Diverse Routing for Shared Risk Resource Groups (SRRG) failures in WDM Optical Networks *

Pallab Datta and Arun K. Somani
Dependable Computing & Networking Laboratory
Department of Electrical & Computer Engineering
Iowa State University, Ames, Iowa 50011
E-mail:{pallab, arun}@iastate.edu

Abstract

Failure resilience is one of the desired features of the Internet. Most of the traditional restoration architectures are based on single-failure assumption which is unrealistic.

Multiple link failure models, in the form of Shared-Risk Link Groups (SRLG's) and Shared Risk Node Groups (SRNG's) are becoming critical in survivable optical network design. We classify both these form of failures under a common heading of shared-risk resource groups (SRRG) failures. In our research, we propose graph transformation techniques for tolerating multiple failures arising out of shared resource group (SRRG) failures.

*Diverse Routing in such multi-failure scenario essentially necessitates finding out two paths between a source and a destination that are SRRG disjoint. The generalized diverse routing problem has been proved to be **NP-Complete**. The proposed transformation techniques however provides a polynomial time solution for certain restrictive failure sets. We study how restorability can be achieved for dependent or shared risk link failures and multiple node failures and prove the validity of our approach for different network scenarios.*

1. Introduction

WDM optical networks have evolved as the primary transport medium in modern day networks. Customers expect to see uninterrupted service, even in the event of failures such as power outages, equipment failures, natural disasters and cable cuts. Many optical-layer protection schemes for WDM networks have been proposed in the literature [1][2]. Protection schemes can be classified either as *link protection* or *path protection* based on the initialization locations of the re-routing process. Link protection schemes route a connection around a failed link. In case of a failure, the nodes connected to the failed link routes the connection around the failed link to the neighboring node of the original path. Path protection attempts to provide a backup path from the source to the destination that maybe independent of the working path. Path-based protection has been established to be the more capacity-efficient approach for mesh based networks as compared to link based rerouting schemes [1][3]. Hence the protection model assumed in our work is *path protection*.

In order to provide end-to-end path based restoration, for each demand the network is required to provide two diverse paths: the service path and the restoration path. When the service path fails, the traffic gets re-routed to the restoration path. There are two commonly used protection schemes: shared path protection and dedicated path protection. In case of shared path protection, spare capacity is shared among different protection paths, while in dedicated path protection, the spare capacity is dedicated to individual protection paths. Shared path protection, although more difficult to implement, have been proven to be more capacity efficient than dedicated path protection.

The diverse routing problem is to find two paths between a pair of nodes in the optical layer such that no single failure in the physical layer may cause both paths to fail. The problem of finding two diversely routed paths in optical networks is much more difficult than the traditional edge/node disjoint path problem in graph theory [8][14].

Instances where separate fiber optic links share a common failure structure is often referred to as SRLG (Shared-Risk Link Group) [4][5]. Two examples of such shared-risk link groups are shown in Fig. 1, which illustrates two diverse fiber links which may be placed in the same conduit at the physical layer and are subject to a single point of fail-

*The research reported in this paper is funded in part by the National Science Foundation under grant ANI-9973102, Defense Advanced Research Projects Agency and National Security Agency under grant N66001-00-1-8949, Junkins Professorship and NSF CNS 0323374.

ure. In this paper, we consider the diverse routing problem within a generalized framework where SRLG's are used to represent a set of optical links that are affected by a single failure in the physical layer. Finding a pair of diverse paths in the optical layer translates to finding a pair of SRLG-diverse paths.

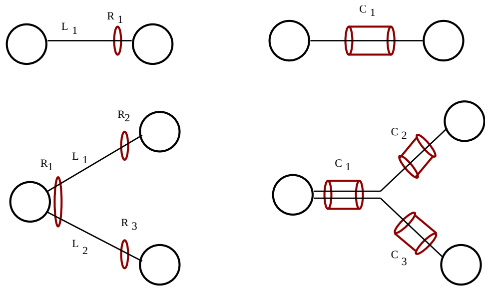


Figure 1. Shared-Risk Link Groups and their corresponding physical routes.

The diverse routing problem have been shown to be NP-Complete in [6], a result that has been conjectured by several other researchers in [7][8][14]. Recent studies have proven the NP-completeness of the generalized SRLG diverse routing [6][7][9][10], a special case of the diverse routing problem. The two sub-problems, the least coupled SRLG path problem and the minimum cost SRLG diverse routing problem has been also shown to be NP-Complete [6]. The routing problem under both wavelength capacity and path length constraints has been also shown to be NP-complete [12].

In [7][14], different heuristic approaches have been studied for diverse routing in presence of SRLG's. One of the common problems that arises in restoration path computation is the existence of a *trap topology*. With a trap topology, if a service path is independently routed over a trap topology, then there may not exist a diverse restoration path, even though two diverse paths exist in the network. Different heuristics such as Active Path First (APF), which finds an active path first, followed by an SRLG-disjoint backup path, is a simplistic viable technique but it sometimes leads to trap scenarios [13]. Alternative solutions for avoiding trap scenarios in shared risk link disjoint routing have been proposed in [7][13].

In this paper, we address the problem of diverse routing in SRLG situations as well as multiple failures arising out of nodes sharing a common risk of failure. We classify both these sub-problems under a generalized heading of shared risk resource group (SRRG) routing. In this paper we propose a polynomial time graph transformation heuristic for solving a sub-set of the generalized version of the diverse routing problem in networks with shared risk resource groups. We analyze the complexity of these routing methodologies and also validate the correctness of these algorithms, thus making it feasible to be applied to large networks with huge traffic demands.

1.1. Outline of the paper

The remainder of the paper is organized as follows: Section II will give a brief description of shared risk resource groups (SRRG's). Section III describes the graph transformation techniques for diverse SRLG routing. Section IV performs the complexity analysis of the SRLG disjoint routing methodology. Section V discusses about diverse routing in SRNG scenarios. Section VI presents the computational complexity for SRNG disjoint routing. We conclude the paper in Section VII.

