Resilient Routing Layers and p-Cycles: Tradeoffs in Network Fault Tolerance

Tarik Čičić*, Amund Kvalbein*, Audun Fosselie Hansen*†, Stein Gjessing*
Email: {tarikc, amundk, audunh, steing}@simula.no, Telephone: (+47) 67 82 82 00
* Simula Research Laboratory, PB. 134, 1325 Lysaker, Norway † Telenor R&D, 1331 Fornebu, Norway

***Abstract*— We compare p-cycles and the recently introduced Resilient Routing Layers as candidate schemes for network-level fault protection. Using computational routing trials we show that RRL has shorter backup path lengths and more successful double-link fault protection. On the other hand, p-cycles may require less forwarding state. Several tradeoffs of interest for network designers are described.**

***Index Terms*— Computer network reliability, recovery, protection, performance, routing.**

I. Introduction

Recent evolution of the Internet toward the critical tool for business and real-time communications emphasizes the importance of service reliability and availability. Internet network resilience in presence of link and router outages is however unsatisfactory. One problem is that network protection is often deployed on the link level and below, and thus unable to catch network-level issues such as router software malfunctions. Another problem is that standard IP routing protocols such as OSPF perform network recovery by rerouting, which can be too slow for certain network services. Much faster recovery can be achieved by network protection—if the backup routes are calculated before the failure.

Fast recovery performance of network protection comes at the cost of increased forwarding state. With rerouting, new forwarding state can be calculated based on the original routing tables and the received failure information. With protection, the forwarding state is pre-calculated and stored in the network nodes.

Network protection seemingly defies the default Internet Protocol behavior in that IP is connectionless, while protection schemes define backup paths that diverge from the default IP routes. There are however proposals to use MPLS [1] or multi-topology routing [2] as IP protection technologies.

p-Cycles [3] is among the most researched network protection schemes. Primarily targeting optical networks, p-cycles is also studied in the IP context [4]. The idea is to configure one or more logical cycles to protect all links and, optionally, nodes. When a link or node fails, the traffic is forwarded on its protection cycle, around the failure.

We recently introduced a new recovery scheme called Resilient Routing Layers (RRL), for guaranteed protection from a single node or link failure in biconnected networks [5]. RRL generalizes interconnect fault-tolerance concepts introduced in [6]. The idea in our approach is that in each network we can calculate a set of *safe layers*, i.e., spanning topology subsets that can handle any traffic affected by a fault in any network node or link. These layers should be calculated in advance, and can be used for safe packet forwarding to all destinations over the operational part of the network.

RRL has several qualitative advantages compared to other known protection methods. The layer layout has few constraints. Layers can be efficiently constructed algorithmically, and can be optimized in various dimensions. RRL's main application domain is packet networks, and it easily supports connectionless protection. RRL covers both link and node failures per default and through the same mechanism.

In this paper we make an initial comparative evaluation of RRL and p-cycles fault tolerance in IP networks. We first present the technology basics. We then compare the state requirements, double link failure protection success rate and the backup path lengths of the schemes. Finally, we provide a conclusion and some interesting directions for future work.

II. Technology Background

A. p-Cycles

1) p-Cycles layout: In p-cycles, the network is covered by a set of logical rings. Each ring protects all links it contains, but also all "straddling" links, i.e., links between non-neighboring nodes within the ring. It is not specified how the rings are to be arranged on a topology, and there are many possible layouts. A layout can be created manually or by a centralized algorithm.

A single cycle incorporating all nodes in the network is one possible p-cycles layout. Creating such a cycle in an existing network is the well-known NP-complete Hamiltonian Cycle problem. If links designated to network protection can be added to the network, it becomes realistic to use a single cycle as the p-cycles layout. Recent research has demonstrated that configuring one Hamiltonian cycle can be the most bandwidth-efficient layout when new links are added [7].

Working with Hamiltonian cycles has a limited use in topologies where new links cannot be added, due to the computational hardness of the problem. Instead, we can use practical heuristics to create reasonably large cycles.

p-Cycles redirects the network traffic so it follows the cycle layout. In the case of large cycles, this is expected to cause long protection paths. Surprisingly, p-cycles protection path lengths have not been studied in the literature, to our best knowledge. Intuitively, smaller cycles should have shorter backup path lengths.

