p^2-Cycles: p-Cycles with Parasitic Protection Links

Long Long Ahmed E. Kamal
Dept. of Electrical and Computer Eng., Iowa State University
Ames, IA 50011, U.S.A.
E-mail: {longlong, kamal}@iastate.edu

Abstract

The p-cycle and its Failure Independent Path Protection (FIPP) extension are known to be efficient and agile protection strategies. The p-cycle is pre-configured such that if there is a failure, only the switches at two end nodes need to be reconfigured. In this paper, we extend the p-cycle by allowing cycles to have attached links, called Parasitic Protection Links (PPL), in order to protect paths whose source and destination nodes are not only located on the cycle but also connected by a PPL to the cycle. A p-cycle with PPL is named p^2-cycle.

We address the unicast service protection problem against single-link failures by using p^2-cycle in mesh networks for both static and dynamic traffic scenarios. In the static case, the problem is formulated as an Integer Linear Program (ILP). We further propose two p^2-cycle based heuristic algorithms, Strict Routing Protection (SRP) and Flexible Routing Protection (FRP), to address the dynamic traffic case. The numerical results show that the p^2-cycle scheme provides better capacity efficiency than the FIPP p-cycle scheme in all the traffic scenarios considered and achieves only less than 1% extra total cost over the optimum in COST239, provided by Shared Backup Path Protection (SBPP) approach when the traffic load is high. We also study the failure recovery performance in terms of average number of switch reconfigurations (NOR), and show that the performance of the p^2-cycle becomes much better than that of SBPP and gets close to FIPP as the traffic demand increases. In the dynamic case, both SRP and FRP outperform FIPP p-cycle schemes in terms of blocking probability in most scenarios considered. In general, the p^2-cycle protection scheme outperforms the p-cycle based in terms of capacity efficiencies which being slightly slower in terms of traffic recovery speed.

I Introduction

Network survivability, defined as the ability of networks to continue to function properly in the presence of the failures of network components [1], is an important requirement for WDM optical networks due to their ultra-high capacity. A single failure can disrupt millions of applications and users. Ring-based networks and resilience schemes are prevalent due to the simple manageability and fast recovery mechanism, in which the traffic recovery process can be completed within 50-60 ms, but require 100% capacity redundancy [2]. As mesh-based networks emerged, more capacity efficient protection schemes were proposed which allow backup capacity sharing. These schemes fall into three categories: link-based, segment-based and path-based [3, 4]. Link-based protection schemes produce the fast traffic recovery speed but suffer from the worst resource efficiency [5]. As capacity cost is one of key factors in network design [6, 7], path-based protection schemes are usually proposed to achieve the best capacity efficiency. Among them, a path protection scheme, namely, Shared Backup Path Protection (SBPP), was shown to be the most capacity efficient protection scheme [3]. However, it suffers from long traffic recovery time upon a network failure. Segment-based protection schemes lie between the link-based and path-based schemes, and offer a better combination of bandwidth efficiency and recovery time [8, 9].

The pre-configured protection cycle approach, referred to as p-cycle, combines the merits of both ring-based and mesh-based protection schemes and achieves the recovery speed of ring-based with the capacity efficiency of mesh protection [10, 11]. A thorough study of p-cycle-based survivability techniques was conducted by Grover in [12]. Since the concept of p-cycle was first introduced in [10], a large amount of work in the literature studied the p-cycle design problem with unicast traffic against a single-link failure. The authors in [10, 13] introduced a tractable solution by solving the problem in two steps: by first routing the connections, and then selecting the best p-cycles candidates from the enumeration of all the cycles to protect the established connections. In [14, 15], however, the problem were solved jointly by minimizing the total capacity cost used by both primary paths and protection p-cycles.

Besides link protection, p-cycles has been extended to protect segments and paths in [16, 17]. Reference [17] proposed a Failure Independent Path-Protecting (FIPP) p-cycle which is a more capacity efficient protection strategy than link protecting p-cycle. Recently, the author of [19] introduced a new 1+N protection scheme against single-link failures by combining network coding and p-cycles. Besides p-cycles, other pre-configured structures are also used for fast recovery, such as non-simple p-cycle [20, 21], p-trails [22], p-trees [24, 25] and Cooperative Fast Protection(CFP) [23]. A cycle is a non-simple cycle if one or more node on the cycle is traversed by the cycle more than twice. The study in [21] reveals that the major capacity gain of non-simple p-cycles over simple p-cycles lies in small networks with lightly-loaded traffic. In [25], the authors extended traditional p-tree by adding links to form a more flexible protection pattern, such as cycles, trails or trees. It is a link-based protection scheme and provides higher protection capacity than link-protecting simple and non-simple p-cycles. However, the short recovery time cannot always be guaranteed due to the flexibility of the protection structure. The authors in [23] enhanced the protection capacity utilization by solving the backhaul problem, in which the same link is traversed twice in opposite directions by the protection path before reaching the destination after a link failure. However, it suffers from longer switch reconfiguration time due to the fact that all failure-aware nodes need to carry out protection switching after failure detection.

Regardless of the protection schemes, the trade-off between the capacity efficiency and failure recovery speed always exists [26]. Since the p-cycle has a good combination of capacity and time efficiency, we attempt to further increase the capacity efficiency of FIPP p-cycles without sacrificing too much of its fast recovery property.

There are two basic motivations behind the work introduced in this paper, and both of these motivations have their root in the fact that protection using FIPP p-cycles requires the provisioning of complete cycles in order to provide protection. We are therefore motivated to: (1) find protection schemes which are more efficient than FIPP p-cycles in terms of protection resource requirements. We would like to do this while still guaranteeing recovery from failures within 50-60ms, i.e., by reducing the number of reconfigurations; and (2) find protection strategies that can be used under dynamic traffic which do not require the creation of new FIPP p-cycles, i.e., by extending existing FIPP p-cycles. Our solution to these two problems is to extend the FIPP p-cycle paradigm to a new one in which each p-cycle may be augmented with a number of protection links that are attached to the cycle, called "Parasitic Protection Links (PPL)". This new structure is more efficient than FIPP p-cycles since it may add one or two links, rather than constructing a new p-cycle. Moreover, it is faster to provision since it does not require finding new cycles in the network, and this also makes it more suitable for dynamic traffic

The rest of the paper is organized as follows: In Section II, we analyze the p^2-cycle protection scheme in more detail. In Section III, we consider unicast protection problem with static traffic demands using p^2-cycles as the protection method. The problem is formulated as an Integer Linear Program (ILP). In Section IV, we further consider dynamic traffic scenarios, in which two heuristic algorithms are proposed. Performance evaluation of multiple criteria for both static and dynamic traffic scenarios will be presented in Section V. Finally, we conclude the paper in Section VI.

II Overview of p^2-Cycles

In this section, we provide an overview of p^2-cycle protection scheme and elaborate the details of protection mechanism and traffic recovery time.

II.1 Concept

An example is shown in Figure 1 to illustrate the concept of the p^2-cycle. In Fig. 1(a), a p-cycle $(A-B-C-D-E-F-A)$ is used to protect two bidirectional paths, P_1 and P_2, where path P_1 traverses on-cycle span (D, E) and (E, F) and is protected by on-cycle segment $(F-A-B-C-D)$ and path $P_2(A-C)$ is a straddling path that is protected by on-cycle segment $(A-B-C)$. Working paths are denoted by solid lines and protection paths are represented by dashed lines. Assuming we have another working path P_3 (shown in Fig.1(b)) traversing

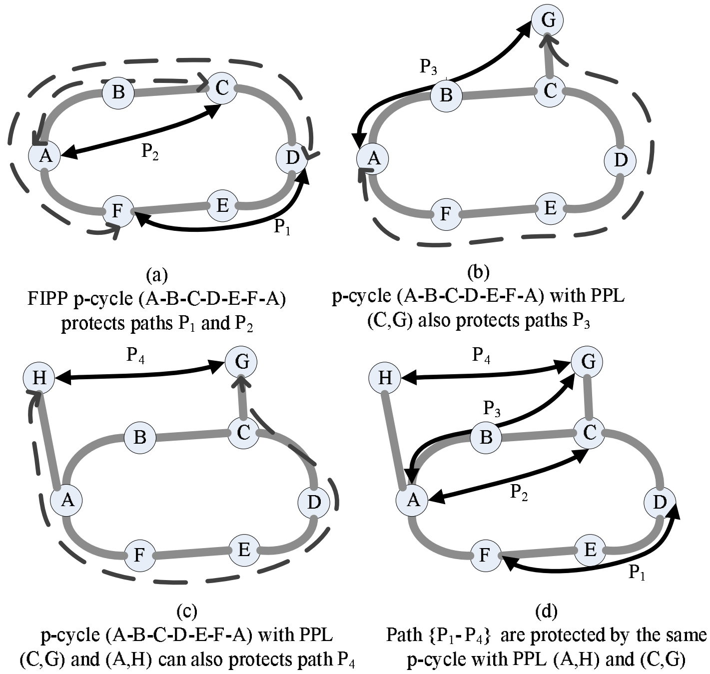


Figure 1: An example of a p-cycle with PPLs

on-cycle span (A, B) and non-cycle span (B, G), the original p-cycle cannot protect it, since the end node G is not on the cycle. We then extend the p-cycle to have a PPL (C, G) and hence protect P_3 by using the path $(A{-}F{-}E{-}D{-}C{-}G)$, which is partly on-cycle and partly on PPL. The idea can also be applied to a path whose two end nodes are not on the cycle, such as path p_4 shown in Fig.1(c). Two PPLs (A, H) and (C, G) can be used to construct the protection path $(H{-}A{-}F{-}E{-}D{-}C{-}G)$. Therefore, the augmented p-cycle with the two links (A, H) and (C, G) can protect four paths (shown in Fig.1(d)). Hence, augmenting a p-cycle to have PPLs enhances the flexibility of protection and thus may decrease spare capacity redundancy and reduce overall capacity cost.