2. Shared-Risk Resource Groups (SRRG's)

In traditional networks the importance of protection against failures arising out of shared risk resource groups (SRRG) is increasing, thus motivating us to study different fault tolerant mechanisms. One of the classes of SRRG's comprises of multiple links sharing a common component whose failure causes failure of all links in that group. One such common class of components include ducts, or conduits through which multiple independent logical links are routed in the ground. Any physical failure of one of these ducts can invoke a logical failure of multiple links as illustrated in Fig. 2(a). A single link can be part of more than one SRLG. As shown in Fig. 2(a) the link connecting nodes 1 and 3 is part of two SRLG's R_1 and R_2. In our research, we concentrate on *co-incident SRLG's* [4], which are groups incident on a common node.

Another instance of a shared-risk resource failure (SRRG), is the failure of two or more nodes that are connected by a common channel or link and are often referred to as shared risk node groups (SRNG's) (Fig. 2(b)) . In practice such a failure of one or more nodes may be due to some malicious signal which corrupts the transmission (laser's) at both the end-points of the link or maybe due to some power outage in an area, leading to simultaneous failure of some nodes in the network.

2.1. Graph Transformation Technique for Diverse SRRG Routing

In the following sections we present graph transformation techniques for finding out two shared-risk group disjoint paths to tolerate shared-risk resource group failures. We assume in our work that there will be at most one SRRG failure at any given time.

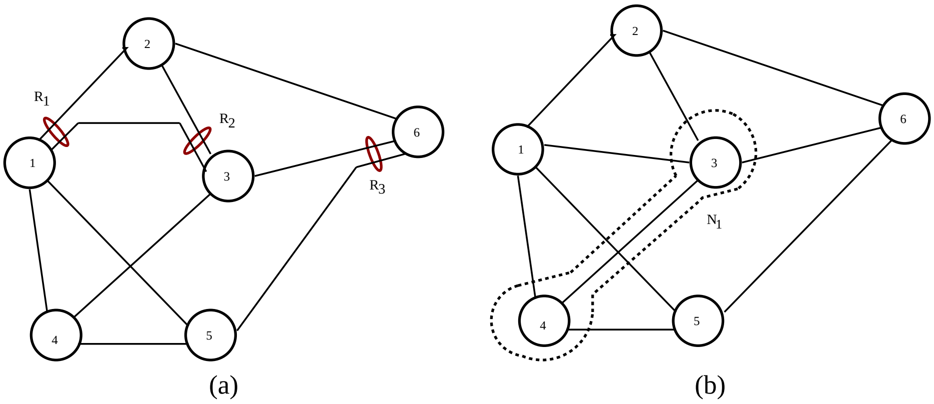


Figure 2. Network With (a) Shared-Risk Link Groups (SRLG's) and (b) Shared Risk Node Groups (SRNG's).

2.2. Notations

The following are some of the notations that we would be consistently using in the following sections for describing the graph transformation algorithms.

- *G = (V, E)*: Directed graph G, where *V* is the set of vertices and *E* is the set of edges.
- α_e: Weight of an edge *e* in the directed graph G. Assuming bi-directional links, we have $\alpha_{uv} = \alpha_{vu}$.
- K_L:
  Total number of Shared-Risk Link groups(SRLG's) in the network.
- R_i: The i^{th} Shared Risk Link Group. $i = 1 \cdots K_L$.
- K_N: Total number of Shared-Risk Node groups(SRNG's) in the network.
- N_j: The j^{th} Shared Risk Node Group. $j = 1 \cdots K_N$
- d_i: A dummy vertex representing the i^{th} shared risk link group or a shared risk node group.
- *N(v)*: The neighborhood of the vertex *v*.

3. SRLG Diverse Routing

In this section we study a graph transformation technique to tolerate failure scenarios arising out of shared-risk link groups (SRLG's). The initial graph comprising of shared-risk link groups is shown in Fig. 2(a). Let the original graph be represented by *G = (V, E)*, where each vertex $v \in V(G)$ and each edge $e \in E(G)$. Let the weight of each edge *e* be denoted by α_e. Let the total number of SRLG's in the network be denoted by K_L. The following are the assumptions under which the proposed tranformation technique yields a polynomial time solution.

- There can be any number of shared-risk link groups in a network.
- Each shared risk link group size is a smaller than the degree of the node on which the group is incident.
- An edge can be shared between utmost two shared risk link groups.

Each shared-risk link group R_i can be represented as $R_i \subset E(G)$. Let the shared-risk link groups in the network be $R_1 \cdots R_{K_L}$. We introduce *dummy vertices* d_i, for each shared-risk link group in the transformed graph *G' = (V', E')*. We derive a new graph *G'* from *G* by following the vertex transformation : $V'(G') := V(G) \cup \{d_1 \cdots d_{K_L}\}$. Hence in the transformed graph *G'*, the total number of vertices $|V'|$ is given by $|V'| = |V| + K_L$ as shown in Fig. 3. The shared-risk link groups information is stored in the following data structure :

$$R_1 : \{uv : uv \in R_1\}$$
$$R_2 : \{uv : uv \in R_2\}$$
$$R_i : \{uv : uv \in R_i\}$$
$$\cdots\cdots$$
$$R_{K_L} : \{uv : uv \in R_{K_L}\}$$

The shared-risk link groups incident on node 1,3 and 6 in the initial graph in Fig. 2(a) are stored as $R_1 : \{(1,2)\ (1,3)\}$, $R_2 : \{(3,1)\ (3,2)\}$ and $R_3 : \{(6,3)\ (6,5)\}$. It is to be noted that the edge (1,3) is identical to the edge (3,1), since each link is assumed to be a bi-directional.

We start scanning all the edges belonging to the shared-risk link groups. Let R_i be comprised of the set of edges R_i = $\{uv_1, uv_2 \cdot\cdot, uv_k\}$. If none of these edges in R_i belongs to any other shared-risk link group, then we modify these edges in the transformed graph *G'*, according to the rule, *G'* = $G\backslash$ e, s.t. e $\in R_i$ and introduce a new set of edges ud_i, $\{d_i v_1, d_i v_2, \cdot\cdot d_i v_k\}$. The new edge ud_i in the transformed graph *G'* has a weight of zero and the edges $d_i v_1 \cdot\cdot d_i v_k$ from the dummy node to the vertices $v_1 \cdot\cdot v_k$ on the transformed graph gets the original weight α_e of the edges $e \in R_i$.

