Survivability under Shared Risk Link Group for Optical WDM Mesh Networks
Suhas Nandakumar, Aiyappa Chottangada Ganesh

Abstract:
In this paper, we try to survey survivability schemes in optical networks, specifically, concentrating on protection under SRLG constraints. Approaches for protection can be divided based on granularity a particular scheme works at. Protection schemes can be applied to individual link or spans, or can be applied at end to end path level in optical transport network. In this paper, we speak about protection mechanism for OTN under single SRLG failure model. We speak specifically about Segment Based Protection and p-Cycle based protection schemes for survivability under SRLG constraints with their respective algorithms, problem formulation.

Index Terms— p-Cycles, Optical WDM Mesh Networks, NP Completeness, protection, wavelength division multiplexing

1. Introduction
This paper begins by introducing need for survivability mechanisms, followed by, a description of SRLG, problem description of survivability mechanisms under SRLG constraints and complexity analysis of the same in Section 2. Finally, we speak specifically about Segment Based Protection and p-Cycle based protection schemes for survivability under SRLG constraints with their respective algorithms and problem formulation in Section 3.
The rapid increase of the Internet demands large volumes of bandwidth. Wavelength division multiplexing (WDM) technology has the potential to meet this need by allowing simultaneous transmission of traffic on multiple wavelengths in a fiber. A wavelength-routed network (WRN) based on WDM technology is deemed as a promising candidate for the core network of the next generation Internet.

Fault recovery capability is critical for optical networks, as a single failure may affect a large volume of traffic. With advent in terms of Traffic grooming, a failure may indeed cause huge loss of data due to multiplexing of several traffic streams.

There are generally two types of fault recovery mechanisms, namely protection and restoration. Protection aims at extremely fast recovery. The backup connection is established before the failure. Restoration, on the other hand, dynamically establishes a connection to recover from a failure after the failure occurs. Restoration, although relatively slow, uses less resource than protection. Note that irrespective of whether protection or restoration is used, spare capacity needs to be preplanned in order to provide survivability in optical networks.

1.1 Causes of Failure [13]

It is reasonable to ask why fiber optic cables get cut at all, given the widespread appreciation of how important it is to physically protect such cables. Isn’t it enough to just bury the cables suitably deep or put them in conduits and stress that everyone should be careful when digging? In practice what seems so simple is actually not. Despite best-efforts at physical protection, it seems to be one of those large-scale statistical certainties that a fairly high rate of cable cuts is inevitable.

And so it is with today's widespread fiber networks: it doesn't matter how advanced the optical technology is, it is in a cable. When you deploy 100,000 miles of any kind of cable, even with the best physical protection measures, it will be damaged and with surprising frequency. One estimate is that any given mile of cable will operate about 228 years before it is damaged (4.39 cuts/year/1000 sheath-miles) [ToNe94]. At first that sounds reassuring, but on 100,000 installed route miles it implies more than one cut per day on average. As published in 2002 by FCC the metro networks annually experience 13 cuts for every 1000 miles of fiber, and long haul networks experience 3 cuts for 1000
miles of fiber [VePo02]. These frequencies of cable cut events are hundreds to thousands of times higher than corresponding reports of transport layer node failures, which helps explain why network survivability design is primarily focused on recovery from span or link failures arising from cable cuts.

1.2 Crawford's Study

After several serious cable-related network outages in the 1990s, a comprehensive survey on the frequency and causes of fiber optic cable failures was commissioned by regulatory bodies in the United States [Craw93]. Figure 3-1 presents data from that report on the causes of fiber failure. As the euphemism of a "backhoe fade" suggests, almost 60% of all cuts were caused by cable dig-ups. Vehicle damage was most often suffered by aerial cables from collision with poles, but also from tall vehicles snagging the cables directly or colliding with highway overpasses where cable ducts are present. Human error is typified by a craftsperson cutting the wrong cables during maintenance or during copper cable salvage activities ("copper mining") in a manhole. Power line damage refers to metallic contact of the strain-bearing "messenger cable" in aerial installations with power lines. The resulting $I^2R$ (heat dissipation) melts the fiber cable. Rodents (mice, rats, gophers, beavers) seem to be fond of the taste and texture of the cable jackets and gnaw on them in both aerial and underground installations. The resulting cable failures are usually partial (not all fibers are severed). It seems reasonable that by partial gnawing at cable sheaths, rodents must also compromise a number of cables which then ultimately fail at a later time. Sabotage failures were typically the result of deliberate actions by disgruntled employees, or vandalism when facility huts or enclosures are broken into. Today, terrorist attacks on fiber optic cables must also be considered. [1]
1.3 Managing Link Failures