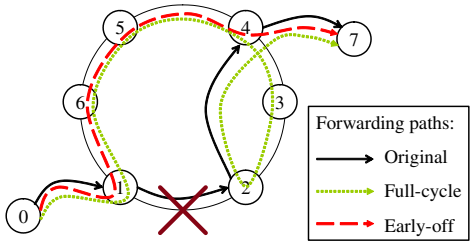

Fig. 1. Full-cycle and early-off backup paths.

p-Cycles backup path state requirements at each node are limited by the node degree d (i.e., number of links), and the number of p-cycles the node is a part of. [4] sets the number of additional routing entries to $2d$ per cycle per node.

Recent research has shown that p-cycles can be successfully used to protect against double link failures. It is shown that p-cycles offers 50-70 % traffic restoration under double link faults, using the available spare link capacity in a selected WDM network [8]. 90 % protection can be reached with an algorithm improvement in a 3-connected network with 90 % spare link capacity increase [9]. If only path availability is analyzed, up to 95 % dual-failure restoration is shown [10].

2) Implementation strategy: Once the p-cycles are generated, a network mechanism must exist that will utilize them for fault protection. In IP networks, MPLS paths may be laid between selected node pairs in each cycle. In case of failure, the node makes a local decision where to forward the packets in order to avoid the failed link.

It can be distinguished between two forwarding strategies for p-cycles. The packets can either be forwarded the full round on the cycle to the neighbor on the opposite side of the failed link, or the packets may leave the cycle on a node closer to the destination. The first strategy we call *full-cycle* forwarding, and the other strategy we call *early-off* forwarding.

Full-cycle forwarding can be expected to result in poorer backup paths. In Fig. 1, path 0-1-2-4-7 is broken in link 1-2. It is protected by the cycle that includes nodes 1, 2, ..., 6, and the backup path will be 0-1-6-5-4-3-2-4-7. It would be more optimal to leave the cycle from a node closer to the destination (node 4 in Fig. 1). As an implementation strategy for early-off forwarding, [4] proposes extending the packet headers by the original path length from the cycle-entry node to the destination. The forwarding logic in the nodes would then have to check at each hop whether its path to the destination is shorter than the original path, and, if so, release the packet from the cycle.

The proposed early-off implementation strategy implies packet header modifications and more complicated forwarding logic in network nodes. An alternative implementation strategy could be to maintain a per-node mapping from adjacent links and all destinations to cycle-exit nodes. Then, the node that redirects the packet to the cycle can determine from which node the packet should leave the cycle.

Compared to the full-cycle forwarding, the shorter backup paths of early-off come at the cost of either an increase in the state amount or a more complex forwarding mechanism. Both forwarding strategies may have their application areas, and in this paper we evaluate both.

B. Resilient Routing Layers

1) Layers layout: RRL generates spanning topology subsets, or *layers*, which include all nodes but only a subset of the links in the network. For each node, there must exist a layer in which this node will not be used to forward any transit traffic. We say that the *node is safe* in that layer. A link is safe in a layer if it is *not* included in that layer.

One way to make a node safe in a layer is to exclude all its links except one from that layer. A routing algorithm determines the forwarding information so that the safe nodes are not used for transit traffic. Clearly, shortest-path routing algorithms, or any routing algorithms with a requirement for loop-freeness, will not forward traffic through a safe node. Consequently, if a node fails, the traffic between any other two nodes in the network can be routed in the failed nodes safe layer.

Layers can be constructed manually or by a centralized algorithm. Manual construction may be used to optimize specific protection features, such as heavy reliance on links known to be more stable. Algorithmic layer construction may be preferred. Different algorithms can be used to optimize different performance metrics of RRL. For example, the number of layers can be kept low, minimizing the additional forwarding state requirements. Our study [5] shows that, even in large networks, 3-4 layers often suffice to make all nodes safe—6 layers was the highest number we encountered in a large range of tested real and synthetic random topologies. Furthermore, if relatively few links per layer are removed, the protection path lengths will be shorter. Guaranteed protection against link failures only can be achieved with only few layers [11]. If many links per layer are removed, the layers will more often tolerate multiple failures. Multi-fault tolerance can be improved by adding more layers, but comes on cost of the increased state [12].