If however any one of the edge $uv_1, uv_2 \cdot\cdot, uv_k \in R_i$, R_j, $j \neq i$, i.e. the edge is part of more than one shared-risk link group, the transformation follows by connecting the dummy nodes representing the link groups in which this edge belongs, and connecting the two dummy nodes to the two end-vertices of the common edge, to show the commonality of the shared link between the two groups. Delete this edge *uv* from the shared-risk groups R_i and R_j. The edges between the dummy nodes $d_i d_j$ in the transformed graph *G'* gets a weight of α_{uv} of the edge $uv \in R_i$, R_j and the edge ud_i assumes an edge weight of zero and the edge $d_j v$ from the dummy node to the end vertex assumes an edge weight of $\alpha_H \geq \{max(\alpha_i), 1 \leq i \leq |E|\}$. Once an edge is transformed from the graph *G* to *G'*, it is not considered for further transformation. We scan through all the remaining edges in the graph, and the edges that doesn't belong to any shared risk link group R_i, remains untempered in the transformed graph *G'*. The choice of a higher edge weight α_H is primarily to avoid selection of an edge that is shared amongst multiple shared risk link groups.

As can be seen in Fig. 3, there is a link between dummy nodes d_1 and d_2 to represent the original link between nodes 1 and 3. The edge *e'* connecting d_1 and d_2 is replaced by the weight α_{uv}, of the edge *uv* which is common to both the shared-risk link groups. It is to be noted that, in such instances when a link is common to two shared risk link groups, failure of any link in one group doesn't propagate to the other group. For example in Fig. 2(a), failure of link $1 \rightarrow 2$ doesn't imply failure of link $2 \rightarrow 3$.

In Fig. 3 the dummy nodes d_1, d_2 and d_3 are introduced to represent the three shared-risk link groups R_1, R_2 and R_3. In order to find out two shared-risk group disjoint routes between any s-d pairs, we apply the edge-disjoint shortest-cycle algorithm [14] on the transformed graph *G'* to find the two group-disjoint routes for a given s-d pair. For example the two group-disjoint paths between nodes $1 \rightarrow 6$ in the original graph *G = (V, E)* can be possibly the two paths corresponding to either of the following cycles in the transformed graph *G'*: C_1: {1-d_1-2-6, 1-4-3-d_3-6} or C_2: {1-d_1-2-6, 1-4-5-d_3-6} or C_3: {1-d_1-2-6, 1-5-d_3-6} as shown

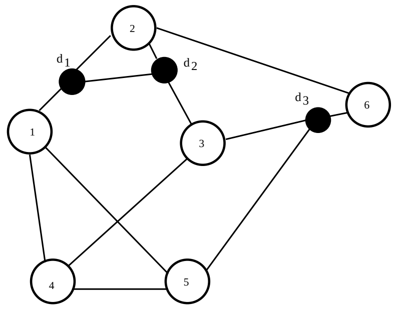
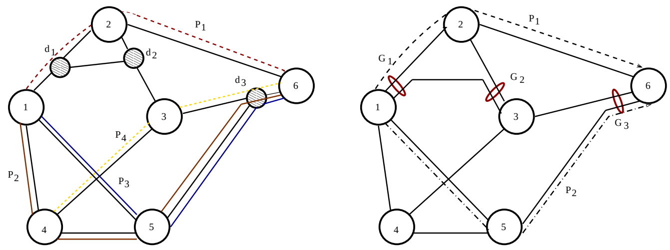


Figure 3. Graph Transformation using Dummy Nodes for Diverse SRLG Routing.



(a) (b)

Figure 4. (a) SRLG Disjoint Routes on transformed Graph G', (b)SRLG Disjoint Routes on the Original Graph G.

in Fig. 4 (a). It will be shown that the two paths derived out of the shortest-cycle on the transformed graph *G'* actually gives us two SRLG disjoint routes.

Once we obtain the shortest-cycle on the transformed graph *G'*, we need to map these two routes on the original graph *G*. The reverse transformation of the routes can be done either in the forward direction, i.e. starting from the source towards the destination or in the reverse direction, i.e. initiating from the destination towards the source. Now the paths contained in the cycles can be either in the form of *P*: $(s - \cdot\cdot d_i - x - y \cdots - d)$ or *P'*: $(s - \cdot\cdot d_i - d_j - x - y \cdots - d)$. In the first case the final path on the original graph *G* can be obtained by the reverse transformation of the edges of the form ud_i-$d_i v$ to the edge *uv*. This is obtained by dropping the dummy node d_i from the path *P*. In the second case the final path on the graph *G* is obtained by the reverse transformation of the edge $d_i - d_j$ by the edge *uv* in the original graph *G* such that $uv \in E(G)$, $uv \in R_i \cap R_j$ and ud_i, $d_j v \in E'(G')$. The detailed algorithm is not presented here due to space limitations and can be found in [16].

As an example, if we choose the cycle C_3 in Fig. 4(a) to be the shortest-cycle (assuming equal weights of all links in the original graph *G*) between nodes 1→6, then the two routes, obtained from the cycle are P_1: 1-2-6 and P_2: 1-5-6 as shown in Fig. 4(b).

Lemma: If there exists an edge-disjoint shortest cycle in the transformed graph *G'*, then using the above transformation, the paths P_1 and P_2 comprising the cycle can be always mapped to two SRLG disjoint routes in the original graph *G*.

If there exists an edge-disjoint cycle in the transformed graph *G'* then the degree of each vertex *v* is greater than the size of the group incident at that vertex. If P_1 and P_2 comprises the two paths of the edge-disjoint shortest cycle in the transformed graph *G'*, then each dummy vertex d_i can appear utmost once in either of the paths P_1 or P_2.