Although the failure of a single component such as a link or a node is the most common failure scenario, it is possible to have multiple links fail simultaneously. In particular, double-link failures can happen in the following scenarios.

Scenario 1:

The first link fails. The recovery from the failure of the first link is completed within a few milliseconds to a few seconds. However, it may take a few hours to a few days for the failed physical link to be repaired. It is certainly conceivable that a second link might fail during this period, thus causing two links to be down at the same time.

Scenario 2:

Two links may be physically routed together for some distance in real situations. A single backhoe accident may lead to the failure of both links. This is a typical example of SRLG failure.
In order to protect connections from link failures in the network, two paths are often assigned: a primary path on which a connection is established and a backup path on which a connection will be re-established in the case where the primary path fails.

**Link restoration** schemes provide a detour around a failed link that does not necessarily affect the entire source–destination path. **Path restoration** schemes, in general, attempt to provide a backup path from the source to the destination that is independent of the working path. Path restoration schemes are classified into two categories based on knowledge of the link failure. A backup path that can be used for any link failure on the working path and is link-disjoint with the working path is referred to as *failure-independent path restoration*. Alternatively, a connection may be assigned more than one backup path depending on the failure scenario. Such an approach requires complete knowledge of the failure in the network; hence it is referred to as *failure-dependent path restoration*.

Path-based restoration has been established to be a more capacity-efficient approach for mesh-based networks compared to link-based restoration approaches.

The minimum sufficient condition for network to be single failure survivable in cases of link or path failures is that the network should be at least 2-**edge connected**. For multiple link failures, the graph needs to have higher connectivity.

Both path and link based protection schemes can be either dedicated or shared. In the former case, for each primary path/link a dedicated backup path/link is pre-computed. In the latter, the backup counterparts can be shared if their primary path/link is mutually disjoint.
Comparison between Dedicated and Shared Protection Schemes

More recently, approaches like segment based protection and p-cycles based protection have evolved for single and multiple link failures.

P-Cycle is a promising approach for survivable WDM networks design because of its ability to achieve ring-link recovery speed while maintaining the capacity efficiency of a mesh-restorable network.

Fig: p-Cycle with on-cycle and straddling link failure.
A: Network and p-Cycle 1, 2, 3, 4, 5, 6
B: On cycle link 1-2 failed and traffic re-routed through 2 → 4 → 5 → 1
C: Straddling link 1 → 5 failure and traffic re-routed through 2 alternative paths, a) 1 → 6 → 5 b) 1 → 2 → 4 → 5
Like a self-healing ring, a p-cycle provides one restoration path for every on-cycle span; unlike a self-healing ring, a p-cycle also provides two restoration paths for every straddling span—a span whose two end nodes are on the cycle but itself is not on the cycle.

In the latter case, Segment based protection refers to protection using segments instead of entire protection paths. A segment is nothing but a set of contiguous links on the path between the source and the destination. We observe that when a link in the primary path connecting the source and destination were to fail, then we might opt for a secondary path that is both edge and vertex diverse from the primary path. However, this scheme would also need to take care of SRLG constraints and hence the secondary path would also require being SRLG diverse from the primary path. Traps [11] are hence needed to be avoided.