II.2 Protection Mechanism

The protection ability of a p^2-cycle is an enhancement to that of the p-cycle by adding attached spans to the cycle, which enables the cycle to provide protection to the connections whose end nodes are one hop away from the cycle. All the nodes on the cycle still remain pre-configured. For the nodes that also connect to PPLs, they only reconfigure the switches when the attached PPLs are activated to provide protection upon a network failure. Given a unicast session, the primary path and its fully disjoint corresponding protection path, which may consist of an on-cycle segment and one or two PPLs, will be determined in advance regardless of the location of the failure. Hence, the p^2-cycle protection scheme is also failure-independent[17].

Upon a link failure, the failure will be detected by the end nodes of the failed span and the corresponding signals will be transmitted to the source and destination nodes of the path. The distinction between a p^2-cycle and an FIPP p-cycle here is that the source or the destination may not be on the cycle. Therefore, in order to reroute the traffic onto the backup path, the source and destination nodes need to reconfigure their switches, as do the end nodes on the protection cycle that connect to an activated PPL. The rest of the nodes on the protection path are pre-configured. If PPLs are activated in the protection path, an extra local signal may be sent by the adjacent end node to the branch node for guidance of which part of the cycle to be routed.

Let us review the examples in Figure 1. In Figure 1(b), if a failure happens to span (A, B) or (B, G), both end nodes A, G and on-cycle node C will reconfigure their switches to reroute the traffic through the backup path. Similar reconfigurations should be done by node H, G, A and C in Figure 1(c) upon any span failure on the primary path P_4, given that P_3 and P_4 share the same on-cycle paths from C to A through the lower part of the cycle. If they are protected by two separate on-cycle segments, node A and C need a signal sent by H and G, respectively, to indicate which on-cycle segment is used to re-route the traffic. However, such information is only required between the end nodes of the primary paths and the branch nodes on their protection p-cycles if the end nodes are not on the cycle. The information stored at each involved nodes are local and implementation is straightforward on the control plan.

II.3 Traffic Recovery Time

In general, the traffic recovery process consists of three phases: failure detection, fault signaling and switch reconfigurations. However, switch reconfigurations needed for recovery using the protection circuits (either at the end nodes or at intermediate nodes), are usually the most time consuming phase during the process, since each reconfiguration takes 10 - 20s ms [27] depending on the technology used. More node reconfigurations on the protection path will result in longer traffic recovery. Thus, the average *Number of Node Reconfigurations* (NOR) is a key factor to inspect the traffic recovery speed of a given protection scheme [1].

It is apparent that NOR of FIPP p-cycle scheme is always equal to two, one at the source and the other at the destination, and rest of the nodes are pre-configured on the cycle. However, for a path protected by a p^2-cycle, NOR can be two, three or four depending on how many PPLs are used by a protection path. In Fig. 1, path 1 and 2 are protected by the cycle without usage of any PPL and thus the NOR of them is two. However, the NOR of protection path 3 and 4 equals three (A,C and G) and four (A,H,C and G), respectively.

It is worth noting that the distance between any end node of a connection and its protection cycle does not have to be fixed by one hop, such as PPL (A, H) and (C, G) in Fig. 1. Instead, we can extend the capability of a p-cycle to protect the connections whose end nodes are not only one hop but k hops away where $k > 1$. The overall protection capacity efficiency may be enhanced. However, each PPL will easily grow to be a segment that may have multiple links in order to protect more connections. In that case, the length of each protection path may increase and the local routing database may also increase. But more critically, the NOR required on a protection path may increase significantly since every node on a stretched "PPL" may potentially become a reconfiguration node. In addition to the original reconfiguration nodes, such as end nodes or nodes on the cycle that are connected to PPLs, the traffic recovery time may soon become unacceptable due to a large number of reconfigurations. Therefore, in order to achieve great improvement in capacity efficiency without sacrificing too much recovery time, we limit the number of hops of any "PPL" to one link for every p^2-cycle.

II.4 Motivations

As p^2-cycle is derived from FIPP p^2-cycle, which was initially proposed for static traffic protection, p^2-cycle has a natural fit to solve optimization of network provisioning and protection if the traffic demand is given as priori. We will formulate the problem in Section III using Integer Programing and minimize the overall capacity cost using p^2-cycles to protect the provisioned connections.

However, this may not be the case when dealing with dynamic traffic without the prior knowledge of arrival time of future requests. Due to the pre-configuration property of traditional p-cycles, it is very difficult to re-provision all the protection cycles whenever a new session arrives in order to minimize overall cost. Each provisioning takes large computation cost and complex network reconfiguration. Therefore, most of the work in the literature assume that established p-cycles should not vary with time or traffic [29, 30, 31]. The authors in [29] proposed three different routing algorithms along with link-based p-cycle protection scheme to deal with dynamic traffic. They first select an optimal set of cycles and then try to provision incoming traffic as much as possible. The results indicate that the proposed p-cycle based design performs better than SBPP in dense networks but worse in sparse networks. The scheme proposed in [31] includes a connection teardown procedure that computes and releases all p-cycles that are no longer needed, but

[1]The recovery time includes the time taken to detect and signal the failure, the delay to reconfigure switches, and then the difference in propagation delay between the protection and working paths. In this study, we ignore the optical signal propagation delay since it is much lower than switch reconfiguration delays, where the latter is on the order of 10ms to 20ms. Since the time to detect and signal failures is similar between FIPP p-cycles and p^2-cycles, we only concentrate on the number of switch reconfigurations.

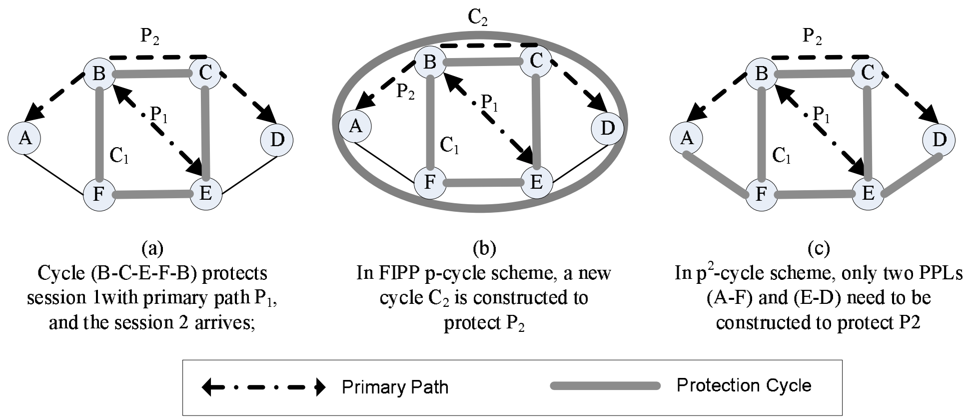


Figure 2: Demonstration of p^2-cycles in dealing with dynamic traffic

p-cycles are not reconstructed in the process once established. The comparison of a number of resilience approaches in protecting dynamic traffic was conducted in [28]. Protected Working Capacity Envelopes (PWCE) is another method to address dynamic traffic scenarios [30, 32]. It divides the entire network into two partitions: working and protection. The working partition provisions the demand of forecasted traffic where as protection partition, a prior of protection structure, provides protection for the traffic. Both static and dynamic traffic can be accommodated as long as the total traffic does not exceed the limit of working envelopes.

Therefore, p-cycles still have such intrinsic weakness in dealing with dynamic traffic. If an incoming session whose end nodes do not lie on any cycle, it cannot be protected and a new cycle has to be constructed to protect this session, or the existing cycles must be reconfigured. An example shown in Fig. 2 illustrates such weakness and also reveals the advantage of p^2-cycles. In Fig. 2(a), session 1 has been provisioned and protected by cycle $C_1(E{-}C{-}B{-}F{-}E)$. As session 2 arrives, the primary path of session 2 is provisioned as $P_2(A{-}B{-}C{-}D)$. Under FIPP p-cycle scheme, cycle C_1 cannot protect it and thus a new cycle $C_2(A{-}B{-}C{-}D{-}E{-}F{-}A)$ is constructed to protect it as shown in Fig. 2(b). However, instead of building a new cycle, using p^2-cycle approach we can add two PPLs (A,F) and (D,E) to connect the end nodes of P_2 such that C_1 can also provide a protection segment $(A{-}F{-}E{-}D)$ for P_2 as shown in Fig. 2(c). Therefore, both sessions are protected by a p^2-cycle with much less cost (a total of 6 links compared to 10 links).

III Static Traffic Scenarios

In this section, we address the static unicast services protection problem using p^2-cycle protection scheme in WDM networks against single-link failure scenarios. The problem will be defined first and then formulated as an Integer Linear Program (ILP).

III.1 Problem Statement

Given a number of static traffic requests, the problem can be solved in two ways. The first method is to divide the problem into two sub-problems: working paths provisioning and protection cycles provisioning. Then the two sub-problems are solved sequentially. However, the optimality of the solution may be compromised. In order to study the overall optimal performance of the p^2-cycle scheme, we address the joint capacity placement (JCP) problem in which working paths and protection cycles are provisioned jointly such that the minimum total cost is achieved.

A number of assumptions are given as follow, in which a session refers to a provisioned traffic demand or request:

1. Each unicast session is bidirectional with a unitary traffic rate (one wavelength) and the traffic in both directions has to be routed through the same paths and protected by the same p^2-cycle.