Thus the paths P_1 and P_2 is either in the form of *P*: ($s-\cdots-x-d_i-x'-y\cdots-d$) or *P'*: ($s-\cdots-u-d_i-d_j-v-w\cdots-d$). In case 1, there is an exact mapping of the edges $u-d_i-v$ in the transformed graph *G'* to the edge *uv* in the original graph *G*. In case 2, the edge d_i-d_j in the transformed graph exactly maps to the edge *uv*, such that $uv \in E(G)$, $uv \in R_i \cap R_j$ and ud_i, $d_jv \in E'(G')$. Moreover since both paths do not include overlapping d_i's, the mapped routes on the original graph are always SRLG disjoint.

4. Complexity Analysis of SRLG Routing

This section evaluates the overall complexity of the SRLG disjoint routing algorithm using the graph transformation technique as described in the previous section. The computational complexity can be broken up into three parts, one the complexity involved in the transformation of the graph, second the complexity of finding out edge-disjoint shortest cycles in the transformed graph and finally the complexity of mapping the paths obtained on the transformed graph to the paths on the original graph.

The total number of shared-risk link groups in the network is given by K_L and each group can possibly have a *maximum* of *E* edges. Another additional constraint limits the existence of any edge in not more than two groups. Thus as can be seen from the algorithm presented for graph transformation, the complexity involved in determining whether an edge belongs to more than one shared-risk link group involves an exhaustive search, which is computationally of the order of O ($E \cdot K_L$). Depending on the outcome of this decision, the edges are transformed following the algorithm presented above. The detailed algorithm is presented in [16]. Thus the overall complexity of the graph transformation is O ($E \cdot K_L$).

The transformation of the graph from $G \rightarrow G'$ adds K_L number of nodes. Two paths that are shared-risk group disjoint are computed using the shortest-cycle algorithm on the transformed graph as described in [14]. The computational complexity of the shortest-cycle algorithm is given by O ($|V|^2 + |E|^2$). We have two distinct cases, one in which the SRLG's are such that there is no edge which belongs to more than one group and another scenario where an edge can possibly belong to two shared risk groups. In the first case, when each edge belongs to only one SRLG, the additional number of edges introduced in the transformed graph *G'* is K_L. Hence the complexity of finding two SRLG disjoint paths would be O ($(V+K_L)^2 + (E+K_L)^2$) $\approx$ O ($(E+K_L)^2$), assuming that the number of edges in a mesh network is much larger than the number of vertices.

However if an edge belongs to two SRLG groups, e.g.as shown in Fig. 2(a), then the number of additional edges in this transformed graph *G'* is equal to K_L. The overall complexity of finding two SRLG disjoint routes in this transformed graph is O ($(V+K_L)^2 + (E+K_L)^2$) $\approx$ O ($(E+K_L)^2$), assuming that number of edges in the graph supersede the total number of vertices.

Once the edge-disjoint shortest cycle is computed on the transformed graph, the paths comprising the cycle, is mapped back on the original graph *G*. This requires a computational overhead of O ($E + K_L$) since a path can have a maximum of ($E + K_L$) edges in a transformed graph *G'*. Combining the three above complexities, the overall complexity of the diverse SRLG routing is given by the dominant term, which is the complexity involved in finding the edge-disjoint shortest cycle on the transformed graph. Thus the overall complexity is given by O ($(E+K_L)^2$).

5. SRNG (Shared Risk Node Group) Diverse Routing

In this section we develop a graph transformation technique for finding out two routes in scenarios where more than one node shares a common risk of failure. We identify such scenarios where more than one node shares a common risk of failure as shared risk node groups (SRNG's). For simplicity we assume that the size of each SRNG is limited to two adjacent nodes sharing an edge between them.

Let the original graph be represented by *G = (V, E)*. Let the total number of SRNG's in the network be denoted by K_N. Each shared node risk group N_i can be represented as $N_i \subset V(G)$. The two vertices *u*,*v* of each node risk group N_i in the original graph *G* is contracted to a single *dummy vertex* d_i in the transformed graph *G' = (V', E')*.

In a weighted graph *G*, we define a graph *G'* by contraction of nodes and edges as follows: replace vertices *u* & *v* by a single dummy vertex d_i and delete all edges incident to *u* & *v* and introduce new edges from d_i to $N(u) \cup N(v)$, where *N(u)*, *N(v)* stands for the neighborhood of the vertices *u* & *v*. The weights of the unchanged edges remain the same in the transformed graph *G'*. The modified edges during the

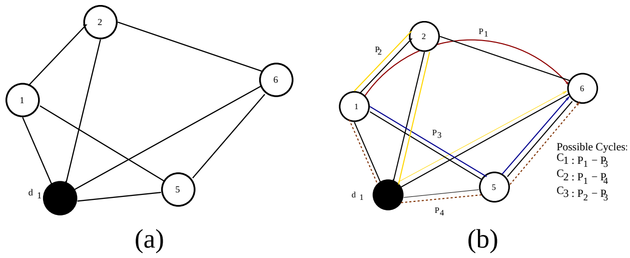


Figure 5. (a) Transformed Graph Indicating Shared Node Risk Groups,(b) SRNG Disjoint Routes on the transformed Graph G'.

transformation assumes weights as shown by the following function:

$$\alpha_{d_i x} = \begin{cases} \alpha_{ux} & if\ x \in N(u) \setminus N(v) \\ \alpha_{vx} & if\ x \in N(v) \setminus N(u) \\ \alpha_{ux} + \alpha_{vx} & \text{otherwise} \end{cases} \quad (1)$$

Let edge-set E_s be defined as the set of edges from all such vertices, which has edges to only one of the nodes of the node-risk group, and critical edges C_s, the set of edges between the vertices *u*,*v* of all the node-risk groups. All the edges of the edge-set E_s gets deleted i.e. *G' = G*\ e, where e $\in E_s$ and are replaced by new edges between *x* and the *dummy node* d_i. During this transformation the edge xd_i assumes the weight of the original edge *ux* or *vx* as shown in Equation 1.