Consider the network topology in Fig above. We observe that if the primary path from source node 0 to destination node 5, passing through 0-1-2-5, were to fail, then since no other SRLG diverse path exists, we could hence use two shared segments to protect the primary path. One segment runs from 0-3-2 and the second segment runs from 2-3-4-5.
2. Shared Risk Link Group

Optical networks have at least two layers: the physical layer and the optical layer. The physical layer consists of fiber spans and nodes that represent locations where fiber spans terminate. The optical layer consists of optical links (or lightpaths) and a subset of nodes contained in the physical layer. An optical link is a path connecting a pair of nodes via a set of fiber spans in the physical layer. Therefore, an optical link may traverse several fiber spans and nodes. In addition, several optical links may traverse a single fiber span or node. Therefore, a single failure in the physical layer can cause multiple failures at the optical layer.

Routing diversity is needed to achieve the reliability and survivability expected of modern transport networks. Risks of failure are modeled, and routes designed to share no common risk in the modeled set. In today’s networks, manual provisioning techniques are used to ensure routing diversity.

Implicit in the provisioning operation is the notion of a Shared Risk Link Group (SRLG), a group of links that share a component whose failure causes the failure of all links of the group. An important example of such a component is a common conduit in the ground, through which many optical fibers may be routed. Optical lightpaths are provisioned through fibers that belong to an SRLG associated with the conduit. A typical requirement in transport network management might be that for each operational lightpath there is an associated “backup” lightpath, and there is no SRLG on which both of these lightpaths are routed.
Above Figure illustrates a simple example of the SRLG concept, for two links (A-B and C-D) and corresponding physical routing through conduits (A-X, C-X, X-Y, Y-B and Y-D). The A-B link is routed on the conduit, path A-XY-B, and the C-D link is routed on the conduit path C-X-Y-D. Both A-B and C-D links may each consist of multiple fibers. Each conduit determines a SRLG; in particular, SRLG2 (associated with conduit X-Y) consists of both links A-B and C-D, and represents the shared risk of interrupting both links if the conduit fails, because of backhoes, train derailments and so forth.

2.1 Diverse Routing Problem in Optical Transport Networks [3]

In general, Diverse Routing Problem can be treated as way to route the traffic demand in the network such that failures at the physical level doesn’t affect the flow.

A bandwidth demand requires two paths in a network, one working path and one protection path, so that the service to the demand can be honored in case of a single network failure, such as a fiber cut. A basic requirement for the pair of working and protection paths is that they must be diversely routed. The structure of Optical networks into two layers: the physical layer and the optical layer may result in an optical link to traverse several fiber spans and nodes.
In addition, several optical links may traverse a single fiber span or node. Therefore, a single failure in the physical layer can cause multiple failures at the optical layer. In general, the requirement for working path and protection path is that they have to be diversely routed so that at least one path can survive a single failure in the network, where a single failure may represent a fiber cut or an individual card failure at a node. Such an structure of optical network makes the problem stated above, difficult and more challenging than the classical edge/node disjoint path problems.[2]

In section on **“Nature and Complexity of the problem”** we try to put forward the intuition that, a problem, such as finding the diverse route, is indeed NP Complete. Hence diverse routing under SRLG constraints, which is a more general case, is also NP Complete.

### 2.2 Problem Formulation for SRLG Diverse Routing

In [3], as per the author, for an optical network, graph $G_f = (V_f, E_f)$ is used to represent its physical layer, where $E_f$ is the set of edges representing fiber spans and $V_f$ is the set of nodes representing locations (where the fiber spans terminate). Graph $G_o = (V_o, E_o)$ is used to represent its optical layer, where $E_o$ is the set of edges representing optical links (light paths) and $V_o$ which is subset of $V_f$ represents end points of the optical links. In general, each edge in $E_o$ may correspond to a path in graph $G_f$. Let $R$ be the set of risks (failures). Each risk may represent a fiber cut, a card failure at a node, a piece of software failure, an operational error, or any combination of these factors. Let $E_r$ (subset of $E_o$) denote the subset of optical links that can be affected by $r \in R$, and $E_r$ is referred as an SRLG. For a path in $G_o$, we say it contains $r \in R$ if any edge on the path belongs to $E_r$, and $r_p = \{ r \in R : \text{path } p \text{ contains } r \}$ holds.
Then the SRLG diverse routing problem can be defined as follows.

