Multicast *1+1* protection: the case for simple network coding

Mirzad Mohandespour
Dept. of Electrical and Computer Eng.
Iowa State University
Ames, IA 50011
Email: mirzadm@iastate.edu

Ahmed E. Kamal
Dept. of Electrical and Computer Eng.
Iowa State University
Ames, IA 50011
Email: kamal@iastate.edu

***Abstract*—We discuss how the idea of unicast *1+1* protection can be efficiently extended to protect multicast connections in optical backbone networks. Particularly, we show how to achieve instantaneous failure recovery and cost efficiency by allowing intermediate nodes to merge their incoming flows by a simple network code, i.e., logical OR operation. Under simple network coding, the problem of minimum cost multicast *1+1* protection is formulated as a 2-connectivity problem. In order to solve this problem, an optimal ILP and three efficient heuristic algorithms are proposed. Simulation results on real-world networks show that the average cost of our best heuristic algorithm is only 2.6% higher compared to the optimal ILP solution.**

I. Introduction

Multicast is a one-to-many traffic model in which a source node transmits the same information to a set of destination nodes. Such traffic model is used in backbone networks for provisioning high data rate applications such as Internet TV (IPTV) [1][2], distribution of financial information [3], and data dissemination in cloud and grid computing [4]. Many such applications demand highly available always-on connections. Underlying backbone networks, on the other hand, are subject to service disruption because of link and component failures. Moreover, even a single link failure can disrupt the connection to multiple destination nodes in the multicast traffic model. Therefore efficient multicast protection techniques that satisfy availability requirements are needed.

While multicast protection has been the subject of extensive studies, most of proposed techniques are tuned for non-instantaneous recovery, i.e., when a certain amount of delay in recovery is acceptable. In this paper we focus on the problem of multicast protection with instantaneous failure recovery.

Dedicated *1+1* protection has been commonly used in optical networks to provide instantaneous failure recovery against single failures for unicast connections. A pair of disjoint primary and backup routes are used to deliver two copies of each data unit from source to destination simultaneously. Failure of one route therefore causes no service disruption. If a multicast connection was provisioned as a set of independent unicast connections, the same technique could be directly applied to each connection. However, clearly that does not provide a cost efficient solution. When a tree is used to provision a multicast connection, a natural generalization of the idea of *1+1* protection is to have a pair of disjoint primary and backup trees connecting source to all destinations. While the cost efficiency is improved here, the required connectivity could be higher. If the network is 2-connected, one can always find a pair of disjoint paths between the source and each destination node but not necessarily a pair of disjoint trees.

The idea of *1+1* protection has been used in [5] to design a cost-efficient multicast protection technique against single link failures. A minimum cost disjoint path pair, Optimal Path Pair (*OPP*), is found from source to each destination. In order to reduce the cost, path pairs to different destinations are allowed to share bandwidth on common links. Figure 1 shows an example of how *OPP* works. A multicast connection is given with source node s and destination nodes d_1 and d_2 on a Butterfly network. Assuming that one unit of capacity is

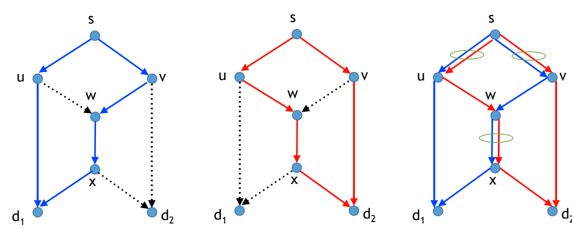

Fig. 1. Unicast *1+1* protection vs. *OPP*

reserved on each link of each path to each destination, the total reserved capacity without sharing is $6+6=12$ in unicast *1+1* protection. In the case of *OPP*, total reserved capacity would be reduced to $12-3=9$ because of sharing on links (s,u), (s,v), and (w,x). However the capacity improvement by *OPP* has a negative effect on the recovery delay.