Input:	
P	The maximum number of p^2-cycles in the solution
p	an index that refers to the pth p^2-cycle where $1 \le p \le P$
l	an index that refers to the lth session where $1 \le l \le M$
Variables:	
f^l_{mn} :	A binary variable, equals 1 if the primary path of session l traverses span $(m, n) \in E$
q^l_{mn} :	A binary variable, equals 1 if the protection flow of session l traverses span $(m, n) \in E$
Zf^l_n :	A binary variable, equals 1 if the primary path of session l traverses node n
Zq^l_n :	A binary variable, equals 1 if the protection path of session l traverses node n
e^p_{mn} :	A binary variable, equals 1 if p^2-cycle p traverses span (m, n)
z^p_n :	a binary variable, equals 1 if p^2-cycle p traverses node n
b^p_{mn} :	A binary variable, equals 1 if span (m, n) is a PPL of p
$B^{p,l}_{mn}$:	A binary variable, equals 1 if PPL (m, n) is used by p to protect session l
x^p_l :	A binary variable, equals 1 if p protects session l
$X^p_{l_1 l_2}$:	A binary variable, equals 1 if session l_1 and l_2 are both protected by p
$\phi^{l_1 l_2}$:	A binary variable, equals 1 if session l_1 and l_2 share protection of any p^2-cycle
$\gamma^{l_1 l_2}_{mn}$:	A binary variable, equals 1 if the primary paths of both session l_1 and l_2 use span (m, n)
$\Gamma^{l_1 l_2}$:	A binary variable, equals 1 if the primary paths of session l_1 and l_2 use at least one common span
μ^p_u :	A binary variable, equals 1 if node u is the master node of p
$\alpha^p_{mn,v}$:	A binary variable, equals 1 if span (m, n) is used to reach node v from the master node of p
$\beta^p_{n,v}$:	A binary variable, equals 1 if node n is traversed by the flow from the master node to node v through the cycle of p
ϵ :	A small positive constant (0.0001)

2. Each p^2-cycle is also bidirectional and has unitary capacity on both on-cycle spans and PPLs.

3. Each span has enough wavelengths and each node is equipped with wavelength converters over all wavelengths, such that wavelength continuity is not required in the network.

We now state the JCP problem formally: Given a bidirectional unicast traffic matrix D where $D = d_l(s_l, t_l)$, $(0 \le l < M)$ where M is the number of connections, and a weighted undirected graph $G = (V, E)$ in which each span $e \in E$ has a cost c_e, provision and protect all the unicast sessions with minimal total cost (the cost consumed on each span is the product of the total number of wavelengths used over the span and the unit cost over the span).

III.2 ILP Formulation

We formulate the JCP problem as an ILP. Since the number of cycles increases exponentially with network sizes, we do not enumerate all the cycles in a given network in the formulation. Instead, the flow variables will form the cycles in the solution. The input parameters and decision variables used in the ILP are defined in the following table.

The objective function is:

$$\text{Minimize:} \quad \sum_{(m,n)\in E} c_{mn} \left(\sum_{0\le l<M} (f^l_{mn} + \sum_{0\le p<P} (e^p_{mn} + b^p_{mn})) \right)$$

Each span between node m and n is denoted by (m, n) where $m<n$ in the network $G=(V, E)$, in which each wavelength cost of span $e \in (m, n)$ is c_{mn}. The objective function minimizes the total cost consumed by the primary paths (first term) and the p^2-cycles used to protect them. Each p^2-cycle is composed of on-cycle spans e (second term) and parasitic protection spans b (third term) .

The constraints are such that:

1. **Flow Conservation Constraints:**

$$\begin{aligned}
&\text{For } n \in V\backslash\{s_l, t_l\},\ \forall l: \\
&\sum_{n:(s_l,n)\in E} f^l_{s_l,n} = \sum_{m:(m,t_l)\in E} f^l_{m,t_l} = 1; && (1)\\
&\sum_{n:(u,n),(n,u)\in E} f^l_{u,n} = 2Zf^l_n, && (2)\\
&\sum_{n:(s_l,n)\in E} q^l_{s_l,n} = \sum_{m:(m,t_l)\in E} q^l_{m,t_l} = 1; && (3)\\
&\sum_{n:(u,n),(n,u)\in E} q^l_{u,n} = 2Zq^l_n; && (4)
\end{aligned}$$

Equations (1)-(4) ensure that each session l has a primary and a protection path. The source and destination nodes of the session connect to only one span used by each path, but each intermediate node is connected by two adjacent spans.

2. **Protection Constraints:**

$$\begin{aligned}
&\sum_p x^p_l = 1, \quad \forall l; && (5)\\
&e^p_{mn} \ge x^p_l \wedge q^l_{mn}, \\
&\quad \forall p, l,\ \forall (m, n) \in E,\ m, n \ne \{s_l, t_l\}; && (6)\\
&B^{p,l}_{mn} = x^p_l \wedge q^l_{mn} \wedge (1 - e^p_{mn}), \\
&\quad \forall p, l,\ \forall (m, n) \in E; && (7)\\
&b^p_{mn} \ge \epsilon \left(\sum_l B^{p,l}_{mn} \right),\ \forall p, \forall (m, n) \in E; && (8)
\end{aligned}$$

Equation (5) ensures that each session is protected exactly once by a p^2-cycle. If session l is protected by each p and the protection flow uses span (m, n), then span (m, n) should be an on-cycle span of p except that m or n is the source or the destination of session l. In that case, (m, n) can be a PPL. This constraint is ensured by equation (6), in which the symbol $\wedge$ denotes a conjunction operation. A conjunction expression $X = \bigwedge_{1\le i\le N} x_i$ can easily be represented by two linear equations $X \le \frac{1}{N}(\sum_i x_i)$ and $X \ge \sum_i x_i - N + 1$, given binary variables X and x_i.

Equation (7) ensures that if a span (m, n) is used by a protection flow to protect session l, it should be a part of p^2-cycle p. However, if it is not an on-cycle span, it must be a PPL, which is denoted by $B^{p,l}_{mn}$. A PPL can be used to protect multiple connections. As long as there exists at least one connection using span (m, n) as a PPL of p, the span (m, n) is counted as a PPL of p. Equation (8) ensures this constraint.

3. **Link Disjointness Constraints:**

$$f_{mn}^{l} + q_{mn}^{l} \le 1, \quad \forall l,\ \forall (m,n) \in E; \tag{9}$$

The working and backup paths of any session l should be link-disjoint to survive any single-link failure. This is ensured by equation (9).

4. **Protection Capacity Sharing:**

For $\forall p,\ \forall l_1, l_2, l_1 < l_2 < M,\ \forall (m,n) \in E$:

$$X_{l_1 l_2}^{p} = x_{l_1}^{p} \wedge x_{l_2}^{p}; \tag{10}$$

$$\phi^{l_1,l_2} \ge \epsilon \left(\sum_{p} X_{l_1 l_2}^{p} \right); \tag{11}$$

$$\gamma_{mn}^{l_1,l_2} = p_{mn}^{l_1} \wedge p_{mn}^{l_2}; \tag{12}$$

$$\Gamma^{l_1,l_2} \ge \epsilon \left(\sum_{(m,n)\in E} \gamma_{mn}^{l_1,l_2} \right); \tag{13}$$

$$q_{mn}^{l_1} + q_{mn}^{l_2} \le 2 - \left(\phi^{l_1,l_2} \wedge \gamma^{l_1,l_2} \right); \tag{14}$$

Equations (10) ensures that if two different session l_1 and l_2 are protected by the same p, then $X_{l_1,l_2}^{p} = 1$. If they share any p^2-cycle p, then $\phi^{l_1 l_2} = 1$ as shown in equation (11). Equation (12) and (13) make sure that $\Gamma^{l_1 l_2} = 1$ if the primary paths of session l_1 and l_2 are not link disjoint. In this case, if l_1 and l_2 also share the protection of the same p, the protection flow of l_1 and l_2 cannot traverse the same span, which is ensured by equation (14).

5. **Cycle Constraints:**

$$\sum_{n:(m,n)\in E} e_{mn}^{p} = 2 z_{n}^{p},\ \forall n \in V,\ \forall p; \tag{15}$$

$$|z_{m}^{p} - z_{n}^{p}| \ge b_{mn}^{p}, \quad \forall (m,n) \in E,\ \forall p; \tag{16}$$

The cycle constraints make sure that each node on the cycle is passed twice by on-cycle spans, as described in equation (15). If span (m,n) is a PPL of p, then one of m and n must lie on the cycle while the other not, which is given by Equation (16). A new variable can be introduced to replace the absolute expression with two linear equations. For instance, $Z_{m,n}^{p}$ can be used to replace $|z_{m}^{p} - z_{n}^{p}|$ with two new constraints: $Z_{m,n}^{p} \ge z_{m}^{p} - z_{n}^{p}$ and $Z_{m,n}^{p} \ge z_{n}^{p} - z_{m}^{p}$. The minimization of the cost of connection provisioning in the objective function will ensure that Z_{mn}^{p} is equal to $|z_{m}^{p} - z_{n}^{p}|$.

6. **Cycle Uniqueness:**

$$\sum_{u\in V} \mu_{u}^{p} = 1, \quad \forall p; \tag{17}$$

$$\sum_{n:(m,n)\in E} \alpha_{mn,v}^{p} = 2\beta_{m,v}^{p} - \mu_{m}^{p}, \quad \forall p,\ \forall \{m,n,v\} \in V,\ m \ne v; \tag{18}$$

$$\sum_{m:(m,v)\in E} \alpha_{mv,v}^{p} = 1 - \mu_{v}^{p},\ \forall p, \forall v \in V; \tag{19}$$

$$e_{mn}^{p} \ge \alpha_{mn,v}^{p} - \epsilon \beta_{m,v}^{p}, \quad \forall p,\ \forall (m,n) \in E, \forall v \in V; \tag{20}$$

However, equation (15) is not enough to guarantee that there is only one cycle with index p, since multiple cycles can be formed with the same index p while still complying with constraint (15). Some work has been done to address this issue. The method proposed in [14] is simple and the number of introduced variables is linear in the size of the network. However, it can only apply to unidirectional cycles. Hence, we use the approach proposed in [18]. This approach picks a node on each cycle randomly and defines it as the master node such that there must exist a flow from the master node to every other on-cycle node through the cycle.