**Definition 1 (SRLG Diverse Routing Problem):**

Find two paths $p_1$ and $p_2$ between a pair of nodes such that $r_{p1} \cap r_{p2} = \text{Null}$, i.e., $p_1$ and $p_2$ don’t contain a common element of $R$. We also say that $p_1$ and $p_2$ are two SRLG diverse paths (with respect to $R$).[3]

In case that there do not exist two SRLG diverse paths, and then a related problem is the following least coupled SRLG paths problem.

**Definition 2 (Least Coupled SRLG Paths Problem):**

Find two paths $p_1$ and $p_2$ between a pair of nodes such that $| r_{p1} \cap r_{p2} |$ is minimized, i.e., the number of common elements in $R$ shared by both paths is minimized.[3]

The least coupled SRLG paths problem was also considered in [4], where it was set up as an ILP problem based on the cut set formulation. Clearly, the SRLG diverse routing problem can be viewed as a special case of the least coupled SRLG paths problem in which $| r_{p1} \cap r_{p2} | = 0$. If there is a cost associated with each edge in $G_o$, then the SRLG diverse routing problem can be extended to the minimum-cost SRLG diverse routing problem.

**Definition 3 (Minimum-Cost SRLG Diverse Routing Problem):**

Find two SRLG diverse paths between a pair of nodes such that their total edge cost is minimized (the total edge cost is defined as the summation of costs of all edges on the two paths).[3]

A special case of the SRLG diverse routing problem is the fiber-span-disjoint paths problem in which $R = E_f$ (only failures considered in the physical layer are fiber cuts) and every edge in $E_o$ represents a path in $G_f$. Other special cases of the SRLG diverse routing problem include the edge-disjoint paths problem ( $R = E_o$ and $E_r = \{r\}$ ) and the node-disjoint paths problem ( $R = E_o$ and $E_r$ and contains all edges in which are connected to node $r$ ).
Other generalizations of the SRLG diverse routing problem include: 1) find $k > 2$ SRLG diverse paths between a pair of nodes and 2) find the maximum number of SRLG diverse paths between a pair of nodes.

### 2.3 Complexity Analysis of the problem

In [3], the author proves that the SRLG diverse routing problem is NP-complete, which immediately implies that both the least coupled SRLG paths problem and the minimum-cost SRLG diverse routing problem are NP-complete.

The proof begins by introducing a special type of optical layer graph $G_{l,o} = (V_{l,o}, E_{l,o})$. The graph $G_{l,o}$ is divided into subgraphs, which are denoted as $G_1, G_2, G_3, ..., G_S$. In subgraph $G_i (i=1,2,3, ..., S)$, there are two end nodes $i$ and $i-1$ which are connected by a set of parallel paths. If multiple edges are allowed between a pair of nodes, then each parallel path between nodes $i-1$ and $i$ can simply be a single edge; otherwise, each parallel path is made of two edges is assumed. Further it is assumed that the (two) edges on each of these parallel paths belong to the same SRLG and the edges on different parallel paths belong to different SRLGs.

Let $R_i = \{ r \in R : r \text{ is contained by an edge in } G_i \}$ i.e., $R_i$ is the subset of $R$ for subgraph $G_i$.

Now with the formulations as described above, the proof for NP Completeness as described in [3] can be described as given below.

**Theorem:** The SRLG diverse routing problem is NP-complete for nodes 0 and $S$ in $G_{l,o}$, hence, it is NP-complete in general.

**Proof:**

Since finding a path between nodes 0 and $S$ is equivalent to finding a subset of which contains at least one element in each $R_i (i=1,2,3, ... S)$, this problem is equivalent to finding two disjoint subsets of such that each of them contains at least one element from each subset $R_i (i=1,2,3, ... S)$.
The proof in the paper shows that the well-known set-splitting problem can be reduced to the SRLG diverse routing problem and hence prove that SRLG diverse routing problem is indeed NP Complete.

Once having proved the general SRLG Diverse Problem, author in [3] carries on proving Least Coupled SRLG Paths Problem and Minimum-Cost SRLG Diverse Routing Problem to be NP Complete as well.