Even though *OPP* finds two disjoint paths per destination node, because of the link sharing it cannot send two disjoint flows to all destination nodes. In Figure 1, node w can only forward one of the incoming flows from nodes u and v because only one unit of capacity is reserved on link (w,x). Figure 2 shows this situation. Suppose node w chooses to forward flow from v to the downstream node x, then destination node d_1 will receive two disjoint flows but that is not the case for node d_2.

Given a failure on link (s,v), node d_1 will still receive data through path $(s-u-d_1)$ however node d_2 will be totally

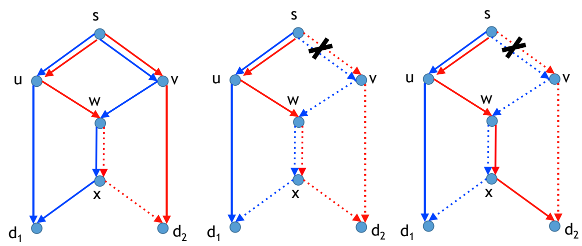

Fig. 2. Required switching at intermediate node w

disrupted. It is only after node w realizes that incoming flow from node v has failed and switches to incoming from node u that the flow to node d_2 could be restored through path $(s-u-w-x-d_2)$ (Figure 2). The recovery delay in this case is due to *OPP* not delivering two disjoint copies of each data unit to each destination. The same situation could happen for multiple intermediate nodes in a general multicast network.

In this paper we propose a solution to the multicast *1+1* protection problem that offers both cost-efficiency of sharing and instantaneous recovery of unicast *1+1* protection. In order to eliminate the recovery delay due to the sharing, our solution introduces the idea of merging flows at intermediate nodes which can be implemented using simple logical OR operation. The problem of minimum cost multicast *1+1* protection is then formulated as an ILP. The optimal results obtained by solving the ILP model are compared with two efficient heuristic algorithms for multicast *1+1* protection.

The rest of the paper is organized as follows: Section II explains the main idea behind multicast *1+1* protection by an example. Section III reviews the related work in the fields of multicast protection in networks and connectivity problems in graph theory. Section IV presents the assumptions and problem statement. In Section V, we present optimal formulation of the problem. Section VI describes the heuristic algorithms. Simulation results for optimal and heuristic algorithms are presented in Section VII. Finally Section VIII concludes the paper.

II. The Idea

As discussed earlier, the recovery delay in *OPP* is due to the fact that the failure will have to be detected, intermediate nodes (such as w in Figure 2) are signaled, and switches will have to be reconfigured. All of this can take tens of milliseconds. This delay can be avoided if the intermediate node merges the incoming flows into one outgoing flow by a simple logical OR operation. As Figure 3 shows, under normal operation node w would OR equal data units from u and v ($\alpha+\alpha=\alpha$). This results in a single data unit which is forwarded to node x. In case of a single failure on an upstream link, e.g., (s,u) or (s,v), node w receives an empty packet (equally a zero data unit) on one link and α on the other. The OR operation would still produce α. Therefore the outgoing flow on (w,x) will not be affected by any upstream single link failure. Both destinations, as a result, will at least receive one copy of α under any single link failure. This example shows how merging flows at intermediate node w, makes it possible to have the benefits of sharing (cost efficiency) and dedicated protection (instantaneous recovery) at the same time. In general, merging happens whenever an intermediate node has multiple incoming flows (belonging to the same multicast connection) that share an outgoing link.