Equation (17) ensures that there is only one unique master node for each p^2-cycle where node u is the master node of p. Equations (18) and (19) ensure the flow conservation between the master node and all other on-cycle nodes. Equation (18) guarantees that the flow uses one span connected to the master node but two connected to any intermediate node passed by the flow and Equation (19) ensures that only one span connecting to the destination node is used by the flow on the cycle. Equation (19) also guarantees that if a node v is on the cycle p, then all the spans traversed by the flow from the master nodes to node v should be on-cycle spans of p. Therefore, the uniqueness of one cycle for each index p is ensured. The number of variables introduced to guarantee a single cycle for each p is $P|E||V| + P(|V|+1)|V|$.

In the worst case, we need one cycle to protect each connection without any sharing. Therefore, we chose $P = M$ in the ILP formulation to guarantee optimal solution. The total number of variables used is dominated by $M^2(|E| + P) + P|V|(|E| + |V|)$ and the total number of constraints can be denoted by $O(M^2(|E|+P) + P(|E||V| + |V|^2 + M|E|))$.

In addition, this ILP formulation can be easily extended to address other network scenarios or design goals. For instance, instead of minimizing total capacity usage, it can be used to minimize only the spare capacity used by protection p^2-cycles given the primary path for each session. Accordingly, all the primary path variables f^l_{mn} become input parameters and will be removed from the objective function. Besides, if the design goal is to balance the traffic load by minimizing the maximum capacity consumed on any span in a network, we just need to introduce a new variable, say ζ, to substitute the original objective function such that $\zeta \geq \sum_{0\leq l<M} f^l_{mn} + \sum_{0\leq p<P} (e^p_{mn} + b^p_{mn})$. In this case, ζ denotes the maximum capacity required on any link.

In terms of the change of network scenarios, for example, if each span has an upper bound on the number of wavelengths, denoted by λ_{mn}, we can add a constraint: $\lambda_{mn} \geq \sum_{0\leq l<M} f^l_{mn} + \sum_{0\leq p<P} (e^p_{mn} + b^p_{mn}),\ \forall(m,n) \in E$ to ensure that the total capacity reserved does not violate the upper bound. Therefore, our ILP model can be flexibly modified to adapt it to various network scenarios and different design goals.

IV Dynamic Traffic Scenarios

In this section, the dynamic traffic scenario will be addressed. We first formally state the problem and then propose two heuristic algorithms based on p^2-cycle to study the problem.

IV.1 Problem Statement

In dynamic traffic scenarios, a WDM mesh network is given with network resources, such as the maximum number of wavelengths and the cost on each span. Each traffic request arrives to the network in a dynamic fashion such that it needs to be considered individually based on the current network status. The network status consists of the detailed working and available wavelengths on each span as well as all the accepted sessions and p^2-cycles provisioned in the network. Hence, the dynamic traffic protection problem can be defined as follows:

Given a network modeled as an undirected graph $G = (V, E)$ where each span $e \in E$ has a cost c_e, the current network which includes the currently used and available wavelengths on each span e, each accepted session l and their protection p^2-cycles. Provision incoming unicast sessions against any single-link failure with the minimum overall blocking probability by using p^2-cycle scheme. The assumptions required in this dynamic traffic case are the same as that in the static case defined in Section III.

Table 1: Symbols used in the algorithms

Notations	Meaning
$\mathbb{D}$	The set of sessions that are active in the network
$\mathbb{C}$	The set of existing p^2-cycles in the network
$\mathbb{P}$	The set of primary paths of the active sessions in the network
$\mathbb{T}$	A temporary set of the combination of $< c_p, q_l, f_l >$
$d_l(s_l, t_l)$	An incoming traffic request l with end nodes s_l and t_l, stored in $\mathbb{D}$ if it is accepted
$c_p \in \mathbb{C}$	The pth p^2-cycle in $\mathbb{C}$
$f_l \in \mathbb{P}$	The primary path of traffic request l
q_l	The protection path of traffic request l
$\delta(d_l, c_p)$	Integer, the distance between the end nodes of a connection d_l to the cycle of c_p.

IV.2 Heuristic Algorithms

We design two heuristics to address the dynamic traffic case. In the first method, named Strict Routing Protection (SRP), the primary and protection path for each incoming session are computed separately. The primary path is firstly provisioned using Dijkstra's shortest routing algorithm. Based on the primary path, either an existing p^2-cycle or a new cycle is found to protect it. In the second method, named Flexible Routing Protection (FRP), the primary and protection paths of an incoming session are constructed jointly. The existing p^2-cycles will be preferred to being used first. If no existing one is able to protect the session, a new cycle will be formed. The notations used in the algorithms are explained in Table 1.

IV.2.i Strict Routing Protection (SRP)

The motivation of SRP is to always choose the shortest path to route the primary traffic in order to leave more spare capacity for protection, since the capacity used for primary path cannot be shared among different sessions. And then we check whether any available p^2-cycle can be exploited to protect this newly established session. Once being set up, the cycle for a p^2-cycle can not be changed. However, PPLs may be added for protecting the connections whose end nodes are one hop away from the cycle. The detail of the algorithm SRP is shown in Algorithm 1, in which the process can be described in following steps:

1. As a new session $d_l(s_l, t_l)$ arrives, establish the primary path f_l between s_l and t_l under current network status by using Dijkstra's algorithm. If it fails, the session is blocked;
2. Sort all the existing p^2-cycles, $c_p \in \mathbb{C}$, in the increasing order of $\delta(d_l, c_p)$, which is computed as follows:

$$\delta(d_l, c_p) = \begin{cases} 0, & \text{if both } s_l \text{ and } d_l \text{ are the on-cycle} \\ & \text{nodes of } c_p; \\ 1, & \text{if one of } s_l \text{ and } d_l \text{ is on the cycle} \\ & \text{and the other is one hop away from} \\ & \text{the cycle;} \\ 2, & \text{if both } s_l \text{ and } d_l \text{ are one hop} \\ & \text{away from the cycle of } c_p; \\ +\infty, & \text{otherwise.} \end{cases}$$

   One hop indicates that there exists a span in the network that connects a node to the cycle. If $\delta(d_l, c_p) = +\infty$ for all $c_p \in \mathbb{C}$, then no existing cycle is able to protect this new session. Thus, a new cycle needs to be constructed to protect d_l.
3. For each existing protection cycle, c_p, we construct a temporary graph G', consisting of only the cycle spans of c_p and all the spans connecting the source and destination nodes of l to the cycle (line 4).

Algorithm 1: Strict Routing Protection (SRP) Scheme

Input: $G(V, E), \mathbb{D}, \mathbb{P}, \mathbb{C}$
Output: Accepted or Blocked?

Given a new session l, find the shortest path f_l in G;
sort $c_p \in \mathbb{C}$ in the increasing order of $\delta(t_l, c_p)$;
for $c_p \in \mathbb{C}$ *and* $\delta(d_l, c_p) < 3$ **do**
  construct a temporary graph $G' = (V', E')$ where $V' = \{\forall v \in c_p\} \vee \{s_l, t_l\}$ and $E' = \{\{\forall e \in c_p\} \vee \{(s_l, v), (v, d_l)\}\} \backslash \{\forall e \in f_l\}$;
  for $d_i \in \mathbb{D}$ *protected by* c_p **do**
    if f_l *and* p_m^i *are not link disjoint* **then**
      $E' = E' \backslash \{e \in p_t^i\}$;
    end
  end
  Run Dijkstra's algorithm to find a protection path q_l between s_l and t_l in G';
  if *Succeed* **then**
    accept session l and update c_p by adding PPLs $e \in q_l$ but $e \notin c_p$;
    update $\mathbb{P}$ and G and exit;
  end
end
if f_l *can not protected by any* c_p **then**
  establish form a new cycle, $c_{|\mathbb{C}|+1}$, to protect f_l;
  if *Succeed* **then**
    add f_l to $\mathbb{P}$ and $c_{|\mathbb{C}|+1}$ to $\mathbb{C}$;
  else
    the request l is blocked;
  end
end

All the spans used by f_l should be removed to ensure that its protection path is link-disjoint. Then, all the sessions protected by c_p are checked and if an existing session in $\mathbb{D}$ can share the same c_p with the new session l, we should make sure that either their primary paths or their protection paths are link-disjoint. In lines 5-9, we remove the protection paths of all the sessions in $\mathbb{D}$ whose primary paths are not link-disjoint with f_l. If a protection path can still be found in the remaining G' (line 10), this protection path will be q_l for l. Accordingly, the protection cycle is also determined, which should be updated if some PPLs are also used (lines 11-14).

4. If every existing c_p fails to protect d_l, a new cycle will be constructed to protect it. We first attempt to find two diverse paths to form a cycle that is link-disjoint to f_l. If such cycle cannot be found, then we find a path, q_l, link-disjoint to f_l and the cycle is formed by combining q_l with f_l. This last part of the algorithm is described by lines 16-22.

In the worst case, every existing sessions d_i will be examined as to whether its primary path is link-disjoint with the primary path of the new session. It takes total of $O(|\mathbb{D}||E|)$ times for the checking process. All existing p^2-cycles may also be checked. For each c_p, the computation cost of graph and path construction is $O(|E| + |V|^2)$. Therefore, the time complexity of SRP algorithm in the worst scenario is dominated by $O(|\mathbb{C}|(|\mathbb{D}||E| + |V|^2))$.

IV.2.ii Flexible Routing Protection (FRP)

Different from SRP, the flexible routing protection scheme considers primary and protection paths jointly for each arriving session. Instead of determining the primary path in advance, we examine each existing p^2-cycle and find each potential protection path along the cycle that can connect the source and destination. For each potential protection path, we try to discover a primary path for it. If it succeeds, the session is accepted. Otherwise, a new cycle is constructed to protect the session. The detailed procedure is described in Algorithm 2.