In this study, we use an algorithm that creates the layers as follows:

1) layers are generated one by one
2) nodes are made safe by removing links from the currently constructed layer
3) links are removed so that, whenever we can choose between more candidates, the links that have been removed fewest times so far are removed
4) the algorithm terminates when all nodes are safe in at least one layer and the specified number of layers is created.

We illustrate the algorithm on the sample network shown in Fig. 2a, and assuming that three layers are to be created. We start with creation of the first layer. Nodes 1, 2 and 3 are made safe first. Node 4 cannot be made safe in this layer, since the topology will become disconnected, but node 5 can, resulting in the layer depicted in Fig. 2b. Then we proceed with creation of the second layer, where nodes 4, 6, 7 and 8 are safe (Fig. 2c). Now, all nodes are safe in one layer, and

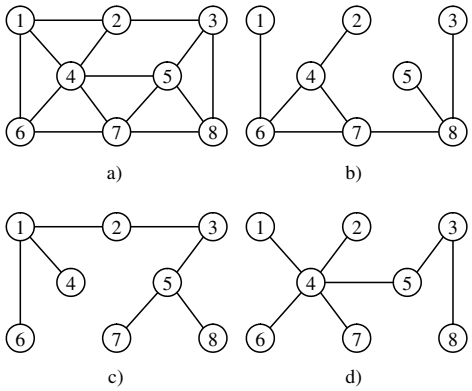

Fig. 2. Sample network topology (a), and three layers that provide fault tolerance for all links and nodes, (b, c and d).

all links are safe in one layer, except links 1-6 and 5-8, which are not safe in any layer, and link 4-5, which is safe in both layers. In the third layer, links 1-6 and 5-8 are excluded by the algorithm (i.e., safe), while link 4-5 is intentionally kept in this layer, since it is safe in two layers already. A possible layout for the third layer is shown in Fig. 2d.

2) Implementation strategy: Once the layer information is disseminated among the network nodes, it can be utilized for network protection. First, the routing algorithm calculates per-layer backup routes. Each node stores a mapping between the adjacent links and nodes and the safe layers for these links and nodes. Here, we assume that the nodes can distinguish between the link and node failures.

When the router detects an adjacent link or node failure, it forwards the packets in the safe layer for the failed link or node. The packet header has to carry information about the current routing layer. Only four bits are needed for up to 15 layers and the original topology; this information can be carried in, e.g., unused header bits, utilizing unused or private address space, or extension headers. This information is local for the network where the layers are calculated.

A router that receives a packet on a given layer, forwards the packet in the same layer. However, if another failure is encountered, care must be taken to avoid permanent routing loops. We suggest the rule that the layers are enumerated from zero upwards, zero being the original full topology. If a failure is detected, the packet can be forwarded in a layer higher than the current layer. If there is no higher layer that is safe for the detected failure, the packet is dropped. Thus, the loops can be avoided with only local knowledge of failed links or nodes.

III. Evaluation Method

A. Metrics

1) State: The fast recovery performance of network protection comes at the cost of the increased forwarding state. Clearly, minimizing the amount of state information is an important feature of any protection scheme.

For each protection scheme, we distinguish between the *addressing needs* and the *technology platform*. By addressing needs we mean the number of network destinations a node needs to know in order to redirect the traffic on the functional links. Technology platforms can be divided into connection-oriented and connectionless, and may be able to optimize storage requirements.

2) Backup path lengths: Backup path length is an important metric for the protection schemes. After the failure, different schemes will choose different backup paths. Shorter backup paths are generally an advantage, since they imply less overall network resource demands.

3) Dual-link fault tolerance: It is well understood that a k-connected network can guarantee recovery from at most (k–1) failures. Due to cost and complexity considerations, most recovery schemes are designed to guarantee single-fault protection, and require biconnected networks.

Regarded as a probability measure rather than a hard guarantee, multi-fault tolerance can be studied in any network. Double link failures represent a realistic and relevant failure scenario [10], and both RRL and p-cycles are well-suited to address them.

B. Evaluation Model

Regarding the state requirements of p-cycles and RRL, our intention is to determine which scheme has better scalability properties. In this paper we restrict our selves to the addressing needs, and estimate the backup state overhead ratio for these two schemes.