In case of vertices *x* where $x \in$ *N(u)*, *N(v)* the edges *ux*, *vx* transforms to a single edge xd_i of weight ($\alpha_{ux} + \alpha_{vx}$) as shown in Equation.1. Let the shared-node risk groups be N_1,$N_2 \cdot\cdot N_{K_N}$. Let d_1,$d_2 \cdot\cdot d_{K_N}$ be the dummy vertices representing these SRNG's. The overall graph transformation can thus be represented as *V'(G')=V(G)*- $\cup_{i=1}^{|R|}$ $\{v_i\}$ + $\{d_1$,$d_2 \cdot\cdot d_{K_N}\}$, where $v_i \in N_i$. The edge transformations are defined above and in Equation 1.

As shown in Fig. 5(a), the links between nodes 2→3, 1→3, 1→4, 6→3 and 5→4 are replaced by modified edges with adjusted edge weights according to Equation 1. Finding out two routes for a given s-d pair in the original graph *G*, that are node group disjoint is equivalent to finding the node-disjoint shortest-cycle [14] in the transformed graph *G'* as shown in Fig. 5(b). Infact finding out a node disjoint shortest cycle on the transformed graph ensures that the two paths comprising the cycle, can guarantee fault-tolerance against failure of any single node on the path and also from node groups present in the topology. For example the two node disjoint paths between nodes 1 → 6 can be possibly the two paths comprising either of the following cycles in the transformed graph *G'*: C'_1: {1-2-6, 1-5-6} or C'_2: {1-2-6, 1-d_1-6} or C'_3: {1-2-6, 1-d_1-5-6} and C'_4: {1-2-d_1-6, 1-5-6} as shown in Fig. 5(b).

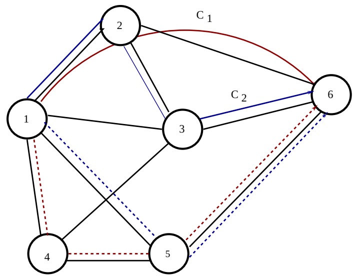


Figure 6. Shared Node Risk Group Disjoint Routes on the Original Graph G.

After the node-disjoint shortest cycle is found on the transformed graph, the paths corresponding to the cycle is mapped back to obtain the two node-group disjoint routes on the original graph *G*. The fact that the routes in *G'* are node-disjoint automatically guarantees node group disjoint routes in the original graph *G*.

If there exists a node-disjoint cycle in the transformed graph *G'* then each dummy vertex d_i appears utmost once in either of the paths P_1 or P_2. Thus the paths P_1 and P_2 are either in the form *P*: ($s - \cdot\cdot - u - d_i - v - \cdot\cdot\cdot - d$) or *P'*: ($s - \cdot\cdot - u - d_i - d_j - v - \cdot\cdot\cdot - d$).

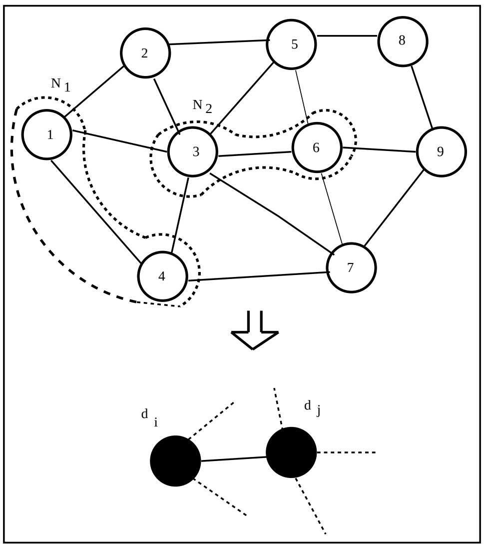


Figure 7. Example showing links between two dummy nodes and the reverse transformation.

Case 1: In the reverse mapping of the path *P* from *G'* → *G*, we need to transform the edges *u*-d_i-*v* to obtain the corresponding edges in the original graph *G*. It can be obtained by the matching of vertices belonging to the neighborhoods of both the vertices of the shared node risk group.

We scan the neighborhood's of both the vertices *u* & *v* and select vertex *w* such that $w \in$ *N(u), N(v)* & $w \in N_i$. If there are more than one choice of such a vertex, the one that *minimizes* α_{uw} + α_{wv} is selected. The mapped edges in the final path hence becomes *u-w-v*. If no such vertex *w* exists, then select vertices *w* & *w'* such that $w \in$ *N(u)*, N_i and *w'* $\in$ *N(v)*, N_i. The mapped edges in the final path hence becomes *u-w-w'-v*, where the edge *ww'* connects the two nodes of the shared node risk group N_i.

Case 2: In the reverse mapping of the path *P'* from *G'* $\rightarrow$ *G*, the same technique described in *Case 1* can be used to map the path *u*-d_i-d_j-*v* in the original graph. The edge d_i-d_j appears in *G'* if there are two overlapping SRNG's (described in more detail in a later section), or if the path *P'* contains vertices belonging to two groups joined by a bridge edge *e* as shown in Fig. 7. Assuming non-overlapping SRNG's, let *e'*, *e"* denote the critical edges connecting the vertices in the node-groups N_i and N_j respectively. The reverse mapping of the path *u*-d_i-d_j-*v* hence leads to a selection of vertices *w*, *w'* such that $w \in$ *N(u)*, N_i and *w'* $\in$ *N(v)*, N_j, and corresponding edges which satisfy that one of α_{uw} + α_e + $\alpha_{w'v}$ or α_{uw} + $\alpha_{e'}$ + α_e + $\alpha_{w'v}$ or α_{uw} + α_e + $\alpha_{e''}$ + $\alpha_{w'v}$ or α_{uw} + $\alpha_{e'}$ + α_e + $\alpha_{e''}$ + $\alpha_{w'v}$ is *minimized*, depending on the choice of vertices.

For example if the cycle C_3' is chosen, then the paths on the original graph would be 1-2-6 and 1-4-5-6 (and not 1-3-6 because, there is no link between 3$\rightarrow$5). However if the cycle C_4' is chosen, then the two routes are 1-2-3-6 and 1-5-6. This is shown in Fig. 6(b). The detailed algorithm for finding out two node group disjoint routes are presented in [16].