Least Coupled SRLG Paths problem is proved to be NP Complete as summarized below. The fact that the problem of finding two node covers which share minimum number of common nodes can polynomially transformed to least coupled SRLG paths for nodes 0 and S in G_{10} with |R_i| = 2 can be used to prove that the latter is NP-Complete using following 2 Lemmas. Lemma 2 is proved to be equivalent to the maximum bipartite subgraph problem by showing that for every 2 node covers a bipartite subgraph can be generated and set of common nodes shared by the two node covers is the complementary node set of the corresponding bipartite subgraph.

Lemma 1: For a given graph G = (V,E), its maximum bipartite subgraph problem, i.e., finding a bipartite subgraph with maximum number of nodes, is NP-complete.

Lemma 2: For a given graph G = (V,E), the problem of finding two node covers that share the minimum number of common nodes is NP-complete.
3. Approaches for protection under SRLG constraints

In the single link failure model, the protection that can be provided for a link depends on their relationship. Upon an SRLG failure, all links in this SRLG are gone. To restore such a failure, every failed link must be taken care of by protection/restoration mechanism.

Hence existing protection/restoration schemes may not be applied without modifications to work for SRLG related failures. Following sections surveys 2 different schemes for protection under SRLG constraints

3.1 p-Cycle Based Protection upon a SRLG Failure

As explained earlier, in the single link failure model, the protection that can be provided by a p-cycle for a link depends on their relationship. Specifically, if the link is on-cycle, then the p-cycle can offer one restoration path in case the link fails; if the link straddles the p-cycle, then two restoration paths are provided by the p-cycle when the link is broken; otherwise the p-cycle cannot protect the link.

Upon an SRLG failure, all links in this SRLG are gone. To restore such a failure, every failed link must be taken care of by some p-cycles. Meanwhile, since multiple links may fail in case of an SRLG failure, if two or more failed links happen to be on the same p-cycle, then the p-cycle is broken, which makes the situation more complicated than in the single link failure model as explained in the figures below.
As shown in the figure above. When the links 1→5, 2→4 fail p-cycle is still connected and provides 2 restoration paths the straddling spans. Since the span 3→6 is not part by the p-Cycle 1 → 2 → 3 → 4 → 5 → 6, it is not protected. All these links belong to same SRLG.

As shown in the figure to the left. When the links 1→5, 2→4 fails p-cycle is still connected and provides 2 restoration paths these straddling spans. Since the span 3→6 is not part by the p-Cycle 1 → 2 → 3 → 4 → 5 → 6, it is not protected. All these links belong to same SRLG.
The above case shows, when links 1→2, 3→4, 7→4 and 6→5, the p-Cycle is disconnected.
Hence, a p-Cycle based restoration approach for SRLG failures is more challenging.

3.1.1 Algorithms for p-Cycle based protection

In order to provide the p-Cycle protection under SRLG constraints, the first and foremost thing is to pre-compute cycles that form part of final p-Cycle design. However, since the number of cycles in a network grows exponentially with the network size, various methods have been proposed to reduce the size of the candidate p-Cycle set. One method is to limit the maximum length of the candidate p-cycles [5][6]. Another method is to select a certain amount of candidates from all cycles according to the a priori efficiency metric pre-computed for each cycle [7]. While both methods can reduce the number of candidate p-cycles, they still require the enumeration of all cycles in the network. To address this problem, an algorithm called SLA was proposed in [8] wherein one p-cycle for each spans in the network so that the span is a straddling span of the p-cycle.
To circumvent the inherent hardness of ILP, a heuristic algorithm called CIDA was proposed in [5]. In CIDA, authors choose a p-cycle from candidate p-cycles according to an efficiency score and placed in the network to reduce the unprotected working capacity iteratively until all working capacities are protected.