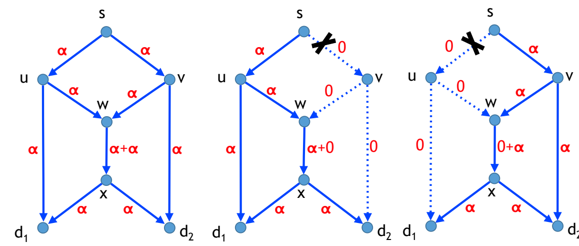

Fig. 3. Multicast *1+1* protection using merging flows

III. Related work

In [5] and [6] two different classifications of general multicast protection techniques are given. Here we focus on multicast protection techniques that offer instantaneous failure recovery. Due to the recovery requirement, such techniques are mainly categorized as dedicated protection techniques. Therefore in the case of multiple sessions, there is no inter-session backup sharing and each session is protected independently.

As mentioned before one idea is to find a primary tree and protect it with a disjoint secondary tree. Normally a Steiner Minimal Tree heuristic such as MPH (Minimum Path Heuristic) [7] is used to find the primary tree. In [8] a dual-tree approach was proposed. A primary tree is protected by secondary (node or link) disjoint tree that connects all the leaf nodes. More recently in [9] authors propose a novel Steiner tree heuristic called Steiner Node Heuristic (SNH). SNH incrementally adds non-destination nodes to the set of destinations and finds a new Steiner tree using MPH. The algorithm proceeds to the next step only if adding a new terminal reduces the cost. Therefore, by construction SNH is proved to be at least as good as MPH. SNH is then used to find a pair of disjoint trees.

The other idea is to use a ring subgraph to provision and protect a multicast connection. In [10] a collapsed-ring, i.e., a ring on which data is transmitted in both directions is proposed. The authors specifically describe their method as a multicast *1+1* protection.

In the context of network coding, *Robust Network Coding* [11] provides instantaneous failure recovery for multicast. Static linear codes are designed such that a feasible multicast rate can be protected against any failure pattern for which the rate remains feasible. In [12] a review of optical multicast protection using network coding is presented. Implementing robust network coding in optical backbone networks is challenging. The main problem is implementation of linear network coding functions at the optical layer. In [13] optical-electrical-optical (OEO) conversion is assumed at network nodes in order to implement linear network coding functions.

Robust network coding is then used to protect against single link failures. Authors propose a minimum cost network coding subgraph formulation and a multi-objective formulation to account for the cost of OEO ports.

An all-optical implementation of robust network coding is presented in [14]. Instead of OEO converters which require terminating of optical signal, all-optical implementation of linear network coding is discussed including optical switching, buffering, and logical operations. The problem of unit rate multicast protection against k link failures is then addressed using robust network coding. A heuristic algorithm, Robust Coded Multicast (RCM), for the general case of k failures and an ILP formulation for the special case of single failures is proposed. We make the observation that in the case of unit rate multicast, one simple network code, i.e., OR operation is sufficient thereby simplifying both the algorithm and the implementation.

IV. Assumptions and problem statement

The backbone network is modeled as directed graph $G(V, E)$ where V is the set of nodes and E is the set of directed edges. A physical bidirectional link between a pair of nodes u and v is modeled as two edges: (u, v) and (v, u). One wavelength channel is defined as the unit capacity. We assume that each link carries W wavelength channels in each direction. Failure of a link causes all those channels to fail. This is modeled by removal of both directed edges. The cost of reserving one unit of capacity on edge (u, v) is defined as c_{uv} which could be different for each edges depending on physical link properties such as physical length. A multicast request is represented as $M(s, D = \{d_1, ..., d_k\})$ where $s \in V$ is the source node and there are k distinct destination nodes such that $\forall i, d_i \in V$ and $s \notin D$. We assume unit multicast rate, i.e., multicast demand can be delivered over a single wavelength-channel in an optical network. This assumption is justified by the high bandwidth offered by a single optical channel. Moreover we assume static traffic model and no traffic grooming. We further assume that network nodes are capable of merging incoming flows by simple logical OR operation. Finally, the single link failure multicast *1+1* protection problem is defined as follows:

Problem. *Given $G(V, E)$ where nodes are capable of merging incoming flows and a unit rate multicast connection $M(s, D = \{d_1, ..., d_k\})$, find the minimum cost (link cost) subgraph $H \subseteq G$ that provides instantaneous single link failure recovery.*

In the following we describe necessary and sufficient conditions for subgraph H.