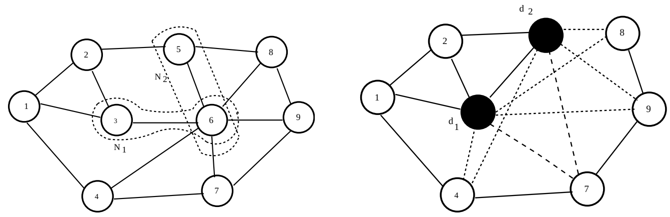
Lemma: In a graph with non-overlapping SRNG's, if there exists a node-disjoint shortest cycle in the transformed graph *G'*, then using the above reverse mapping, the paths P_1 and P_2 composing the cycle always maps to two SRNG disjoint routes in the original graph *G*.

Since each *dummy node* representing a shared risk node group can appear utmost once in any one of the two routes P_1 or P_2 and since these routes are node-disjoint, and the graph doesn't have any overlapping SRNG's, transformation of these routes on the original graph *G* automatically guarantees that they are SRNG disjoint.

The SRNG's can give rise to another scenario, where a vertex *v* is part of more than one SRNG, i.e. $v \in N_i \cap N_j$, $i \neq j$. This scenario is demonstrated in Fig. 8(a). In this figure, node 6 is part of both SRNG's N_1 and N_2.

The two vertices *u,v* of each node risk group N_i in the original graph *G* is collapsed to a single *dummy node* d_i in the transformed graph *G' = (V', E')*.

In a weighted graph *G*, we define a new graph *G'* by



(a) (b)

Figure 8. (a) Situations where a node belongs to more than one SRNG, (b)Graph Transformation in case of nodes belonging to more than one SRNG.

collapsing nodes and contracting edges as follows: replace vertices *u* & *v* by a single dummy vertex d_i and delete all edges incident to *u* & *v* and introduce new edges from d_i to *N(u)* $\cup$ *N(v)*, where *N(u)*, *N(v)* stands for the neighborhood of the vertices *u* & *v*. Let *v* be the vertex that is part of more than one SRNG, i.e $v = N_i \cap N_j$, $i \neq j$. In the transformed graph *G'* we have an edge connecting the two dummy nodes d_1 and d_2 to represent this overlapping SRNG, as shown in Fig. 8(b). The weights of the unchanged edges remain the same in the transformed graph *G'*. The modified edges during the transformation assumes the weights as shown by the function below

$$\alpha_{d_i x} = \begin{cases} \alpha_{ux} & if\ x \in N(u) \setminus N(v)\ \&\ u \notin N_i \cap N_j \\ \alpha_{vx} & if\ x \in N(v) \setminus N(u)\ \&\ v \notin N_i \cap N_j \end{cases} \quad (2)$$

$$\alpha_{d_i x}, \alpha_{d_j x} = \begin{cases} \alpha_{ux} & if\ u \in N_i \cap N_j \\ \alpha_{vx} & if\ v \in N_i \cap N_j \end{cases} \quad (3)$$

and $\alpha_{d_i d_j} = 0$.

Let edge-set E_s be defined as the set of edges from all such vertices, which has edges to only one of the nodes of the node-risk group, and critical edges C_s, be the set of edges between the vertices *u,v* of all the node-risk groups.

All the edges of the edge-set E_s gets deleted i.e. *G' = G*$\setminus$ e, where e $\in E_s$ and are replaced by new edges between *x* and the *dummy node* d_i. During this transformation the edge xd_i assumes a weight according to the function as shown in Equation 2 & 3.

During the graph transformation, all edges $\{uv$: $u \notin N_i$, $v \in N_i \cap N_j$, $i \neq j\}$ are transformed to the edges ud_i and ud_j, thus giving us the choice of selection of dummy vertices in the transformed graphs *G'* as shown by the dotted lines in Fig. 8(b). The weight of these transformed edges are given in Equation 2 & 3. For example, in Fig. 8(b) the vertices

4,7,8 and 9 are connected to both the dummy vertices d_1 and d_2.

Let the shared-node risk groups be N_1,$N_2 \cdot \cdot N_{K_N}$. Let d_1,$d_2 \cdot \cdot d_{K_N}$ be the dummy vertices representing these SRNG's. The overall graph transformation can thus be represented as *V'(G')=V(G)-* $\cup_{i=1}^{|R|}$ $\{v_i\}$ + $\{d_1, d_2 \cdot \cdot d_{K_N}\}$, where $v_i \in N_i$. The edge transformations are defined above in Equations 2-3.

Finding out two node group disjoint routes on the original graph *G* is equivalent to finding out the node-disjoint shortest cycle on the transformed graphs *G'* as described in the last section, and collapsing the paths of the shortest cycle, on the original graph *G* as demonstrated in Fig. 6(b).

After all the node-disjoint shortest cycles are computed for all the transformed graph *G'*, the paths comprising the cycles can be either in the form *P*: $(s{-}\cdots{-}u{-}d_i{-}v{-}y{\cdots}{-}d)$ or *P'*: $(s - \cdots - u - d_i - d_j - v \cdots - d)$, and the reverse mapping of the paths are done using the following rules:

Case 1: In the reverse mapping of the path *P* from *G'* $\rightarrow$ *G*, we need to transform the edges *u*-d_i-*v* to obtain the corresponding edges in the original graph *G*. It can be obtained by the matching of vertices belonging to the neighborhoods of both the vertices of the shared node risk group.

We scan the neighborhood's of both the vertices *u* & *v* and select vertex *w* such that *w* $\in$ *N(u), N(v)* & *w* $\in N_i$ or N_j. If there are more than one choice of such a vertex, the one that *minimizes* α_{uw} + α_{wv} is selected. The mapped edges in the final path hence becomes *u-w-v*. If no such vertex *w* exists, then select vertices *w* & *w'* such that *w* $\in$ *N(u)*, N_i or N_j and *w'* $\in$ *N(v)*, N_i or N_j. The mapped edges in the final path hence becomes *u-w-w'-v*, where the edge *ww'* connects the two nodes of one of the shared node risk groups N_i or N_j.