In the paper [10], the authors consider single SRLG failure model. As explained, SRLG failures are more serious to p-Cycle design, since failure of an SRLG can disconnect an entire p-Cycle. The authors define CYCLE_LINK_SRLG algorithm, that takes a cycle i, a link j and a SRLG k as inputs and return number of restoration paths that can be provided for link j by cycle i in case of failure of SRLG k. The pseudo-code is as given below.

```plaintext
int CYCLE_LINK_SRLG(cycle i, link j, SRLG k)
1. if j doesnot-belong k then
2. return 0; // The link does not belong to the SRLG and therefore does not fail.
3. if link j ’s end nodes are not both on cycle i then
4. return 0; // The link cannot be protected by the cycle.
5. Remove links in SRLG k from cycle i .
6. if cycle i remains a cycle then
7. return 2;
8. else // Cycle i is broken into one or more segments.
9. if link j ’s end nodes are on the same segment then
10. return 1;
11. else
12. return 0;
```
cycle i, link j, SRLG k

j belongs to k? 

NO

Return 0

YES

Is j's end nodes are both on cycle i?

NO

Is link j's end nodes are on the same segment?

NO

Return 0

YES

Remove links in SRLG k from cycle i

YES

Cycle i remains a cycle?

NO

Return 2

YES

Return 1
3.1.2 Algorithm for the generation of candidate cycles

As in the single link failure model, enumerating all cycles of the network is necessary to achieve the optimal p-cycle design. This requirement blows up the complexity of the p-Cycle design since the number of cycles in a network grows exponentially with the network size. To ease this difficulty, the authors give an algorithm for generating a small subset of all cycles as the candidate p-cycle set such that a p-cycle design can be found to fully survive any single SRLG failure in the network given enough spare capacities. The algorithm works as follows. For each SRLG, first all the links are removed from the network graph, then for each pair of end nodes of a removed link, its shortest path as well as two node-disjoint shortest paths (if exist) in the remaining graph is found. The shortest path is combined with the removed link to form a cycle that contains the removed link as on-cycle link, and the two node-disjoint shortest paths (if exist) are combined to form a cycle on which the removed link straddles. The distinct cycles generated are collected into the candidate p-cycle set. FIND_BASIC_CYCLES is a function implementing is the algorithm.

Given a 2-edge-connected network and an SRLG set such that any single SRLG failure does not disconnect the network, the algorithm can always find a candidate p-cycle set that can provide 100% restorability in case of any single SRLG failure given enough spare capacities. The reason is that in case of to find a shortest path between the two end nodes of the link because the network is still connected. This guarantees that when an SRLG fails, each link in the SRLG has at least one cycle that can provide a restoration path for it.
FIND_BASIC_CYCLES (network $G$, SRLG set $R$)

1. $P = \emptyset$;

2. for each SRLG $k \in R$ do

3. let $G = (V, L - k)$;

4. for each link $l = (u, v) \in k$ do

5. Compute the shortest path between $u$ and $v$ in $G$.

6. let $c$ = the cycle formed by the shortest path and $l$;

7. let $P = P \cup \{c\}$;

8. Compute two node-disjoint shortest paths
between $u$ and $v$ in $G$.

9. if such a pair of node-disjoint paths exist then

10. let $c$ be the cycle formed by the two paths;

11. let $P = P \cup \{c\}$;

12. return $P$;
Network G, SRLG set R

$P = \emptyset$

for each SRLG $k \in R$

$G = (V, L - k)$;

for each link $l = (u, v) \in k$

Compute the shortest path between $u$ and $v$ in $G$;
Let $c$ = the cycle formed by the shortest path and $l$;
Let $P = P \cup \{c\}$;

Compute two node-disjoint shortest paths

Does such a pair of node-disjoint paths exist?

Let $c$ be the cycle formed by the two paths;
Let $P = P \cup \{c\}$;

Return $P$
3.1.3 ILP Formulation for Optimal p-Cycle Design

Authors in [3] consider the following p-Cycle design problem: given a network represented by a graph $G = (V, L)$, a set of SRLGs in $G$, a set of distinct candidate $p$-cycles in $G$, and the working capacity on each link in $G$, compute a set of $p$-cycles that minimizes the total spare capacity cost on all links subject to the constraint of 100% restorability in case of a single SRLG failure.