Lemma 1. *Subgraph $H \subseteq G$ provides instantaneous single link failure recovery iff it includes 2 link-disjoint paths from s to each d_i.*

Proof: (Sufficient condition) Assume H includes 2 link-disjoint paths from s to every d_i which are denoted by p^1_{s,d_i} and p^2_{s,d_i}. A single link failure could at most hit one of the two paths for each destination. For a specific destination d_i, suppose p^1_{s,d_i} is failed and p^2_{s,d_i} is not. The data unit traveling on p^2_{s,d_i} would possibly be merged with other flows at intermediate nodes. Since any such merging operation would be a logical OR operation whose other operands are either zero or the same data unit (from other intact flows), the data unit traveling on p^2_{s,d_i} will not be affected by the failure and will be delivered to d_i. In the same way any destination node will receive at least one copy of each data unit in the event of any single link failure.

(Necessary condition) Assume subgraph H provides unit rate multicast connection $M(s, D = \{d_1, ..., d_k\})$ with instantaneous recovery for any single link failure. If H does not include (at least) two link disjoint paths from s to (at least) one destination d_i, then min-cut between s and d_i is at most 1. This means there is a single link whose failure disconnects s from d_i which contradicts the assumption of single failure protection. ■

It is also worth noting that the necessary condition applies to any approach that provides single link failure protection even if it does not support instantaneous recovery. Finding minimum cost subgraph H that provides bi-connectivity between source and each destination node is known to be NP-hard [14].

V. Optimal formulation

Based on the necessary and sufficient conditions presented in Section IV, the problem of minimum cost multicast *1+1* protection is equivalent to finding a minimum cost subgraph that provides bi-connectivity between the source and each destination. Such subgraph can be optimally found using the following ILP formulation.

$$Minimize: \sum_{(u,v)\in E} c_{uv} x_{uv} \tag{1}$$

$$\sum_{(u,v)\in E} f^{d_i}_{uv} - \sum_{(w,u)\in E} f^{d_i}_{wu} = \begin{cases} +2 & u = s \\ -2 & u = d_i \\ 0 & o.w. \end{cases} \quad \forall u \in V, \quad \forall d_i \in D \tag{2}$$

$$x_{uv} \geq f^{d_i}_{uv} \qquad \forall (u, v) \in E, \forall d_i \in D \tag{3}$$

$$f^{d_i}_{uv}, x_{uv} \in \{0, 1\} \qquad \forall (u, v) \in E, \forall d_i \in D \tag{4}$$

Binary variable x_{uv} is equal to one if edge (u, v) is used in the solution. Binary variable $f^{d_i}_{uv}$ represents the flow from s to d_i on edge (u, v). Equation 1 presents the total cost of the solution to be minimized. Equation 2 is flow conservation at the source, destination nodes, and other intermediate nodes. Equation 3 makes sure that $x_{uv} = 1$ if edge (u, v) is used by any flow. Equation 4 defines the binary variables.

The ILP basically sends 2 units of flow from s to each destination d_i. Since flow variables are defined as binary, 2 units of flow would be carried by 2 edge-disjoint paths.

VI. Heuristic algorithms

The problem of minimum cost *1+1* protection for multicast is NP-hard. Obtaining the minimum cost by solving ILP formulation may not be practical for the real scenarios of dynamically changing multi-session multicast traffic. Therefore it is necessary to propose heuristic algorithms capable of providing fast yet efficient online solutions.

