \in V$: Apply $Dijkstra(n_i, G)$.
   b) For each destination $d_i, 1 \leq i \leq k$, $d_i \notin A$:
   If ($index[d_i]$ equals 0) calculate the distance from SG to d_i s.t. $CONset_1$ and derive $SP[SG][d_i]$.
   Else calculate the distance from SG to d_i s.t. $CONset_2$ and derive $SP[SG][d_i]$.
   c) Find destination d^* with the smallest distance $|SP[SG][d^*]|$.
   If $(|SP[SG][d^*]| < \infty)$ {
   - add the nodes and arcs of $SP[SG][d^*]$ in SG and remove its arcs from G.
   - increment by 1 $index[d^*]$.
   - **If** ($index[d^*]$ equals 2) add d^* in A.}
   Else $false = 1$.
   }

The output of the algorithm is the desired subgraph SG if $false = 0$. Otherwise, SG was not possible to be derived.

A. Time-Complexity of DSH

1) Step 2 is repeated $2k$ times since two link-disjoint paths must be found for each one of the k destinations.
2) Step 2(a) can be executed in $O(nm \log_{(1+m/n)} n)$ time, if Dijkstra's algorithm is implemented as described in [15].
3) The *excluded set* for the secondary paths is created for each $d_i, 1 \leq i \leq k$, $d_i \notin A$ with $index[d_i] = 1$, for each iteration of Step 2(b). It can be created by deriving the predecessor on $SG(V_{SG}, E_{SG})$ of each node v in V_{SG} with $index[d_i] = 1$. The predecessors can be derived simultaneously with the calculation of $SP[SG][d_i]$ in Step 2(b). Therefore, the excluded set for each $d_i, 1 \leq i \leq k$, $d_i \notin A$ with $index[d_i] = 1$, can be calculated in $O(n)$ time, since it may consist of at most n nodes. The comparison of the cost of the shortest paths between the nodes of SG and node d_i can be performed in $O(n)$ time as well, since SG may consist of at most n nodes. The result is that Step 2(b) can be executed in $O(n)$ time for each $d_i, 1 \leq i \leq k$, $d_i \notin A$, i.e., in $O(kn)$ time.
4) Step 2(c) terminates after at most k iterations, since the distances between SG and at most k destinations are compared.

The conclusion of the above is that Steps 2(a)-2(c) are executed in $O(nm \log_{(1+m/n)} n)$ time, and the time-complexity of DSH is $O(knm \log_{(1+m/n)} n)$.

V. Performance Evaluation

A. Simulation Setup

The performance of the proposed heuristic algorithm (i.e., the cost of the multicasting subgraphs derived by it) was evaluated through simulations on the well-known USNET network [13], illustrated in Figure 1.

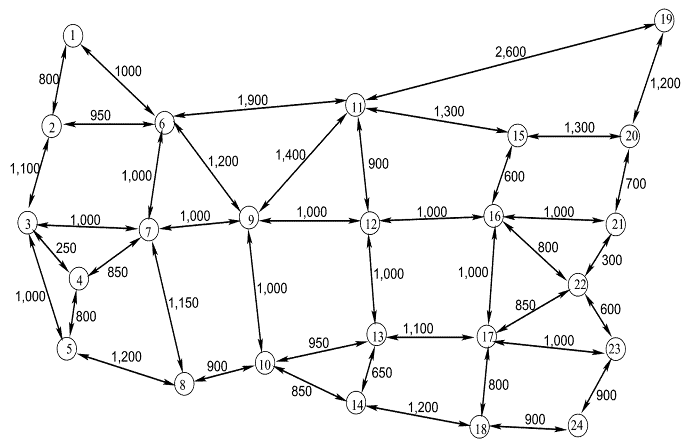

Fig. 1. Network graph used for simulations.

The USNET network consists of 24 nodes and 43 links and it is undirected (i.e., every link consists of two fibers of opposite orientation). A cost is assigned to each network link as shown in the figure.