To guarantee the existence of a solution to this problem, the following conditions are assumed:

- The network is 2-edge-connected so that each link can be protected by at least one cycle.
- The failure of any single SRLG does not disconnect the network.
- In case of any single SRLG failure, for each link in this SRLG, at least one cycle exist in the candidate $p$-cycle set such that the cycle can provide at least one restoration path for it.
- There is enough capacity on each link in the network.

B.

Sets: (input)

$L$: The set of all links.

$P$: The set of candidate $p$-cycles.

$R$: The set of SRLGs

Parameters: (input or pre-computed)

$w_j$: The working capacity on link $j$.

$c_j$: The cost of one unit of spare capacity on link $j$.

$pi_j$: 1 if link $j$ is on cycle $i$, 0 otherwise.

$bjk$: 1 if link $j$ is in SRLG $k$, 0 otherwise.
\( xi \cdot jk \): The number of restoration paths for link \( j \) that can be provided by cycle \( i \) in case SRLG \( k \) fails. This value can take 0, 1, or 2, which is pre-computed by the function CYCLE_LINK_SRLG\((i, j, k)\).

Variables: (to be determined)

- \( s \cdot j \): The spare capacity on link \( j \).
- \( n \cdot i \): The number of copies of cycle \( i \) needed in the \( p \)-cycle design.
- \( n \cdot ik \): The number of copies of cycle \( i \) needed in case SRLG \( k \) fails.
- \( n \cdot ijk \): The number of copies of cycle \( i \) needed for link \( j \) in case SRLG \( k \) fails.

Objective Function

**Minimize**

\[ \sum_{j \in L} c \cdot j \cdot s \cdot j \]

Subject to:

1. \( s \cdot j = \sum_{i \in P} p \cdot i \cdot j \cdot n \cdot i \quad \forall \ j \in L \)
2. \( b \cdot jk \cdot w = \sum_{i \in P} x \cdot i \cdot jk \cdot n \cdot ijk \quad \forall \ j \in L, \forall \ k \in R \)
3. \( n \cdot ik = \sum_{j \in L} b \cdot jk \cdot n \cdot ijk \quad \forall \ i \in P, \forall \ k \in R \)
4. \( n \cdot i \geq n \cdot ik \quad \forall \ i \in P, \forall \ k \in R. \)

Constraints (1) reflects the relationship between the spare capacities and the result \( p \)-cycle design. Constraints in (2) guarantee that if link \( j \) is in SRLG \( k \) (i.e., \( b \cdot jk = 1 \)), its working capacity should be protected in case SRLG \( k \) fails. And if link \( j \) is not affected by SRLG \( k \)'s failure (i.e., \( b \cdot jk = 0 \)). Constraints in (3) ensure that when SRLG \( k \) fails, all links in \( k \) should be restored. So the number of copies of cycle \( i \) needed for SRLG \( k \)'s failure is the sum of the number of copies of cycle \( i \) needed for all links in SRLG \( k \). Constraint (4) is equivalent to \( n \cdot i = \max_{k \in R} n \cdot ik \). This equation holds because under the single SRLG failure
assumption, the number of copies of cycle i needed is dominated by the maximum requirement over all single SRLG failures.

3.2 Segment Based Protection Scheme under SRLG constraints

3.2.1 Traps and their avoidance

Traps are nothing but situations where we could possibly come to a complete stand-still in so far as finding a protection path is concerned. Finding protection paths requires us to use diverse routing schemes to find edge, vertex and SRLG disjoint routing algorithms. When an algorithm is not able to find a pair of edge, vertex or SRLG disjoint paths for a given source and destination node pair, the algorithm is said to be in a trap [11].

A trap could either be a real trap or an avoidable trap. A real trap is a consequence of the connectivity, which includes the topology and link bandwidth or capacity trap [11]. This implies that there does not exist any edge, vertex or SRLG disjoint paths between the source and destination under consideration. We can hence not use any algorithms to avoid real traps.

Avoidable traps are induced due to limitations of the algorithm being used. These can however be avoided by using more extensive algorithms such as ILP[11].