Algorithm FRP is explained in following steps:

1. Given a new session $d_l(s_l, t_l)$, all the available p^2-cycles $c_p \in \mathbb{C}$ are sorted in the increasing order of $\delta(d_l, c_p)$.

2. For each available c_p, list all the possible protection paths for d_l. If the end nodes s_l and t_l are on the cycle, there are two possible segments along the cycle. If s_l or(and) t_l is not on the cycle, the path will be composed of parasitic links connecting s_l or t_l to the cycle and an on-cycle segment.

   We assume the average node degree in a given network is denoted by θ. Each cycle can provide two on-cycle segments between any pair of on-cycle nodes. Each end node, s_l or t_l, can be connected to the cycle by at most θ PPLs given the node degree θ. Hence, the average number of candidate protection paths provided by any p^2-cycle for d_l is $2\theta^2$. Lines 3-7 are used to construct all candidate protection paths for d_l.

3. For each candidate q_l, run Dijkstra's algorithm to find a primary path f_l in G that is not only link-disjoint to q_l but also link-disjoint with other primary paths protected by the same cycle if their protection paths are not link-disjoint. If it succeeds, we store the combination $< c_p, q_l, f_l >$ in a temporary set $\mathbb{T}$, which is initialized as $\emptyset$. After checking all the existing p^2-cycles, we check set $\mathbb{T}$ and find the combination $< c_p, q_l, f_l >$ with minimum cost of f_l. We recover the spans removed from G and update the network status. The process is described by lines 8-18.

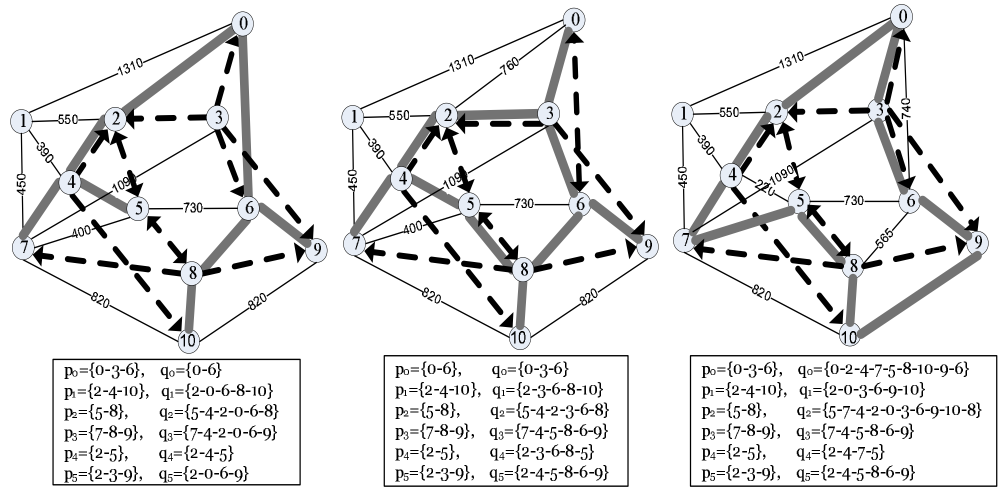
4. If no existing p^2-cycle can be used to protect session d_l, we use Bhandari's algorithm to find two link-disjoint paths between s_i and t_i to form a new p^2-cycle. If it fails, the session is blocked. Otherwise, the session is accepted and one of the paths (usually the shorter one) is used as the primary path f_l, and the network is updated.

In the worst case, all the existing p^2-cycles will be examined and the total number of protection path candidates provided by a cycle is $2\theta^2$. It takes $2|V|$ operations to check all the candidates of protection path q_l, from line 3 to line 7. For each candidate, a total of $O(|\mathbb{D}||E|)$ time is consumed by checking link-disjointness between q_l provided by each c_p and the protection paths of other sessions. In addition to the

Algorithm 2: Flexible Routing Protection (FRP) Scheme

Input: $G(V, E), \mathbb{D}, \mathbb{P}, \mathbb{C}, \mathbb{T} = \emptyset$
Output: Accepted or Blocked?

Given a new session $d_l(s_l, t_l)$, sort $c_p \in \mathbb{C}$ in the increasing order of $\delta(d_l, c_p)$;
for $c_p \in \mathbb{C}$ *and* $\delta(d_l, c_p) < 3$ **do**
    if s_l *or(and)* t_l *is not on the cycle of* c_p **then**
        construct each candidate q_l by combining PPL (s_l, u) or(and) (v, t_l) and an on-cycle segment between on-cycle node u and v;
    else
        construct candidate q_l by using an on-cycle segment of c_p between s_l and t_l;
    end
    for *each candidate* q_l **do**
        for $d_i \in \mathbb{D}$ *protected by* c_p **do**
            remove the primary path f_i from G if q_i is not link-disjoint with q_l;
        end
        remove q_l and run Dijkstra's Algorithm to find f_l between s_l and t_l in G;
        if *Succeed* **then**
            add the combination $< c_p, q_l, f_l >$ to $\mathbb{T}$;
        end
    end
end
if $\mathbb{T}$ *is not empty* **then**
    Session l is accepted and choose $< f_l, q_l, c_p >$ with the minimum cost of f_l;
    Add f_l to $\mathbb{P}$ and $e \in q_l$ but $e \notin c_p$ to c_p;
end
else
    Run Bhandari's Algorithm[33] to find two link-disjoint paths between s_l and t_l;
    Choose shorter one as f_l and combine them as a new p^2-cycle $c_{|\mathbb{C}|+1}$;
    if *Failed* **then**
        The request is blocked;
    end
end



(a) solution of SBPP protection scheme with total cost 8695.0 and running time 3336.34 sec

(b) solution of *p2*-cycle protection scheme with total cost 8945.0 and running time 814.38 sec

(c) solution of FIPP *p*-cycle protection scheme with total cost 9470.0 and running time 111.11 sec

Figure 3: Comparison of the total cost of six unitary unicast sessions: $\{(0,6), (2,10), (5,8), (7,9), (2,5), (2,9)\}$ in COST239 network using: (a) SBPP, (b) p^2-cycle and (c) FIPP

time of running Dijkstra's algorithm, $O(|V|^2)$, lines 8-18 consume a total time; that is $O(2\theta^2(|\mathbb{D}||E|+|V|^2))$. This part of the algorithm actually dominates the running time. Therefore, adding up the cost for each iteration of c_p, the total time complexity of FRP algorithm is $O(2\theta^2|\mathbb{C}|(|\mathbb{D}||E|+|V|^2))$. As we can see, FRP has higher time complexity than SRP due to multiple options of each primary path.

V Performance Evaluation

In this section, we investigate the performances of the proposed p^2-cycle protection scheme under both static and dynamic traffic scenarios. In the static traffic case, the unicast traffic requests are given in advance. We compare p^2-cycle performance with two other path protection schemes, the SBPP and FIPP p-cycle, in terms of two criteria: total capacity cost and average number of reconfigurations (NOR). We also study the performance of p^2-cycle scheme in handling dynamic traffic demands by using the proposed heuristic algorithms, SRP and FRP, and compare them with FIPP p-cycle in terms of overall blocking probability and average NOR.

V.1 Results for Static Traffic

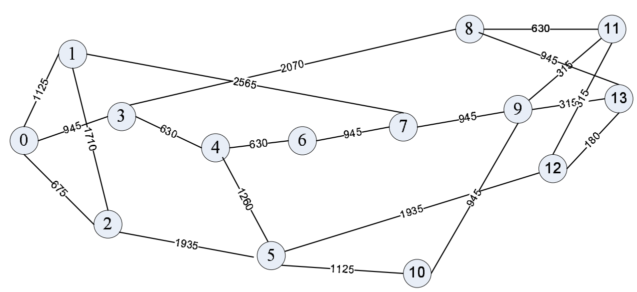


Figure 4: NSFNET network (14 nodes, 21 spans)

With the prior knowledge of all the traffic information, we obtain the optimal solutions of the JCP problem by formulating it as an ILP and solving it by a commercial software - ILOG CPLEX 10.1.0 on a Linux server

Table 2: Comparison of average total capacity cost in NSFNET

No. Sess.	SBPP	p^2-cycle (extra cost(%))	FIPP (extra cost(%))
2	10734.8	12335.4 (14.9)	13310.1 (24)
3	14774.7	16060.5 (8.7)	17515.8 (18.6)
4	19146.1	19759.5 (3.2)	21185.1 (10.6)
5	21818.3	22624.2 (3.7)	24122.7 (10.6)
6	25539.2	26395.2 (3.4)	27514.8 (7.7)
7	29525.2	30327.5 (2.7)	31518.5 (6.8)
8	32610.8	33570.1 (2.9)	34567.3 (6.0)
9	35905.4	36806.2 (2.5)	37890.5 (5.5)
10	39792.5	40736.2 (2.3)	42100.5 (5.8)

with four Xeon 2.4GHz CPU and 4 GB of RAM. The ILP for SBPP is obtained from [4]. Since the ILP for FIPP p-cycle proposed in [17] does not address JCP problem but only spare capacity assignment, we use the ILP proposed in this paper without using PPLs.

The experiments are conducted on two practical networks, NSFNET and the pan-European COST 239, shown in Fig. 4 and 3, respectively. Both networks have similar numbers of nodes, but COST239 has a larger average node degree (4.7) than NSFNET (3.0). Each span in the two networks has a cost, which is the actual distance between the two end nodes in kilometers. We assume that the networks have wavelength conversion capabilities and unlimited wavelengths on each span.