Case 2: In the reverse mapping of the path *P'* from *G'* $\rightarrow$ *G*, the same technique described above can be used to transform the edges *u*-d_i-d_i-*v* to obtain the corresponding edges in the original graph *G*. It is obtained by matching of sets of vertices belonging to the neighborhoods of both the vertices of the shared node risk group. We scan the neighborhood's of both the vertices *u* & *v* and select a vertex *w* such that *w* $\in$ *N(u), N(v)* & *w* $\in N_i \cap N_j$. The mapped edges in the final path hence becomes *u-w-v*. If no such vertex *w* exists, then select vertices *w* & *w'* such that *w* $\in$ *N(u)*, *w* $\in N_i \cap N_j$, *w'* $\in$ *N(v)*, *w'* $\notin N_i \cap N_j$. **OR** *w* $\in$ *N(u)*, *w* $\notin N_i \cap N_j$, *w'* $\in$ *N(v)*, $N_i \cap N_j$. The mapped edges in the final path hence becomes *u-w-w'-v*, where the edge *ww'* connects the two nodes of one of the shared node risk groups N_i or N_j.

6. Complexity Analysis of SRNG Routing

This section evaluates the overall complexity of the SRNG disjoint routing algorithm using the graph transformation technique as described in the previous section. The computational complexity can be broken up into three parts, one the complexity involved in the transformation of the graph, second the complexity of finding out node-disjoint cycles in the transformed graph and finally the complexity of mapping the paths obtained on the transformed graph to paths on the original graph.

The total number of shared risk node groups in the network is given by K_N and each group can possibly have a *maximum* of two nodes. Another additional constraint limits the sharing of any node by not more than two groups. Thus as can be seen from the algorithm presented for graph transformation, the complexity involved in determining whether an edge has an end-vertex which belongs to one or more than one shared node risk group and transforming it involves an exhaustive search, which is computationally of the order of O $(2|V| \cdot K_N) \approx$ O $(|V| \cdot K_N)$ in case of non-overlapping shared node risk groups and O $(2|V| \cdot K_N + 4K_N^2) \approx$ O $(|V| \cdot K_N + K_N^2)$ in case of overlapping shared node risk groups. Depending on the outcome of this decision, the edges are transformed following the algorithm explained above. The detailed algorithm is presented in [16]. Thus the overall complexity of the graph transformation is O $(|V| \cdot K_N)$ or O $(|V| \cdot K_N + K_N^2)$ depending on the structure of all the shared node risk groups in the network.

The transformation of the original graph *G* into the final graph *G'* reduces K_N number of nodes in the original graph. Two paths that are shared-risk group disjoint are computed using the node-disjoint shortest-cycle algorithm on the transformed graph as described in [14]. The computational complexity of the shortest-cycle algorithm is given by O $(|V|^2 + |E|^2)$. We have two distinct cases, one in which the SRNG's are such that there is no node which belongs to more than one group and another scenario where a node can possibly belong to two node risk groups. Let us consider the first case, where each node belongs to utmost one SRNG group as shown in Fig. 2(b). During the graph transformation there is a reduction of K_N number of nodes (for a maximal size of each node group to be 2).

Following the graph transformation all the edges between the nodes of each node group gets deleted, hence leading to a deletion of total K_N number of edges. Let us denote the set of vertices which has an edge to both the nodes of any SRNG group as the vertex set V_s. Let the cardinality of this set be denoted by $|V_s|$. Transition from *G* $\rightarrow$ *G'* leads to a further deletion of $|V_s|$ - 1 number of edges. Moreover let dual-edge set be defined as DE_s = {uv: u $\in N_i$, v $\in N_j$, u, v $\notin V_s$, i $\neq$ j}. Hence during the graph transformation, $|DE_s|$ - 1 number of edges gets deleted, where $|DE_s|$ is the cardinality of the set DE_s. The minimal reduction of edges during the graph transformation from *G* $\rightarrow$ *G'* is K_N + $|V_s|$ + $|DE_s|$ - 2. Hence the complexity of finding two SRNG disjoint paths is O $((V - K_N)^2 + (E - K_N - |DE_s| - |V_s|)^2)$.

Now let us consider the case, where we have a node belonging to more than one SRNG's. Following the graph transformation all the edges between the nodes of each node group gets deleted, hence leading to a deletion of total K_N number of edges. Let us consider overlapping-vertex set OV_s as the set of vertices which has an edge to the nodes that belongs to more than one SRNG i.e. $v \in N_i$, N_j, i $\neq$ j. Let the cardinality of this set be denoted by $|OV_s|$. Graph transformation from $G \rightarrow G'$ leads to addition of $|OV_s|$ edges since we have two possible choices of graphs as explained in Section V. Moreover the graph transformation leads to addition of one more edge between the two dummy nodes as shown in Fig. 9. Hence the total number of edges in the transformed graph is given by E - K_N + $|OV_s|$ + 1. The total number of nodes or vertices in the transformed graph is given by $|V|$ - $|n|$ where $|n|$ is the number of vertices that are part of more than one SRNG. A search for SRNG disjoint routes, would necessitate searching node-disjoint shortest cycles on the transformed graph and choosing the minimum weighted cycle amongst all of them. Hence the overall complexity of finding two SRNG disjoint paths is given by O ($\{(V - |n|)^2$ + $(E - K_N + |OV_s|)^2\}$) $\approx$ O ($\{(V - |n|)^2$ + $(E - K_N + |OV_s|)^2\}$).

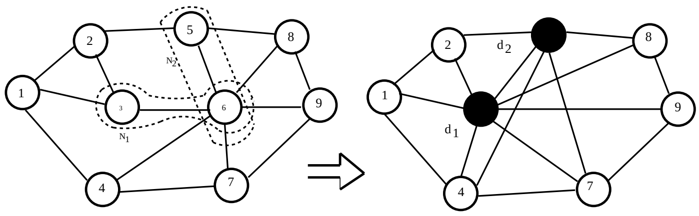


Figure 9. Transformation of edges and vertices for SRNG's.