We can still define two more conditions that could also be a classification of real traps. Dangling traps are those conditions when a single failure of the physical network is sufficient to completely isolate a node from network. Non-dangling traps are those real traps that do not result in dangling traps.
3.2.2 Algorithms for Segment Based Protection

We first consider the Two-Segment (TS) Algorithm [12]. This algorithm is proposed to support the claim that just two backup segments are sufficient to protect any backup path. The details of the algorithm are as follows [12]:

**Input:** A graph $G(V, E)$, SRLG information of each link, and an active path $P = (s, v_1, \ldots, v_n, d)$ from source node $s$ to destination node $d$.

**Output:** A SRLG-disjoint backup path for the given active path $P$ or two backup segments for protecting the active path.

**Step 1:** Remove the links on path $P$ as well as any other links that are not SRLG-diverse with path $P$, and use a shortest path algorithm such as Dijkstra’s algorithm to derive a SRLG-disjoint path for path $P$. If such a path exists, return the derived path; and otherwise, go to step 2.

**Step 2:** Construct an auxiliary graph $G' = (V', E')$. Initialize $V'$ to be the nodes in the active path $P$ and $E'$ be empty.

**Step 3:** For each directed link $(i, j)$ in $P$, add its reverse link $(j, i)$ into the graph $G'$ and assign a cost 0 to these links.
**Step 4:** For each immediate node $V_i$ in path $P$, let $AS(s, V_i)$ be the active segment in path $P$ from source node $s$ to node $V_i$.

a) Remove the links on the active segment $AS(s, V_i)$ as well as any other links that are not SRLG-diverse with the active segment $AS(s, V_i)$ from $G$, and

b) Use a shortest path algorithm such as Dijkstra’s algorithm to derive a SRLG-disjoint backup segment for the active segment $AS(s, v_i)$.

c) If such a backup segment exists, add a directed link from source node $s$ to node $v_i$ to the auxiliary graph $G'$ and assign the cost of the backup segment as the link cost. If all immediate nodes are considered, go to step 5.

![Diagram](image_url)

**Step 5:** For each immediate node $V_i$ in path $P$, let $AS(V_i, d)$ be the active segment in path $P$ from node $V_i$ to destination node $d$.

a) Remove the links on the active segment $AS(v_i, d)$ as well as any other links that are not SRLG-diverse with the active segment $AS(v_i, d)$ from $G$, and

b) Use a shortest path algorithm such as Dijkstra’s algorithm to derive a SRLG-disjoint backup segment for the active segment $AS(v_i, d)$.

c) If such a backup segment exists, add a directed link from node $v_i$ to destination node $d$ to the auxiliary graph $G'$ and assign the cost of the backup segment as the link cost. If all immediate nodes are considered, go to step 6.
Step 6:

a) Use a shortest path algorithm to derive a shortest path from source node s to destination node d in the auxiliary graph G’.

b) If a shortest path exists, return the backup segments corresponding to the links in the shortest path. And

c) Otherwise, return FAIL and reject the connection request.
4. Conclusion

We studied the problem of protecting optical transport networks under single SRLG failure model. The techniques discussed described the computational nature the problem and specific approaches for providing protection through mechanisms based on segments and p-cycles. All these approaches and various approaches currently under research arena try to deal with inherent complexity associated with the problem through ILP formulations to optimize the exponential solution set and then applying heuristics to provide polynomial time algorithm that produces near optimal solutions.

References

[1]. Survivability and traffic grooming in WDM optical networks – Arun K Somani
[3]: Diverse Routing in Optical Mesh Networks – Jian Qiang Hu

[10] p-cycle based design under SRLG constraints – Chang Lui, Lu Ruan

[11] Trap Avoidance and Protection Schemes in Networks With Shared Risk Link Groups Dahai Xu, Student Member, IEEE, Yizhi Xiong, Associate Member, IEEE, Chuming Qiao, Member, IEEE, and Guangzhi Li, Member, IEEE.

[12] Insights for Segment Protection in Survivable WDM Mesh Networks with SRLG Constraints - Hongbin Luo, Lemin Li, and Hongfang Yu