V.1.i Total Capacity Cost

We first study a special case in COST239 network in which six unicast traffic requests with unitary traffic rate (one wavelength) need to be provisioned, and we obtain the solutions of each protection scheme, as shown in Fig. 3. The source and destination of each session is depicted in a pair of parenthesis, indexed from 0 to 5, counted from left to right. The routes of the primary path, denoted by p, and the corresponding protection path, denoted by q, of each session are described in the boxes.

The optimal solution obtained by employing the SBPP, p^2-cycle and FIPP p-cycle are presented in Fig. 3(a), (b), (c), respectively. One wavelength assigned for protection on each span can be shared by multiple sessions. For instance, in Fig. 3(a), the spare capacity on span $(6, 8)$ is shared by sessions 1 and 2 and $(0, 6)$ is shared by sessions 0,1,2,3 and 5. This feature makes SBPP the most capacity efficient scheme but also takes the longest time to obtain the optimal solution due to the high complexity. The p^2-cycle scheme uses more capacity than SBPP but less than FIPP p-cycle. The FIPP scheme uses 8.9% more capacity over the optimal solution achieved by SBPP whereas the p^2-cycle reduces this number to 2.9%, which is very close to the optimal solution, because it utilizes a combination of a smaller cycle and a number of PPLs compared to a large cycle provisioned by FIPP.

We also studied the average performance of each scheme in both NSFNET and COST239 networks and the results are presented in Table 2 and 3, respectively. Nine different traffic scenarios are simulated, in which a total of 2 to 10 sessions are provisioned. In each scenario, we ran 50 independent cases and then took the average value of the total cost. The end nodes of each session were randomly chosen, but the three schemes use exactly the same traffic demands in each case in order to make a fair comparison.

The first column denotes the number of sessions. In the third and fourth column, the extra cost over the optimum is calculated as $(cost - optimum)/optimum$, where optimum is achieved by SBPP. We can observe that the p^2-cycle always achieves better results than FIPP in each scenario. The capacity efficiency of cycle-based protection schemes increases as the number of sessions increases, since there are not enough connections to share the protection of cycles when the traffic is low. As the number of sessions increases, a cycle is more likely to protect multiple connections and become more capacity efficient. When the number of sessions is large in both networks, the p^2-cycle becomes extremely efficient and only uses less than 1% extra cost over the optimal solution. Hence, one may conclude that p^2-cycle will be extremely close to the optimal solutions as the traffic keeps on increasing.

Table 3: Comparison of average total capacity cost in COST239

No. Sess.	SBPP	p^2-cycle (extra cost(%))	FIPP (extra cost(%))
2	3602.3	4261.7 (18.3)	4669.6 (29.6)
3	4888.3	5496.3 (12.4)	5820.8 (19.1)
4	6268.2	6794.6 (8.4)	7094.5 (13.2)
5	7530.0	7963.9 (5.8)	8277.6 (9.9)
6	8468.8	8674.8 (2.4)	8947.8 (5.7)
7	9545.0	9735.5(2.0)	9945(4.2)
8	10614.1	10789.3(1.7)	10973.5(3.3)
9	11951.6	12104.5(1.3)	12285.7(2.8)
10	13596.5	13710.7(0.9)	13832(1.7)

Table 4: Comparison of average NOR per connection in NSFNET

No. of Sessions	SBPP	p^2-cycle	FIPP
2	2.48	2.47	2
3	2.77	2.55	2
4	2.84	2.42	2
5	3.01	2.43	2
6	3.09	2.34	2
7	3.19	2.31	2
8	3.26	2.32	2
9	3.47	2.33	2
10	3.61	2.31	2

V.1.ii Average Number of Reconfigurations

We also compare the traffic recovery performance of p^2-cycles to the other two protection schemes in terms of the average number of reconfigurations (NOR) per connection. It is straightforward to obtain the NOR for each connection protected by FIPP p-cycle and p^2-cycle schemes, respectively, given the primary and protection paths for each connection. But it is not as straightforward for SBPP scheme due to the complex protection structure and capacity sharing. Given a session with its primary and protection paths, if there are any links on the protection path whose the protection wavelengths are shared with other sessions, the nodes adjacent to these links on the protection path are potential reconfiguration nodes. In order to further decide whether such nodes need reconfiguration upon the failure on the primary path, we combine all the protection paths that share the same wavelengths into a protection structure. A potential node requires reconfiguration upon a network failure on the primary path if its nodal degree is greater than 2 in this protection structure, since this node needs to reconfigure its switch to establish protection paths for different sessions that share the same backup resource on the same link. We obtain the NOR under SBPP by assigning a specific wavelength for each protection unit.

The results of the average NOR in both NSFNET and COST239 networks under SBPP, p^2-cycle and FIPP are presented in Table 4 and 5. The results are obtained by taking the average value over 50 independent cases in each traffic scenario. Clearly, FIPP achieves the best solution since it always takes only two end nodes to reconfigure upon a failure. On the other hand, the average NOR of SBPP increases as the number of connections increases, since the structures gets more complex due to sharing backup resources and results in more potential reconfiguration nodes. On the contrary, the average NOR of the p^2-cycle scheme actually decreases. One of the reasons is that a larger number of connections usually results in cycles with larger size. Hence, more nodes will be covered by the cycle such that fewer connections will use PPL as a part of protection path. Thus, more sessions require only two node reconfigurations upon a failure.

Based on the results, except for the first two scenarios Table 5, the p^2-cycles always perform better than SBPP in terms of NOR and the advantage becomes more significant and gets closer to that of FIPP as the number of sessions increases. In particular, when there are 10 connections in NSFNET, the NOR of p^2-cycles is equal to 2.26, which is only 15% more than the optimal number of 2, achieved by the FIPP p-cycle,

Table 5: Comparison of average NOR per connection in COST239

No. of Sessions	SBPP	p^2-cycle	FIPP
2	2.36	2.87	2
3	2.43	2.76	2
4	2.66	2.63	2
5	2.80	2.52	2
6	2.89	2.42	2
7	2.85	2.46	2
8	2.88	2.45	2
9	3.10	2.32	2
10	3.15	2.26	2

compared to 3.61, obtained from SBPP. In practice, networks usually accommodate much more than three connections. Thus, we can predict that p^2-cycle will perform very close to FIPP p-cycle in terms of NOR when the traffic demands become large.

V.2 Results for Dynamic Traffic

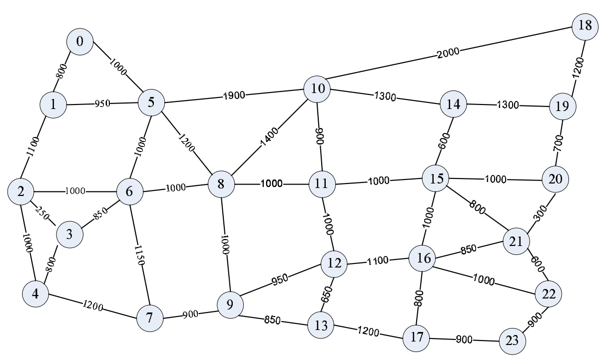


Figure 5: USNET network (24 nodes, 43 spans)

Based on two p^2-cycle protection algorithms, SRP and FRP, proposed for provisioning dynamic requests, we conduct a simulation study to compare the performance of these algorithms under dynamic traffic. The networks used in the simulations are NSFNET, COST239 and USNET, in which USNET network, shown in Fig. 5 has 24 nodes and 43 edges and the average node degree is 3.58.

In each simulation run, 1000 randomly generated unicast requests are loaded to the network sequentially and the reject ratio is recorded. The arrival of traffic follows Poisson distribution with λ requests per second and the duration of an accepted connection is exponentially distributed with a mean of μ. The traffic load measured in Erlangs is $\lambda\mu$. Each connection requires an entire wavelength to transmit the traffic. The maximum capacity on each network link is set to 16 wavelengths.

Figures 6, 7 and 8 show the blocking probability of dynamic traffic using SRP, FRP and FIPP p-cycle in NSFNET, COST239 and USNET networks, respectively. Usually network operators are not interested in the scenarios where overall blocking probability is greater 10%, therefore we only chose to show the scenarios with BP no more than 10%. BP denotes Blocking Probability. Each point in the figures is the average value of 100 simulation runs for each traffic load. For FIPP p-cycle scheme, the primary path of each arriving connection is provisioned first by using Dijkstra's algorithm, and then protected by a p-cycle.

The results show that each scheme has advantages in a particular network based on the results on the given scenarios. In NSFNET, SRP achieves better performance than the other two schemes although FIPP performs slightly better at first two scenarios when the traffic is very low. As traffic load increases over 30 Erlangs, BP of FIPP increases much faster. At the scenario where traffic load equals 50 Erlangs, BP of SRP achieves 25% less than that of FIPP. In USNET, FRP outperforms FIPP under every scenario and achieves

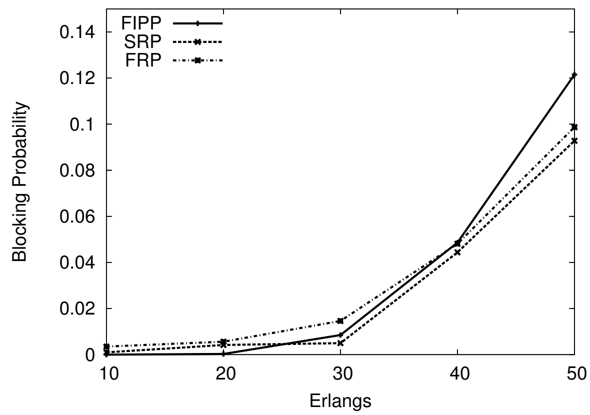


Figure 6: Comparison of blocking probability in NSFNET (W=16)