Backup path length and dual-link fault protection evaluation is done in families of 32, 64, 128, 256, 512 and 1024-node random networks generated using the BRITE network generation tool [13] with the Waxman model [14]. We believe that this range provides a reasonable choice of practically relevant network sizes. All studied networks have twice as many links as nodes and are biconnected. 100 networks are generated in each family.

In the path-length and dual-link failure evaluations, the backup route availability is central for IP-level recovery. We therefore use an evaluation model similar to one described in [10], rather than a link capacity model (as in, e.g., [9]). In the random networks, we algorithmically calculate the p-cycles and routing layers. We then make a number of computational routing trials for up to 1000 node pairs and two link failures on the shortest path between them. We say that a protection case is successful if, after the failures, the source and destination nodes retain connectivity using the applied protection scheme. We log the number of successful protection cases and the backup path lengths[1].

In this study, we generate p-cycles using a ring-cover algorithm similar to the one presented in [15], but where we use our own cycle generation heuristics.

When creating large cycles, we start with an arbitrary cycle, and extend it by replacing its links by paths that start and end in the endpoints of the link and that traverse nodes that are not yet in the cycle. These link-for-path substitutions are done as long as any suitable nodes are available. If additional

[1]The software we implemented for purpose of this study can be downloaded from *www.ifi.uio.no/∼tarikc/software/RRL/protection.tar.gz*

TABLE I

PROPERTIES OF THE GENERATED p-CYCLES, MEAN VALUES.

Topology size (Num. nodes)	Large cycles Number	Large cycles Length	Small cycles Number	Small cycles Length
32	3.38	23.64	23.97	4.36
64	4.95	42.71	44.85	4.82
128	7.35	79.00	85.21	5.37
256	10.00	151.82	161.50	5.89
512	13.80	276.64	317.00	6.38
1024	18.40	524.11	609.80	6.85

TABLE II

PROPERTIES OF THE GENERATED ROUTING LAYERS AND THE SHORTEST PATHS BEFORE AND AFTER THE FAILURES.

Topology size (Num. nodes)	Number layers	Shortest path length Original	Shortest path length After failure
32	4.00	2.73	3.71
64	4.01	3.14	4.24
128	4.01	3.60	4.81
256	4.00	4.06	5.37
512	4.04	4.50	5.91
1024	4.11	4.95	6.44

large cycles need to be created, the first cycle is constructed so that it covers up to three preferred nodes. This large-cycle heuristic runs in polynomial time, with algorithmic complexity of $\mathcal{O}(n^4 log^2 n)$, n being the number of nodes.

Small p-cycles are easily created from a given node and two of its neighbors using the shortest path between the neighbors not passing through the node itself.

The cycles are generated until all links are protected, either as a part of a cycle, or straddling a cycle. Table I shows some basic properties of the generated cycles.

The RRL algorithm delineated in Sec. II-B was run to create four layers, or more if necessary to make all nodes safe. Properties of the generated layers are summarized in Tab. II. Most topologies could be protected by 4 layers. For a comparison, we also included the shortest path lengths in the full topologies, and the topologies without the failed links. (These values are the same for p-cycles and RRL.) The shortest paths were increased by 0.98-1.49 hops on average.

IV. EVALUATION RESULTS

A. State Scalability

We assume that, in a network of n nodes, the p-cycles algorithm has created a cycles of average length b. In a large number of cycle layouts, each node will be in ab/n cycles on average. This measure we call the *p-cycles multiplier* and define

$$M = \frac{ab}{n}$$

Furthermore, we assume that the RRL algorithm generated l layers. The default amount of routing state we denote D.

In p-cycles with full-cycle forwarding, each node needs to address all nodes it has a common link with, on-cycle or straddling. In the network seen as a whole, the average per-node addressing needs will be limited by the average node degree d. The additional per-node backup path addressing needs for full-cycle forwarding can be expressed as

$$A_{\mathrm{FC}} = dM$$

For early-off forwarding, each node in the cycle may be the exit node. The exit node is chosen based on which link has failed and the destination address. If early-off is implemented using additional forwarding state, we have

$$A_{\mathrm{EO}} = dMD$$

In RRL, each node will have to keep a new entry for each of the D destinations for each of l layers:

$$A_{\mathrm{RRL}} = lD$$

The addressing needs ratio for p-cycles with full cycle forwarding and RRL is

$$R_{\mathrm{FC/RRL}} = \frac{A_{\mathrm{FC}}}{A_{\mathrm{RRL}}} = \frac{dM}{lD}$$

and for early-off p-cycles and RRL:

$$R_{\mathrm{EO/RRL}} = \frac{A_{\mathrm{EO}}}{A_{\mathrm{RRL}}} = \frac{dMD}{lD} = \frac{dM}{l}$$

In our experiment setup we have $d = 4$, while Tab. I and Tab. II indicate $M \in (2.5, 9.4)$ for large p-cycles, $M \in (3.3, 4.1)$ for small p-cycles, and $l \approx 4$.