Once the node-disjoint shortest cycles are computed on the transformed graph, the paths comprising the cycle, are mapped back on the original graph *G*. This computational overhead for the reverse transformation of routes are different in cases where nodes belong to only one shared node risk group and where nodes can belong to more than one shared node risk group. In cases where nodes belong to only one shared node risk group, and paths are of the format *P*: ($s - \cdots - u - d_i - v - y \cdots - d$) or *P'*: ($s - \cdots - u - d_i - d_j - v \cdots - d$), we scan through each edge in the path and at the vertices *u* & *v*, we scan through its entire neighborhood and select the appropriate vertices as described in Section V. Hence the complexity involved in this procedure is $(E - K_N - |DE_s| - |V_s|)^2$, since the maximum hop length of a path can be restricted to *E'* edges and at the vertices *u* and *v* we need to scan through all the neighbors and compare the path lengths in all cases, which can be done in constant time units. Hence the complexity involved in this procedure is O ($(E - K_N - |DE_s| - |V_s|)^2$).

In cases where nodes belong to two shared node risk group, and paths are of the format *P*: ($s - \cdots - u - d_i - v - y \cdots - d$) or *P'*: ($s - \cdots - u - d_i - d_j - v \cdots - d$), we scan through each edge in the path and at each visited vertex, we scan through its entire neighborhood and select the appropriate vertices. Hence the complexity involved in this procedure is O ($(E - K_N + |OV_s|)^2$), since the maximum hop length of a path can be restricted to *E"* edges and at the vertices *u* and *v* we need to scan through all the neighbors and compare the path lengths in all cases, which can be done in constant time units. Hence the complexity involved in this procedure is O ($(E - K_N + |OV_s|)^2$).

Combining the three above complexities, the overall complexity of the diverse SRNG routing is given by the dominant term, which is the complexity involved in finding the node-disjoint shortest cycle on the transformed graph. Thus depending on whether we have nodes belonging to multiple groups or not, the complexity is given by O ($|V| \times \{(V - |n|)^2$ + $(E - K_N + |OV_s|)^2\}$) or O ($(V - K_N)^2$ + $(E - K_N - |DE_s| - |V_s|)^2$) respectively.

7. Conclusion

In this paper we propose graph transformation techniques for solving the diverse routing problem in networks with shared risk resource groups. We proposed a methodology for tolerating dependent or shared risk link failures and coordinated node failures in a network, by creating different graph transformations, routing on the transformed graph and transforming the routes on the modified graph to the original graph. One of the elegant features of the proposed strategy is that it can identify whether 100% guarantee can be provided for any single SRLG or SRNG failure in a network.

The proposed graph transformation heuristic only needs addition of a small number of edges and vertices to the original graph, and computation of link-disjoint or node-disjoint shortest cycles in the transformed graph. It provides a polynomial time solution for shared resource groups with certain restrictions, as has been conjectured by previous researchers.

We also validate the correctness of our approach, and how the graph transformation technique always guarantees to yield shared risk group disjoint routes, if such a route exists in the graph. This approach for diverse routing under multiple failure scenarios is extremely elegant and can be applied to large networks with huge traffic demands. As part of our future work, we plan to extend this graph transformation technique for accommodating groups of link failures that are not incident on a common node.

References

[1] S. Ramamurthy and B. Mukherjee, “Survivable WDM Mesh Networks, Part 1: Protection,” *IEEE INFOCOM 2*, pp.744 -751, 1999.

[2] O. Gerstel and R. Ramaswami, “Optical layer survivability-an implementation perspective,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no.10, pp. 1885–1889, Oct 2000.

[3] R. Doverspike and B. Wilson, “Comparison of capacity efficiency of DCS network restoration routing techniques,” *Journal of Network and System Management*, vol. 2, no.2, pp. 95 -123, 1994.

[4] J. Doucette and W. D. Grover, “Capacity design studies of span-restorable mesh transport networks with shared-risk link group (SRLG) effects,” *SPIE Optical Networking and Communications Conference (Opticomm 2002)*, Boston, MA, July-Aug 2002.

[5] D. Papadimitriou et.al, “Interference of shared risk link groups,” *Internet Draft, OIF Contribution OIF 2001-066*, July 2001.

[6] J. Q. Hu, “Diverse routing in mesh optical networks,” *IEEE Transactions of Networking*, To appear.

[7] G. Li, B. Doverspike and C. Kalmanek, “Fiber span failure protection in mesh optical networks,” *SPIE Optical Networking and Communications Conference (Opticomm 2001)*, 2001.

[8] K. Lee and K. Siu, “An algorithmic framework for protection switching WDM networks,” *Proceeding of NFOEC’01*, pp.402-410, Baltimore, MD, July 2001.

[9] E. Modiano and A. Narula-Tam, “Survivable lightpath routing: a new approach to the design of WDM-based networks,” *IEEE Journal of Selected Areas in Communication*, May 2002.

[10] G. Ellinas *et al.*, “Routing and restoration architectures in mesh optical networks,” *Optical Networks Magazine*, Jan 2003.

[11] J. Suurballe, “Disjoint paths in a network,” *Networks*, vol.4, pp.125-145, 1974.

[12] M. Garey and D. Johnson, “Computers and Intractability: A Guide to the theory of NP-completeness,” , 1979.

[13] D. Xu, Y. Xiong and C. Qiao, “Protection with multi-segments (PROMISE) in Networks with Shared Risk Link Groups (SRLG’s),” *IEEE/ACM Transactions on Networking*, vol.11, no.2, pp.248-258, April-2003.

[14] R. Bhandari, “Survivable Networks: Algorithms for Diverse Routing,” *Kluwer Academic Publishers*, 1999.

[15] J. Suurballe and R. Tarjan, “A quick method for finding shortest pair of disjoint paths,” *Networks*, vol.14, pp.325-336, 1984.

[16] DCNL Technical Report, “DCNL-ON-2004-03,” *http://ecpe.ee.iastate.edu/dcnl/DCNLWEB/Publications/pub-research_tech-rep.htm*.