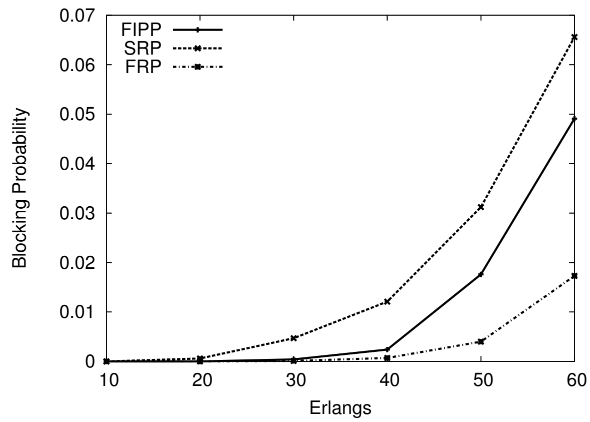


Figure 7: Comparison of blocking probability in USNET (W=16)

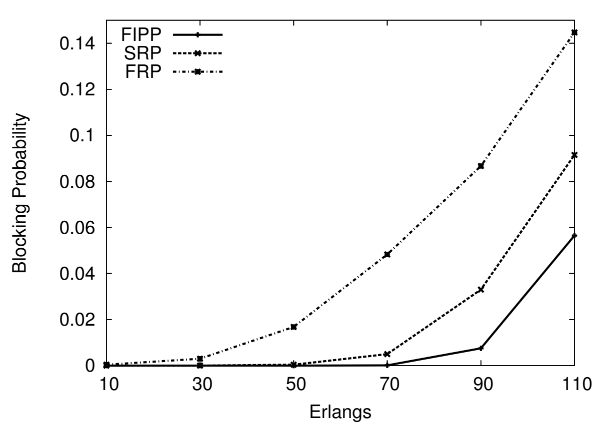


Figure 8: Comparison of blocking probability in COST239 (W=16)

200-300% less than FIPP. In COST239, however, FIPP outperforms both SRP and FIPP when the traffic load is relatively low due to the high density of the network. As the traffic load increases where the network is very saturated, FRP turns to perform better than SRP and FIPP, which is not shown in the results since the BP is over 10%.

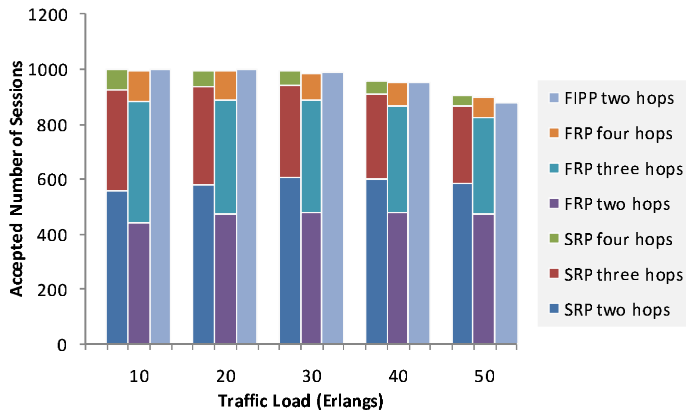


Figure 9: Comparison of Accepted Sessions in NSFNET

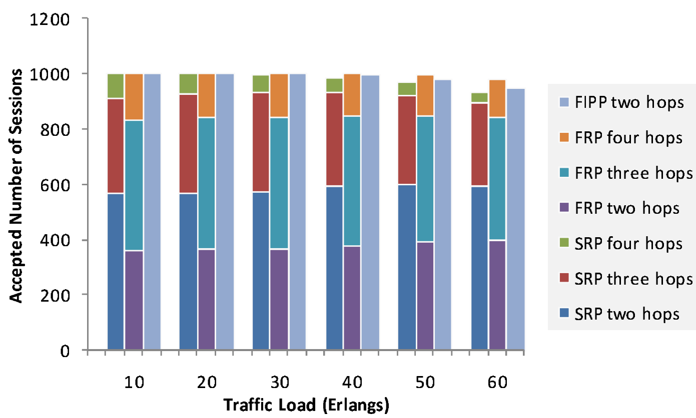


Figure 10: Comparison of Accepted Sessions in USNET

The reason that each scheme performs differently in a different network attributes to the actual topology, particularly average nodal degree. To have a better understanding of each scheme behind numbers, we also provide a detailed comparison of how each incoming session is protected by each scheme, which are shown in Figure 9, 10 and 11. Each figure corresponds to a network where the previous simulation was conducted. X-axis represents the traffic load in terms of Erlangs and Y-axis represents the average number of protected sessions among 100 independent simulations at each scenario. There are total 1000 incoming sessions. More sessions are protected, less sessions are blocked. In the figures, two, three and four hops denote whether a session is protected by a cycle only, a cycle with one PPL and a cycle with two PPLs, respectively. In a given scheme, the total number of accepted sessions is the sum of those three types. We compared the number of sessions of each type as well as the total number accepted sessions.

When the traffic is very low, there are only very small number of cycles in the network. Finding protection PPLs may not be easy in NSFNET due to the low network density (2.7). Thus, more than half of the accepted sessions are actually protected by the on-cycle links without extra PPL. Only less than 100 sessions are protected with two PPLs in SRP scheme. As traffic load increases, the total number sessions protected with two PPLs decreases, because we always prefer on-cycle protection than using PPLs if possible in our schemes. Where there are enough cycles existing in the network, there is a higher chance to find one to protect an incoming session without introducing extra PPLs. In USNET, the advantage of FRP over FIPP

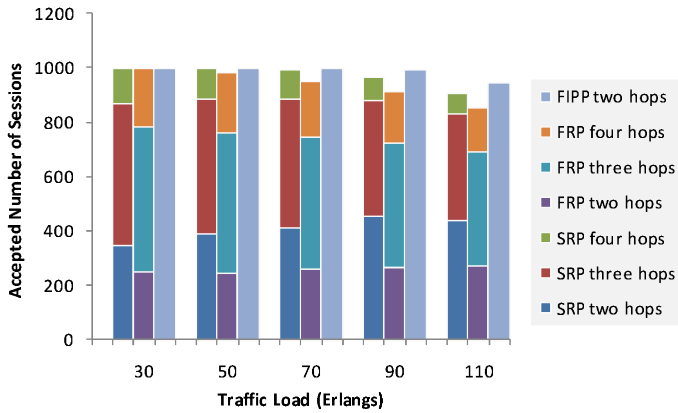


Figure 11: Comparison of Accepted Sessions in COST239

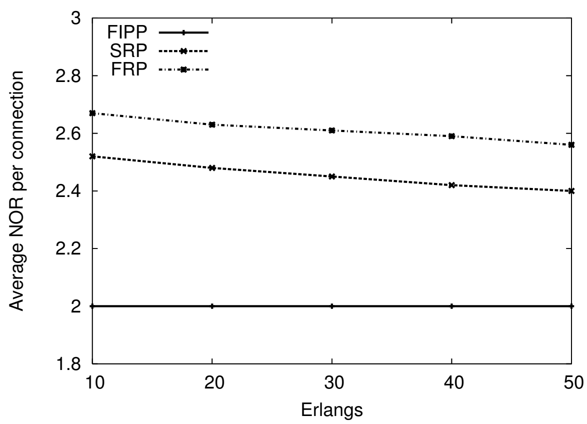


Figure 12: Comparison of NOR in NSFNET (W=16)

is more significant comparing to the advantage of SRP in NSFNET. In COST239, both SRP and FRP use more PPLs in the protection due to the high network density. The number of sessions with three or four hops are much higher in Figure 11 than that in Figure 9 and 10. It is relatively easy to find a PPL to protect a given session. However, it is also higher chance that the end nodes of an incoming session reside on a regular cycle for protection. Thus, FIPP actually has better performance over SRP and FRP when the traffic load is very low. The advantage of SRP and FRP do not show until network is very saturated.

Based on the results, SRP performs better than in sparse networks at a low level of traffic load. FRP achieves the best performance in larger networks, especially when the network is very saturated. One of the reason that SRP performs better in small and sparse networks, such as, NSF, is that to provision a session always using the shortest path will save some capacity for protection in a long run. Hence, more capacity can be used for protection such that more cycles can be established. However, in a network with high nodal degree, a cycle is more likely to reach a large group of nodes compared with a sparse network. In this case, FRP has a higher chance to protect a given session by using existing p^2-cycles when network load is very high and the network is over saturated.

We also studied the average NOR of each accepted connection as in dynamic traffic scenarios and the results are shown in Figures 12, 13 and 14. As expected, FIPP achieves the best solution with exact two node reconfigurations for each connection. Meanwhile, SRP also performs better than FRP in three networks. This reveals that connections protected by FRP use more PPLs than those used by SRP, which coincide with the facts reflected in Figure 9, 10 and 11. The reason is that FRP iterates every existing p-cycle in the network to protect each session before the primary path is established. It has a higher chance of ending up with a protection path with using PPLs instead of constructing a new cycle. It is worth noting that as

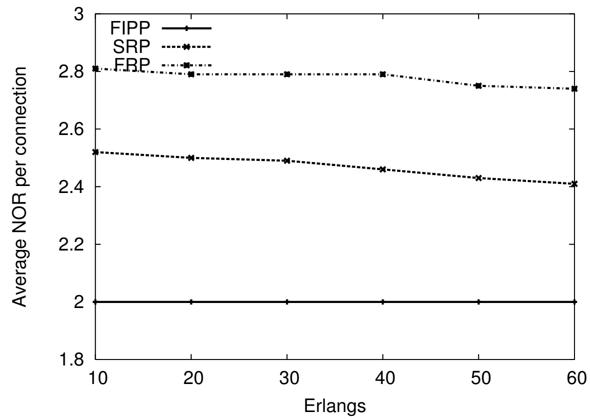


Figure 13: Comparison of NOR in USNET (W=16)

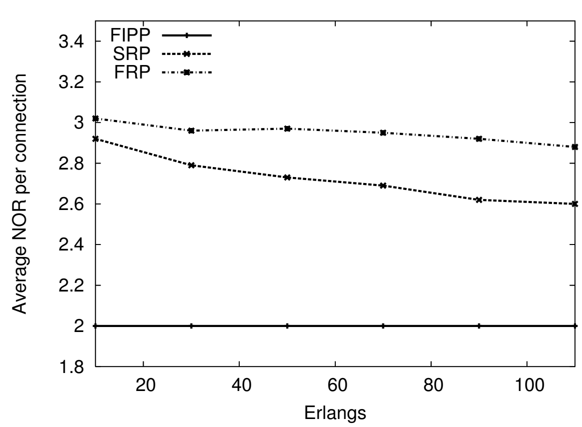


Figure 14: Comparison of NOR in COST239((W=16)

traffic load increases, the average NOR decreases and more sessions will be protected by only on-cycle links. Therefore, based on the simulation results, SRP, FRP and FIPP achieves lower BP over the others in different network scenarios given relatively low traffic loads. SRP has better failure recovery performance than FRP. However, SRP achieve lower BP when networks are sparse and FRP performs better when networks are large and saturated. Considering both static and dynamic traffic scenarios, the p^2-cycle protection scheme is a much faster protection scheme than SBPP and provides an enhancement of capacity efficiency over the FIPP p-cycle with a small increase in the recovery time.