We see that the p-cycles multiplier M for large cycles grows somewhat with the network size. This value depends on the algorithm implementation—ideally, in a Hamiltonian cycle, we would have $M = 1$. On the other hand, creating good large p-cycles is computationally demanding. Our large-cycle heuristic already has a relatively high algorithmic complexity. While more efficient implementations may be possible, we believe that there is a trade-off between the large cycle M-values and the algorithmic complexity, and that M cannot be kept stable and low with an efficient large p-cycles heuristic.

For small cycles, the p-cycle multiplier M is barely affected by the network size. On the other hand, D grows with the network size, and is heavily dependent on the technology platform.

Ratios $R_{\mathrm{FC/RRL}}$ and $R_{\mathrm{EO/RRL}}$ indicate that, in large networks, p-cycles with full cycle forwarding has lower addressing needs than RRL, while the p-cycles with early-off and RRL have addressing needs of the same size order.

To estimate the forwarding state amount more accurately, the network technology platform would have to be taken into account. For example, in MPLS-based p-cycles each node in the cycle is used as the transport for additional 2(b–1) connections, severely affecting the state amount for large cycles. Furthermore, in moderately sized networks with efficient address aggregation, D would be small and $R_{\mathrm{FC/RRL}}$ could be close to 1.

B. Protection Success and Backup Path Lengths

In our experiments, both p-cycles and RRL show good dual-link failure protection ratios. As shown in Fig. 3, RRL finds a backup path in 93-96 % of all cases, while large p-cycles with early-off protects in 90-93 % of all cases. Early-off has a positive effect on the p-cycles protection rate, for large cycles in particular.

We observe that the network size has no major impact on the protection success, except for large p-cycles, where the

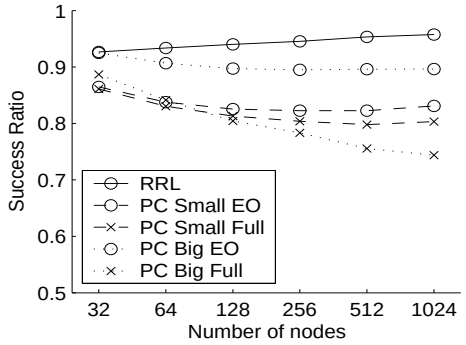

Fig. 3. Double link failure success ratio, semilogarithmic scale.

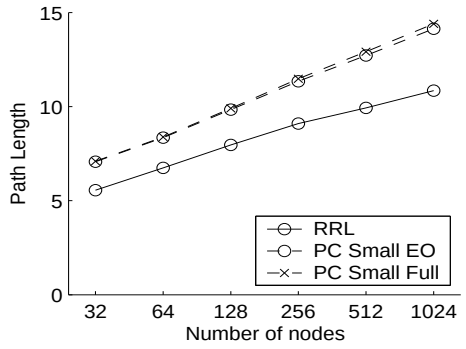

Fig. 4. Path lengths, semilogarithmic scale.

protection success slightly decreases as the network size is increased.

Small p-cycles backup path lengths grow logarithmically with the network size (Fig. 4), while RRL path lengths appear to grow sub-logarithmically. The early-off forwarding strategy has only a limited effect on the path lengths. Our experiments confirm that the large cycles have a disastrous effect on the path lengths, showing that the path lengths grow almost linearly with the network size (not shown in the figure).

V. Conclusion and Future Work

We have evaluated p-cycles and RRL over three metrics relevant for network-level fault protection. Our evaluation has shown that large p-cycles result in prohibitive backup path lengths. It is furthermore indicated that the p-cycles multiplier M in larger networks is higher for large cycles than for small cycles. Thus, the addressing needs and the state requirements will be higher for large cycles than for small ones. In general, our results clearly demonstrate the advantages of small p-cycles in existing network topologies. This result is particularly interesting in the light of recent research [7], which showed that creation of a Hamiltonian cycle is the most bandwidth-efficient strategy to build p-cycles if spare links can be added to the network.