The simulation was repeated for various possible destination-set sizes, from $k = 3$ to $k = 12$, with a step equal to 3. This procedure was executed $I = 500$ times for every case, while the source and destinations of the multicast connections were distributed uniformly across the network. For each destination-set size simulated, a multicasting subgraph was attempted to be derived using the proposed ILP formulation as well as the proposed DSH algorithm, for each one of the randomly created 500 multicast sessions. Since in some cases the ILP was able to give a subgraph and the DSH algorithm did not, the results were averaged over the cases where DSH succeeded in finding a subgraph. The exact formulation for the calculation of the average cost of the derived results is as follows:

$$C_x^k = \frac{1}{I^k} \sum_{i=1}^{I^k} c_{x,i}^k \tag{11}$$

The calculation of C_x^k was performed for the cases where DSH was able to derive a solution. In eq. (11), I^k represents the total number of repetitions of the procedure (i.e., $I^k = 500$ for each k), $c_{x,i}^k$ is the cost of the derived subgraph for the i^{th} multicast session for the case of k destinations, C_x^k is the average cost of the 500 subgraphs for the case of k destinations, x is the method used for the derivation of the subgraphs, and C_x^k was calculated for both $x = ILP$ and $x = DSH$.

The aforementioned procedure was executed for the cases where the 12.5% (i.e., 4 nodes) and 25% (i.e., 8 nodes) of the network nodes are MC. These MC nodes were placed in the network utilizing the *kmaxD* method as described in [16] (the MC nodes are placed at the nodes that have the largest degrees).

B. Simulation Results

Figures 2 and 3 present the simulation results. The average cost of the derived subgraphs for each examined destination-set size is presented for both the ILP and DSH. Figures 2 and 3 present the result for the case where 4 and 8 nodes respectively are considered to be MC.

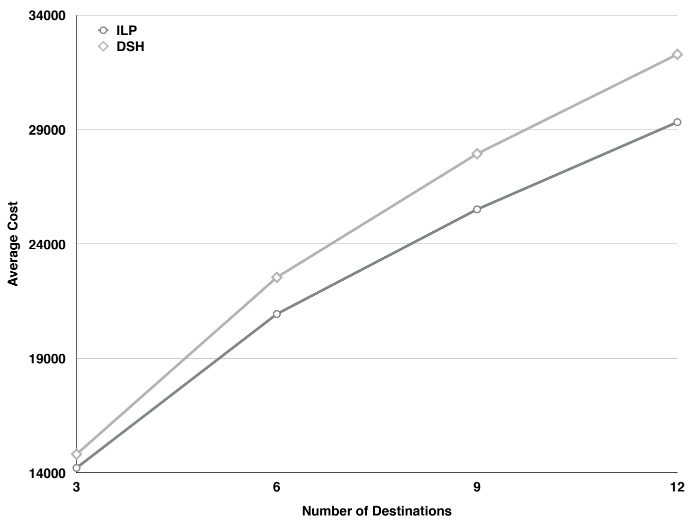

Fig. 2. Average cost of the derived multicasting subgraphs for each examined destination-set size (4 MC nodes).

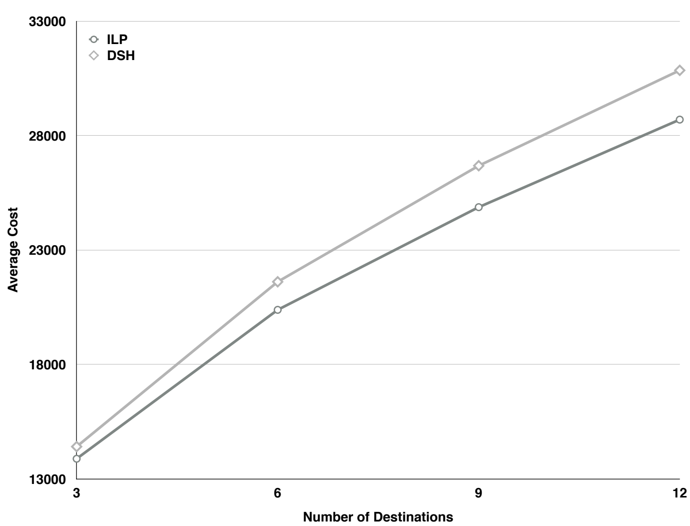

Fig. 3. Average cost of the derived multicasting subgraphs for each examined destination-set size (8 MC nodes).

From the description above, it can be easily calculated that the simulation consisted of 4000 multicast requests. The ILP formulation gave a multicasting subgraph for all of them, while the DSH algorithm gave a multicasting subgraph for 3996 of them, and failed to give a subgraph for 4 cases. Therefore, DSH succeeded in giving a solution for 99.9% of the total cases simulated. Note that in this case the 4 instances where the algorithm failed to produce a subgraph were excluded from the calculation of the average cost.