VI Conclusions

In this paper, we proposed a new p-cycle based protection scheme in mesh network, named p^2-cycle, by augmenting the FIPP p-cycle with attached parasitic protection links (PPL) in order to enhance the protection ability by protecting the paths whose end nodes are not located on the cycle but only one hop away from the cycle. We studied both static and dynamic traffic scenarios. In static traffic case, we solved the joint capacity placement (JCP) problem optimally by formulating the problem as an ILP such that the total working and protection capacity used for all the sessions are minimized. In the dynamic traffic case, we proposed two heuristic algorithms, Strict Routing Protection (SRP) and Flexible Routing Protection (FRP), to handle dynamic traffic demands in order to minimize the total number of blocked sessions.

Based on the numerical results, the p^2-cycle protection scheme is a more capacity efficient than the FIPP p-cycle scheme in most traffic scenarios considered. In static scenario, it achieves a cost close to the optimal solution, achieved by SBPP, given a large number of session as a priori. Meanwhile, the p^2-cycle has much better recovery performance than SBPP in terms of NOR as traffic demands increase. In the dynamic traffic scenario, the p^2-cycle based approach, SRP and FRP achieve better blocking probability than FIPP does in NSFNET and USNET. Considering the trade-off between capacity efficiency and recovery speed, the p^2-cycle protection scheme is a more effective alternative of existent p-cycle-based and path-based protection schemes.

References

[1] D. Zhou and S. Subramanian, "Survivability in optical networks," *IEEE Networks,* 2000.

[2] P. Arijs, B. V. Caenegem, P. Demeester, and P. Lagasse, "Design of ring and mesh based WDM transport networks," *Optical Networks Magazine,* vol. 1, no. 3, pp. 27-41, 2000.

[3] S. Ramamurthy and B. Mukherjee, "Survivable WDM mesh networks. Part I-protection," *in Proceedings of IEEE INFOCOM,* vol. 2, pp. 744-751, 1999.

[4] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh network," *Journal of Lightwave Technology,* vol. 21, no. 4, pp. 870-883, 2003.

[5] P. H. Ho and H. T. Mouftah, "A framework of service guaranteed shared protection for optical networks," *IEEE Communications Magazine,* pp. 97-103, February 2002.

[6] Bharat T. Doshi, Subrahmanyam Dravida, P. Harshavardhana, Oded Hauser, and Yufei Wang, "Optical Network Design and Restoration," *Bell Labs Technical Journal,* JanuaryCMarch 1999

[7] Caihui Ou, J. Zhang, H. Zhang, L. H. Sahasrabuddhe and B. Mukherjee, "New and Improved Approaches for Shared-Path Protection in WDM Mesh Networks," *IEEE Journal of Lightwave Technology,* VOL. 22, NO. 5, MAY 2004.

[8] Dahai Xu, Y. Xiong and C. Qiao, "Novel algorithms for shared-segment protection," *IEEE Journal of Selected Areas on Communications,* v21. p1320-1331, 2003.

[9] Janos Tapolcai and et al. "A New Shared Segment Protection Method for Survivable Networks with Guaranteed Recovery Time," *IEEE Transactions on Reliability,* Vol. 57, pp. 272-282, 2008.

[10] W.D Grover and D. Stamatelakis, "Cycle-oriented distributed preconfiguration: Ring-like speed with mesh-like capacity for self-planning network restoration," in *Proc. IEEE ICC' 98,* 1998, pp. 537-543.

[11] D. Stamatelakis and W. D. Grover, "Thoeretical underpinnings for the efficiency of restorable networks using pre-configured cycles *p*-cycles," *IEEE Trans. Commu.,* vol. 48, no.8, pp.1262-1265, 2000.

[12] Wayne D. Grover, "Mesh-based Survivable Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking," *Prentice Hall PTR,* Upper Saddle River, New Jersey, 2004.

[13] D.A Schupke, C.G. Gruber, and A. Autenrieth, "Optimal configuration of *p*-cycles in WDM networks," in *Proc. of IEEE ICC,* 2002.

[14] Bin Wu, Kwan L. yeung, and Shizhong Xu, "ILP Formulations for *p*-Cycle Design without Candidate Cycle Enumeration," accepted in *IEEE/ACM Transactions on Networking,* 2008. (`http://www.eee.hku.hk/research/doc/tr/TR2008001_IFDCC.pdf`)

[15] W.D. Grover and J.E. Doucette, "Advances in optical network design with *p*-cycles: Joint optimization and preselection of candidate *p*-cycles," in *Proc. of IEE LEOS Summer Topical Meeting,* 2002.

[16] G. X. Shen and W. D. Grover, "Extending the *p*-cycle concept to path segment protection for span and node failure recovery," *IEEE Journal on Selected Areas in Communications,* vol. 21, no. 8, pp. 1306-1319, Oct. 2003.

[17] Adil Kodian and W. D. Grover, "Failure-Independent Path-Protecting *p*-Cycles: Efficient and Simple Fully Preconnected Optical-Path Protection," in *IEEE Journal of Lightwave Technology,* Vol. 23, pp. 3241-3259, 2005.

[18] D. A. Schupke, "An ILP for optimal *p*-cycle selection without cycle enumeration," *8th Conference on Optical Network Design and Modelling, ONDM,* 2004.

[19] Kamal, A. E., "1+N Network Protection for Mesh Networks: Network Coding-Based Protection using *p*-Cycles", *IEEE/ACM Transactions on Networking,* Vol. 18, No. 1, Feb. 2010, pp. 67-80.

[20] Lastine, D., Somani, A.K., "Supplementing Non-Simple *p*-Cycles with Preconfigured Lines", *Communications, IEEE International Conference on*, pp. 5443-5447, 2008.

[21] Bin WU, Kwan L. Yeung, Pin-Han Ho, "ILP formulations for non-simple *p*-cycle and p-trail design in WDM mesh networks", *Computer Networks*, Volume 54, Issue 5, 8 April 2010, Pages 716-725.

[22] T. Chow, F. Chudak, and A. Ffrench, "Fast optical layer mesh protection using pre-cross-connected trails," *IEEE/ACM Transactions on Networking*, vol. 12, no.3, pp. 539-548, 2004.

[23] Bin Wu, P. Ho, K. Yeung, J. Tapolcai and H. T. Mouftah, "CFP: Cooperative Fast Protection," *in the Proceedings of INFOCOM',* 2009.

[24] D. Stamatelakis and W. D. Grover, "Network restorability design using pre-configured trees, cycles, and mixture of pattern types," TRLabs Technical Report, Tech. Rep. TR-1999-05, 1999.

[25] Samir Sebbah and Brigitte Jaumard, "A Resilient Transparent Optical Network Design with a Pre-Configured Extended-Tree Scheme," *in the Proceedings of ICC',* 2009.

[26] Wensheng He and Arun K. Somani, "Comparison of Protection Mechanisms: Capacity Efficiency and Recovery Time," *in the Proceedings of IEEE ICC',* 2007.

[27] S. Thorpe, G. Edwards, and D. Stevenson, "Using Just-in-Time to Enable Optical Networking for Grids", *Proc. IEEE Broadnets Workshop*, Oct. 2004.

[28] D. A. Schupke, M. Jger, R. Hlsermann, "Comparison of resilience mechanisms for dynamic services in intelligent optical networks," *Proc. of DRCN workshop,* pp. 106-113, Oct. 2003.

[29] Wensheng He, Jing Fang, and Arun K. Somani, "A p-cycle based survivable design for dynamic traffic in WDM networks," *in Proceeding of IEEE Globecom,* pp. 1869-1873, 2005.

[30] Gangxiang Shen and Wayne D. Grover, "Design and performance of protected working capacity envelopes based on p-cycles for dynamic provisioning of survivable services," *Journal of Optical Networking,* Vol. 4, Issue 7, pp. 361-390, 2005.

[31] L. Ruan, F. Tang and C. Liu, "Dynamic establishment of restorable connections using p-cycle protection in WDM networks" *Journal of Optical Switching and Networking*, Volume 3 Issue 3-4, December, 2006, Pages 191-201.

[32] Samir Sebbah and Brigitte Jaumard, "Survivable WDM Networks Design with Non-Simple p-Cycle-Based PWCE," *in Global Telecommunications Conference, IEEE*, 2008, pp. 1-6.

[33] Bhandari R. "Optimal diverse routing in telecommunication fiber networks," *Proceedings of IEEE INFOCOM,* Toronto, Ontario, Canada, vol.3, June, 1994, pp. 1498-1508.