The core problem is 2-connectivity from source to all destinations. The 1-connectivity problem is the famous Steiner tree problem for which there is a well-known heuristic, i.e., Minimum Path Heuristic (*MPH*) [7]. The idea is to find the closest destination to the source, connect it by shortest path, set the cost of edges on the path to zero, then find next closest destination, and continue until all destinations are covered. The very same idea can be extended to build a 2-connected subgraph: substituting the notion of shortest path with shortest disjoint path-pair which can be found by Suurballe's algorithm [15] (Algrithm 1). We call this Minimum Path-Pair Heuristic (MPPH).

While *MPPH* maintains 2-connectivity at each step, another method is to start from a 1-connected subgraph (Steiner tree) and augment it to a 2-connected one. In [14] authors have proposed an algorithm based on a similar idea. First a Steiner tree is found, then the connectivity to destinations is augmented one at a time. This is done by removing the edges of the path from source to each destination on the Steiner tree and then finding a second shortest path to that destination. We propose an algorithm that augments a Steiner tree found by *MPH* to a 2-connected subgraph using *MPPH*. Hence called *MPH+MPPH* (Algorithm 2). The cost of Steiner edges returned by *MPH* are set to zero so that *MPPH* has incentive to use Steiner tree edges. In the simulation resutls we also use a more involved version of this algorithm called *MPH+MPPH(all)* which basically runs $|D|$ instances of *MPPH*. In each instance one destination node is fixed as the first destination in *MPPH* algorithm.

Algorithm 1 Minimum Path-Pair Heuristic: *MPPH*

Input: $G(V,E)$, $M(s,D)$
Output: Subgraph H
1: $H \leftarrow s$
2: **while** $D \neq \emptyset$ **do**
3: $\quad j \leftarrow argmin_{d_i \in D}(|p_{d_i}|)$
$\quad \{p_{d_i}$ is the shortest path-pair from s to d_i in $G\}$
4: $\quad \forall(u,v) \in p_{d_j} : c_{uv} \leftarrow 0$
$\quad \{$updates the cost of edges of path-pair p_{d_i} in $G\}$
5: $\quad H \leftarrow H \cup p_{d_j}$
6: $\quad D \leftarrow D \backslash d_j$
7: **end while**
8: return H

Time complexity of *MPH* [7] is $|D|.O(S)$ where $|D|$ is number of destinations and $O(S)$ is the time complexity of shortest path algorithm. Time complexity of *MPPH* depends on finding a shortest path-pair (step 3) which has the same time complexity as shortest path algorithm. In each iteration of *MPPH* we need to find the destination with minimum shortest path-pair among the remaining destinations. Therefore the time complexity of *MPPH* is $|D|^2.O(S)$. Time compexlity of *MPH+MPPH* hence is the same as *MPPH*. Finally

Algorithm 2 *MPH+MPPH*

Input: $G(V,E)$, $M(s,D)$
Output: Subgraph H
$T \leftarrow MPH\,(G(V,E), M(s,D))$
$\forall(u,v) \in T : c_{uv} \leftarrow 0$
$\{$updates the cost of edges of T in $G\}$
$H \leftarrow MPPH\,(G(V,E), M(s,D))$
return H

MPH+MPPH(all) would have time complexity of $|D|^3.O(S)$. In our implementation $O(S) = O(|V|^2)$ however a more efficient implementation can achieve $O(|E| + |V|log|V|)$ for Dijkstra's shortest path algorithm.

VII. Simulation Results

The original Pan-European network COST239 (11 nodes, 26 links) [16] and a modified version are used in the simulations. In the modified version, evey link (u,v) in original network is replaced by a new node w_{uv} and two new links (u, w_{uv}) and (w_{uv}, v). Therefore it has 11+26=37 nodes and 2*26=52 links. We refer to it as COST239+. For COST239 two cost functions are used: 1- unit link cost where links have equal unit cost, 2-physical distance cost where distance between cities is used as the link cost (in km). In the case of COST239+, we only consider physical distance and the new nodes are assumed to be halfway between original nodes.

