Finding Partially Disjoint Routes for Dual Fiber-Cut Protection on Bi-Connected Networks

Victor Yu Liu[1] **and Zhicheng Sui**[2]

[1]*Huawei Technologies, Boston, Massachusetts, USA* [2]*Huawei Technologies, Shenzhen, China*

yuliu@ieee.org, suizhicheng@huawei.com

Abstract: A diverse routing algorithm is introduced to maximize dual fiber-cut protection on bi-connected networks, where each flow have to use two or more partially disjoint backup paths to protect all restorable dual failures.

OCIS codes: (060.4257) Network, network survivability; (060.4261) Networks, protection and restoration

1. Introduction

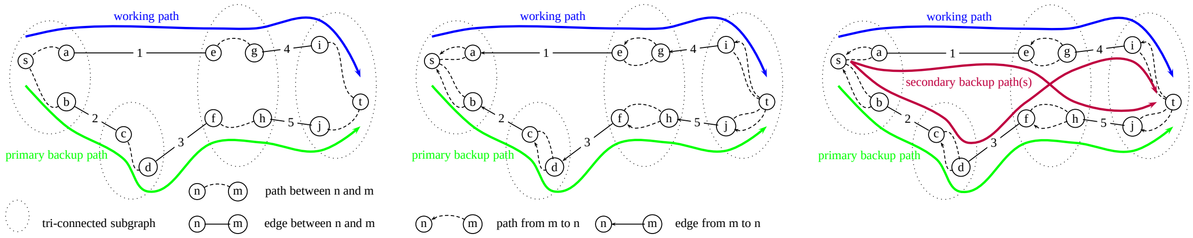
Protecting dual fiber-cuts receives attention in recent years due to large scale fiber deployments and reduced reliability in certain components, procedures and geographical regions.

Current research on network protection and restoration mostly focuses on resiliency approaches for single fiber-cuts and their derivatives such as failures of node or the shared risk link group (SRLG). Among them, the *shared backup path protection (SBPP)* could minimize network redundancy while still maintaining sufficient spare resource for backup paths upon failure scenarios, consequently guaranteeing network survivability with minimum resource. SBPP has been implemented in various backbone network protection and operation strategies. A recent study of using SBPP to protect dual fiber-cuts provides a fast and scalable method for the *spare capacity allocation* problem [4].

This paper studies the *diverse routing* problem using SBPP for dual fiber-cut protection on bi-connected networks. SBPP uses three mutually disjoint paths to protect all dual fiber-cuts. A tri-connected network can guarantee the existence of three disjoint paths for each node pair. However, majority of optical networks nowadays are still bi-connected. Certain flows will not have three mutually disjoint paths. Our purpose here is to find the best partially disjoint paths to protect the maximum subset of dual fiber-cuts.

On a tri-connected network, the *diverse routing* algorithm to find one working path and two disjoint backup paths can be achieved by using the max-flow based algorithms, i.e., the *Shortest Augmenting Path* algorithm in §7.4 of [1]. The method sets the link capacity of a tri-connected graph to one unit, and requests to send three unit of commodity from the source to the destination. After the algorithm find the solution, the used links will accommodate three link disjoint paths.

The max-flow based diverse routing algorithms have an advantage over the k-shortest path (KSP) based algorithms on finding all possible diverse paths when a condition called the *trap topology* exists. This has been explained for the single failure protection in [3]. This advantage can be extended to dual failure protection. An example is in Fig. 4(b). When flow 3–8 has the first two paths passing links 1-5 and 5-7, it is difficult to find another disjoint path using the KSP based algorithms. However, the max-flow based algorithms could still find the next disjoint path after removing these trap links overlapped by multiple paths. The detailed steps can be seen in [2].



(a) A set of cut-pair links (b) Residual graph by removing forward links (c) Add back forward links on cut-pairs

Fig. 1. Partially disjoint routing for dual fiber-cut protection on a bi-connected network

2. Cut-Pairs, Cut-Groups, and Secondary Backup Paths on Bi-Connected Networks

A bi-connected network includes a set of cuts containing only two fibers, called *cut-pairs*. The failure of a cut-pair will partition the network. Hence, the dual failures of any cut-pairs should be removed from the consideration of the partial protection here. The tri-connectivity algorithm in [5] provides a set of cut-pairs in linear time.