In terms of the average cost, DSH gives results close to the optimal ones obtained by the ILP, as shown in Figures 2 and 3. If the results are averaged over the various k that were simulated, DSH has 7.8% more average cost compared to the optimal one, for 4 nodes, and $6,1\%$ for 8 MC nodes respectively.

VI. Conclusions

In this paper, the problem of provisioning survivable multicasting in sparse splitting optical networks was investigated. An integer linear programming (ILP) formulation was presented, as well as a heuristic algorithm, called *Dual-route Sparse-splitting Heuristic (DSH)*. For the network investigated, simulations have shown that the proposed heuristic gives an average cost of the derived multicasting subgraphs that is very close to the optimal one obtained by the ILP, and that it succeeds to give an appropriate multicasting subgraph at practically every case one exists.

Future work will focus on the case of provisioning dynamically arriving survivable multicast requests.

Acknowledgment: This work was supported by the Cyprus Research Promotion Foundation's Framework Programme for Research, Technological Development and Innovation (DESMI 2009-2010), co-funded by the Republic of Cyprus and the European Regional Development Fund, and specifically under Grant TPE/EPIKOI/0311(BIE)/11.

References

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

[2] L. Sahasrabuddhe and B. Mukherjee, "Light-Trees: Optical Multicasting for Improved Performance in Wavelength-Routed Networks", *IEEE Communications Magazine*, 37(2):67–73, 1999.

[3] B. Mukherjee, *Optical Communication Networks*, Mc-Graw-Hill, New York, 1997.

[4] T. Panayiotou, G. Ellinas, N. Antoniades, and A. Levine, "Designing and Engineering Metropolitan Area Transparent Optical Networks for the Provisioning of Multicast Sessions", *Proc. IEEE/OSA Optical Fiber Communications Conference (OFC)*, San Diego, CA, 2010.

[5] X. Zhang, J. Wei, and C. Qiao, "Constrained Multicast Routing in WDM Networks with Sparse Light Splitting", *IEEE/OSA Journal of Lightwave Technology*, 18(12):1917–1927, 2000.

[6] W. Tseng and S. Kuo, "All-Optical Multicasting on Wavelength-Routed WDM Networks with Partial Replication", *Proc. International Conference on Information Networking*, Beppu City, Oita, Japan, 2001.

[7] R.M. Karp, "Reducibility Among Combinatorial Problems. Complexity of Computer Computations", chapter 8 in *50 Years of Integer Programming 1958–2008*, R.E. Miller and J.W. Thatcher, eds., Plenum Press, New York, 1972.

[8] X. Wang, S. Wang, L. Li, and S. Xu, "Achieving Shared Multicast Protection in WDM Networks with Sparse-Light-Splitting Constraint", *OSA Journal of Optical Networking*, 7(1):1–14, 2008.

[9] 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(12):1043–1053, 2009.

[10] N. Sreenath and T.S. Prasad, "Protecting Multicast Sessions from Link and Node Failures in Sparse-Splitting WDM Networks", *Proc. International Workshop on Distributed Computing (IWDC)*, Kharagpur, India, 2005.

[11] X. Wang, S. Wang, and L. Li, "Provisioning of Survivable Multicast Sessions in Sparse Light Splitting WDM Networks", *Proc. International Conference on Communications (ICC)*, Beijing, China, 2008.

[12] A. Frikha, S. Lahoud, B. Cousin, and M. Molnar, "Reliable Multicast Sessions Provisioning in Sparse Light-Splitting DWDM Networks using p-Cycles", *Proc. International Communications Quality and Reliability Workshop (CQR)*, San Diego, CA, 2012.

[13] N.K. Singhal, L.H. Sahasrabuddhe, and B. Mukherjee, "Provisioning of Survivable Multicast Sessions against Single Link Failures in Optical WDM Mesh Networks", *IEEE/OSA Journal of Lightwave Technology*, 21(11):2587–2594, 2003.

[14] E.W. Dijkstra, "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik*, 1(1):269–271, 1959.

[15] R.E. Tarjan, *Data Structures and Network Algorithms*, Society for Industrial and Applied Mathematics, 1983.

[16] S.W. Wang, "Allocation of Light Splitters in All-optical WDM Networks with Sparse Light Splitting Capabilities", *Telecommunication Systems*, 52(1):261–270, 2013.