In each case the costs reported by ILP and heuristics (*OPP*, *MPPH*, *MPH+MPPH*, and *MPH+MPPH(all)*) are compared. Our results cover the complete range of session size which is 2 to 11 for COST239 and 2 to 37 for COST239+. In each case 100 random multicast sessions are generated and the average cost for each session size is calculated. The same random sessions are applied as input to ILP and heuristic algorithms.

Figure 4 shows the results for COST239 with unit distance cost. While all heuristics perform well compared to ILP, *MPH+MPPH* and *MPH+MPPH(all)* are almost the same as optimal. Figure 5 shows the results for COST239 with physical distance as the link cost. Here again *MPH+MPPH* and *MPH+MPPH(all)* peform better than *OPP* and *MPPH*. In Figure 6 the results on COST239+ network are presented. Again we observe that *MPH+MPPH* and *MPH+MPPH(all)* perform very close to optimal. For the sake of readability, in each figure we have only shown a subset of session sizes for which the optimal vs. heuristic difference is more visible. In Table I we summarize the average and worst case performance (over all session sizes) of our best heuristics and *OPP* compared to optimal.

TABLE I
Average/worst case percentage of extra cost (vs. optimal)

Network	OPP	MPH+MPPH	MPH+MPPH(all)
COST239(unit)	4.4/8.6	0.6/2.6	0.1/0.7
COST239(phys)	8.3/11.3	5.0/7.4	2.6/4.7
COST239+(phys)	5.5/7.7	2.6/4.4	1.6/2.4