Given all cut-pairs in a bi-connected network, a flow could find two link disjoint paths first. One is working path, and the other is the primary backup path. Next, a subset of cut-pairs that straddle on these two paths can be identified. If there is no such cut-pairs, we can find a fully disjoint third path as the secondary backup path. If this subset of cut-pairs exist, they should be excluded from the dual fiber-cuts to be resilient by this flow. This can be done by finding one or more partially disjoint paths to reuse the first or second halves of these cut-pairs and to allow protection of the remaining dual fiber-cuts.

An example is given in Fig. 1. From node s to node t, there are one working and one primary backup path as shown in blue and red lines. They straddled on several cut-pairs: $(1,2),(1,3),(4,5)$. To use the augmenting shortest path algorithm to find the third path, we remove the forward directions on links alone the first two paths and leave reverse arrows on these links, as shown in Fig. 1(b). At this moment, the flow s–t is partitioned by cut-pairs. Next, we restore certain cut-pair edges so the partially disjoint secondary backup path can be found.

The subset of cut-pairs for flow s–t needs to be organized together as a set of *cut-groups*. Any two cut-pairs with a common edge in this subset can join together to form a cut-group . This "join" operation is transitive, i.e., if the first and second cut-pairs have a common edge, and the second and third cut-pairs share another common edge, all three cut-pairs should belong to the same cut-group of the given flow. In Fig. 1, an example of the cut-group is $[1],[2,3]$ where the first half of the cut-group is the set of link 1, and the other has the set containing two links: 2 and 3.

In Fig. 1(c), we can see that a single secondary backup path sometimes might not be able to protect all dual fiber-cuts after excluding all cut-pairs straddling the flow. For example, using the first purple path that passes link 1 and 5, the flow is still not protected for the dual fiber-cut of link 1 and 5, which does not belong to any cut-pairs. However, if both purple paths are used together, all non-cut-pair dual fiber-cuts can be protected.

Let k be the number of the cut-groups of a flow. For $k=1$, the minimum required number of partially disjoint secondary backup paths p can be zero, one or two as shown in Fig. 2 respectively.

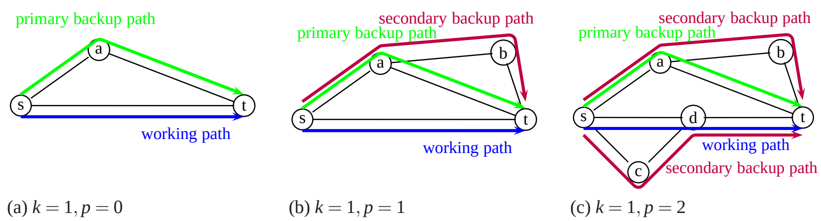


Fig. 2. Special cases when there is one cut-group, $k=1$

Theorem 1. *The minimum required number of partially disjoint secondary backup paths to protect a flow against dual fiber-cuts in bi-connected networks (excluding all cut-pairs) is the number of cut-groups of the flow when there are two or more cut-groups.*

Proof. We need to prove $p=k, k\geq 2$ by induction below.

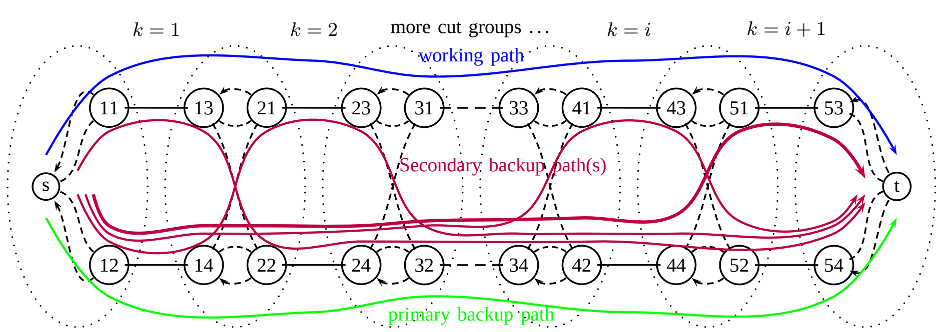


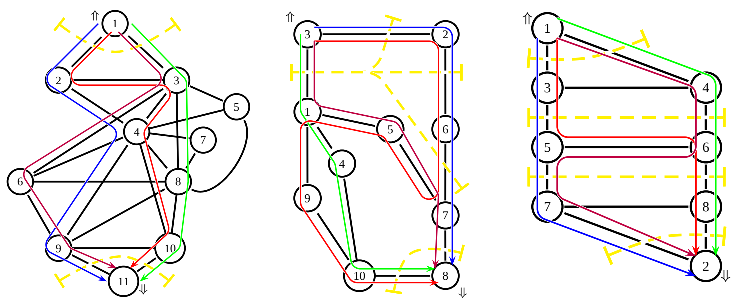
Fig. 3. Construct partially disjoint routes from $k=i$ to $k=i+1$ during the induction