The early-off exit strategy for p-cycles has shown less effect on the backup path lengths and the protection success ratio than we initially expected. In large networks with large cycles, it has a significant effect (Fig. 3), but we can say that this setup has only a theoretical relevance, due to unacceptable recovery path lengths. For small p-cycles, the early-off strategy has a minor positive effect. Keeping in mind that the early-off forwarding comes at the cost of either increased state amount or forwarding engine and packet header modifications, we doubt that it can be recommended in practice.

p-Cycles with full-cycle forwarding scales better with respect to addressing requirements than RRL. In moderately-sized networks, however, the additional state requirements need not be too different.

Protection success ratio under double link failures is roughly 10 % higher in RRL than in small p-cycles, and the backup path lengths are $\sim$25 % shorter. Clearly, this makes RRL a very attractive scheme for network protection.

RRL seems to have larger potential for state optimizations than p-cycles, and we are currently researching this claim. Effects of different protection schemes on data traffic in packet networks is an interesting direction for further research. We believe that RRL will distribute the traffic more equally, but this remains to be proven.

References

[1] P. Pan, G. Swallow, and A. Atlas, "Fast reroute extensions to RSVP-TE for LSP tunnels," Internet Draft, Aug. 2004, draft-ietf-mpls-rsvp-lsp-fastreroute-07.txt.

[2] P. Psenak, S. Mirtorabi, A. Roy, L. Nguen, and P. Pillay-Esnault, "MT-OSPF: Multi topology (MT) routing in OSPF," Internet Draft, Oct. 2004, draft-ietf-ospf-mt-00.txt.

[3] W. D. Grover and D. Stamatelakis, "Self-organizing closed path configuration of restoration capacity in broadband mesh transport networks," in *Proc. CCBR'98*, 1998.

[4] D. Stamatelakis and W. D. Grover, "IP layer restoration and network planning based on virtual protection cycles," *IEEE Journal on selected areas in communications*, vol. 18, no. 10, Oct. 2000.

[5] A. F. Hansen, T. Cicic, S. Gjessing, A. Kvalbein, and O. Lysne, "Resilient routing layers for recovery in packet networks," in *The International Conference on Dependable Systems and Networks (DSN)*, June 2005.

[6] I. Theiss and O. Lysne, "FROOTS - fault handling in up*/down* routed networks with multiple roots," in *Proceedings of HiPC'03*, 2003.

[7] A. Sack and W. D. Grover, "Hamiltonian p-cycles for fiber-level protection in homogeneous and semi-homogeneous optical networks," *IEEE Network*, vol. 18, pp. 49–56, Mar. 2004.

[8] D. A. Schupke, "The tradeoff between the number of deployed p-cycles and the survivability to dual fiber duct failures," in *IEEE ICC*, vol. 2, May 2003, pp. 1428–1432.

[9] D. A. Schupke, W. Grover, and M. Clouqueur, "Strategies for enhanced dual failure restorability with static or reconfigurable p-cycle networks," in *Proc. ICC*, vol. 3, June 2004, pp. 1628–1633.

[10] M. Clouqueur and W. D. Grover, "Availability analysis of span-restorable mesh networks," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, pp. 810–821, May 2002.

[11] A. Kvalbein, A. F. Hansen, T. Cicic, S. Gjessing, and O. Lysne, "Fast recovery from link failures using resilient routing layers," in *submitted to ISCC'05)*, June 2005.

[12] T. Cicic, A. F. Hansen, S. Gjessing, and O. Lysne, "Applicability of resilient routing layers for k-fault network recovery," in *Proceedings of ICN'05*, Apr. 2005.

[13] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: An approach to universal topology generation," in *IEEE MASCOTS*, Aug. 2001, pp. 346–353.

[14] B. M. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1617–1622, Dec. 1988.

[15] A. Fumagalli, I. Cerutti, and M. Tacca, "Optimal design of survivable mesh networks based on line switched WDM self-healing rings," *IEEE/ACM Transactions on Networking"*, vol. 11, no. 3, pp. 501–512, June 2003.