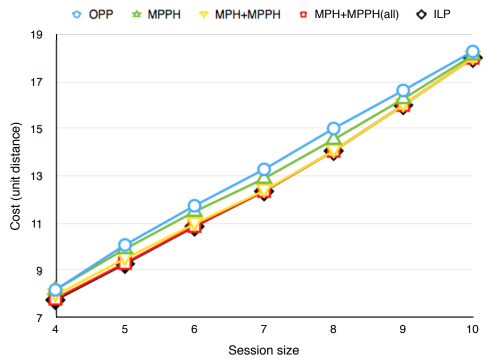

Fig. 4. COST239 network: unit link cost

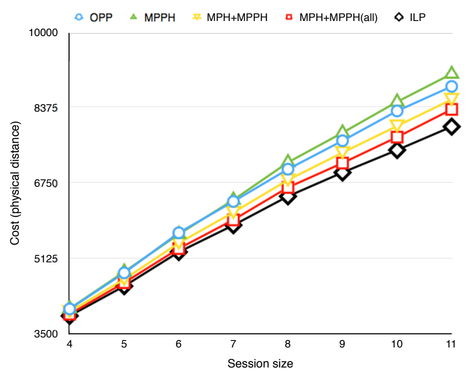

Fig. 5. COST239 network: physical distance as link cost

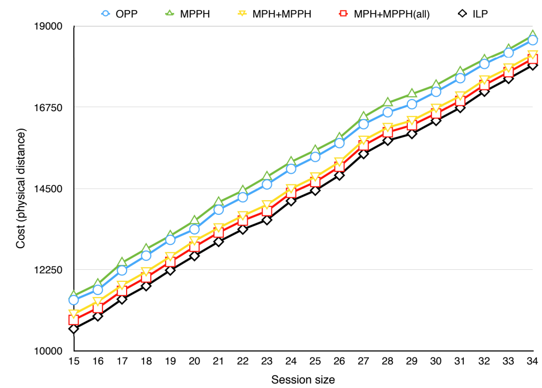

Fig. 6. COST239+ network: physical distance as link cost

VIII. Conclusion

The idea of *1+1* protection is extended to multicast protection using simple network coding. The *1+1* protection sends, simultaneously, two copies of each data unit to every destination, and simple network coding (OR operation) guarantees that upon any single link failure, at least one data copy is received by all destinations. No rerouting or switch reconfiguration is required and destination nodes would not experience any service disruption under any single link/node failure. The necessary and sufficient condition for the existence of *1+1* protection solution under merging flows is simply 2-connectivity from source to each destination. This allows us to easily formulate the problem as a network flow problem which can be solved to find minimum cost subgraph supporting *1+1* protection. An optimal ILP formulation of the problem and three heuristic algorithms are proposed as offline and online solutions. The simulation results on two sample networks show impressive performance by our top two heuristics as compared to the optimal: on average, our best heuristic increases the cost by no more than 2.6% and in the worst case, the gap between our best heuristic and optimal is only 4.7%. Future work would include implementation of OR operation, other failure models and traffic models, e.g., dynamic multi-session multicast.

References

[1] M. Cha, G. Choudhury, J. Yates, A. Shaikh, and S. Moon, "Case study: Resilient backbone design for iptv services," in *Workshop on IPTV Services over World Wide Web*, 2006.

[2] Y. Xiao, X. Du, J. Zhang, F. Hu, and S. Guizani, "Internet protocol television (iptv): the killer application for the next-generation internet," *Communications Magazine, IEEE*, vol. 45, no. 11, pp. 126–134, 2007.

[3] N. Maxemchuk, D. Shur, *et al.*, "An internet multicast system for the stock market," *ACM transactions on computer systems*, vol. 19, no. 3, pp. 384–412, 2001.

[4] K. Chen, C. Hu, X. Zhang, K. Zheng, Y. Chen, and A. V. Vasilakos, "Survey on routing in data centers: insights and future directions," *Network, IEEE*, vol. 25, no. 4, pp. 6–10, 2011.

[5] N. Singhal, L. Sahasrabuddhe, and B. Mukherjee, "Provisioning of survivable multicast sessions against single link failures in optical wdm mesh networks," *Journal of lightwave technology*, vol. 21, no. 11, p. 2587, 2003.

[6] W. Zhong and F. Zhang, "An overview of p-cycle based optical multicast protection approaches in mesh wdm networks," *Optical Switching and Networking*, vol. 8, no. 4, pp. 259–274, 2011.

[7] H. Takahashi and A. Matsuyama, "An approximate solution for the steiner problem in graphs," *Math. Japonica*, vol. 24, no. 6, pp. 573–577, 1980.

[8] A. Fei, J. Cui, M. Gerla, and D. Cavendish, "A dual-tree scheme for fault-tolerant multicast," in *Communications, 2001. ICC 2001. IEEE International Conference on*, vol. 3, pp. 690–694, IEEE, 2001.

[9] C. Constantinou and G. Ellinas, "A novel multicast routing algorithm and its application for protection against single-link and single-link/node failure scenarios in optical wdm mesh networks," *Optics Express*, vol. 19, no. 26, pp. B471–B477, 2011.

[10] T. Rahman and G. Ellinas, "Protection of multicast sessions in wdm mesh optical networks," in *Optical Fiber Communication Conference, 2005. Technical Digest. OFC/NFOEC*, vol. 2, pp. 3–pp, IEEE, 2005.

[11] R. Koetter and M. Médard, "An algebraic approach to network coding," *Networking, IEEE/ACM Transactions on*, vol. 11, no. 5, pp. 782–795, 2003.

[12] A. E. Kamal and M. Mohandespour, "Network coding-based protection," *Optical Switching and Networking*, 2013.

[13] M. Kim, M. Médard, and U.-M. O'Reilly, "Network coding and its implications on optical networking," in *Optical Fiber Communication Conference*, Optical Society of America, 2009.

[14] E. Manley, J. Deogun, L. Xu, and D. Alexander, "All-optical network coding," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 2, no. 4, pp. 175–191, 2010.

[15] J. W. Suurballe and R. E. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, no. 2, pp. 325–336, 1984.

[16] P. Batchelor, *Ultra high capacity optical transmission networks: Final report of action COST 239*. Faculty of Electrical Engineering and Computing, 1999.