When $k = 2$, we design these two secondary backup paths to use different edge sets combination from the two cut-groups, different from those used in the working and the primary backup paths, then all the dual fiber-cuts excluding those cut-pairs can be protected. The example is illustrated in Fig 1(c).

Assume $k = i$ and we have $p = i$, where $i = 2, \ldots$, there are already i secondary backup paths to restore all dual fiber-cuts excluding all cut-pairs inside the first i cut-groups. We construct these i paths as follows: the j-th secondary backup path always uses the first half (top) of the j-th cut-group and the second half (bottom) of other cut-groups, where $j = 1..i$. It is easy to prove that these paths could restore any dual fiber-cuts involving the first i cut-groups. This is shown in Fig. 3. When $k = i+1$, we will use $(i+1)$-th secondary backup path to protect additional dual fiber-cuts introduced by the last cut-group. First, we make all previous i secondary backup paths to use the second half (bottom) of the last cut-group. Then we add one extra secondary backup path using the first half (top) of the last cut-group, but the second half (bottom) of all other i cut-groups, as shown in the bold purple line in Fig. 3. This path will protect the flow against the dual fiber-cuts involves the un-protected cases in previous i paths. Consequently, $p = i+1$. By induction, we have $p = k, \forall k, k \geq 2$. □

3. Test Results

A set of bi-connected networks used to test our partial diverse routing algorithm is given in Fig. 4. The first network in Fig. 4(a) is a New Jersey LATA network. The second has a trap topology for dual fiber-cuts. It also includes a cut-group with multiple cut-pairs. The third network contains four cut-groups for flow $1 \to 2$. Sample flows for test networks, their cut-groups and four partially disjoint paths are shown in different colors. Only two secondary backup paths are computed for demonstration. For a flow with more than two cut-groups, i.e., flow $1 \to 2$ on the third network, these two secondary backup paths are selected to maximize the restorable dual fiber-cuts, when $p = k = 4$, four secondary backup paths (plus working and primary backup paths) are normally considered too many for most real deployments.



Network	(a) NJ LATA	(b) Trap Topology	(c) Multiple Cut-groups
Flow (⇑→⇓)	$1 \to 11$	$3 \to 8$	$1 \to 2$
Cut-groups (yellow)	[1-2],[1-3]; [9-11],[10-11]	[3-2 2-6 6-7],[3-1]; [7-8],[10-8]	[1-3],[1-4]; [3-5],[4-6]; [5-7],[6-8]; [7-2],[8-2]
Path 1 (blue)	1 2 4 9 11	3 2 6 7 8	1 3 5 7 2
Path 2 (green)	1 3 8 10 11	3 1 4 10 8	1 4 6 8 2
Path 3 (red)	1 2 3 4 10 11	3 2 6 7 5 1 9 10 8	1 3 5 6 8 2
Path 4 (purple)	1 3 6 9 11	3 1 5 7 8	1 4 6 5 7 2

Fig. 4. Bi-connected networks and their partially disjoint routes

References

1. R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, New York, 1993.
2. R. Bhandari. *Survivable Networks Algorithms for Diverse Routing*. Kluwer Academic Publishers, 1999.
3. D. A. Dunn, W. D. Grover, and M. H. MacGregor. Comparison of k-shortest paths and maximum flow routing for network facility restoration. *IEEE Journal on Selected Areas of Communications*, 2(1):88–99, 1 1994.
4. V. Y. Liu and D. Tipper. Spare capacity allocation using shared backup path protection for dual link failures. In *International Workshop on Design of Reliable Communication Networks (DRCN)*, Oct. 10–12, 2011.
5. T. H. Tsin. Yet another optimal algorithm for 3-edge-connectivity. *Journal of Discrete Algorithms*, 7:130–146, 2009.