P Cycles: Design and Applications

-Aman Nijhawan

Abstract

Survivability is becoming an increasingly important goal in network design. An important aspect of network design today is designing networks with lowest cost which provide highest capacity efficiency and lowest downtime. Traditionally there are two basic architectures for designing networks. One is the mesh topology and other being the ring topology. Each of them has its own advantages and disadvantages. However recently a Hybrid of both these topologies having advantages of both ring and mesh topology called P- Cycles has gained ground. By the means of this report the author aims to survey basic techniques in the design of P-Cycle based networks and their applications to survivability.

1 Introduction

Survivability is a very important design goal in the design of networks today. With advances in optical network design it is now possible to pack more and more bandwidth of the order of terabits/s in optical networks. These networks are High BDP networks. Amount of traffic carried by a single fiber at any instant is very high and this traffic is expected to grow in the near future. A disconnection of only few milliseconds can cause a severe loss of data of the order of Gigabytes. Thus it becomes very important to protect these networks against failures. Network Failures arise from basically two conditions, node failures and link failures. Node failures are easily protected against by using node redundancy, however fibers are an expensive investment and it is desirable to minimize the total cost of design. Thus minimizing the spare is another important design goal taken into consideration when designing survivable optical networks. However often minimizing spare capacity leads to, an increase in the restoration time of the networks. This dichotomy is explained in the next subsection.

P-Cycles Preconfigured, Protection are a new technique that allows networks to be designed with minimum cost and maximum spare capacity efficiency. They can be used to protect both against node as well as link failures and can be used at physical layer WDM Networks as well as provide IP Layer Restoration. They provide quick restoration speed of rings and high capacity efficiency of mesh networks.

1.1 Ring Mesh Dichotomy [1]

Traditionally there have been two basic, approaches to survivable network design, namely ring and mesh.

Ring restoration uses the protection of the fiber running in the opposite direction to the working flow for protection. Ring restoration schemes can be designed in multiple ways. They can be Line Switched/Path Switched, Unidirectional/Bi Directional. The main quality of rings is that they can perform very fast switching of the order of 50ms. However they suffer from a main drawback of requiring at least 100% spare capacity. Protection paths are usually pre-connected and hence can provide fast restoration. Also they suffer from the limitation that the traffic can only be routed on the ring.

Mesh Restoration is more capacity efficient. Dynamic State Dependent routing mechanisms are used that make each unit of spare capacity reusable. Also in a mesh network the traffic can be routed independent of the topology of the
network using shortest path algorithms. However cost of this efficiency is increased delay. Restoration in a mesh network can take as much as 2 seconds to restore a span. Ring Mesh Dichotomy is illustrated in Table 1.1 [1].

<table>
<thead>
<tr>
<th><strong>Ring</strong></th>
<th><strong>Mesh</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>50 msec restoration times</td>
<td>Up to 1.5 sec restoration times</td>
</tr>
<tr>
<td>Complex network planning and growth</td>
<td>Simple, exact capacity planning solutions</td>
</tr>
<tr>
<td>High Redundancy even 200% is not uncommon</td>
<td>Well under 100% redundancy</td>
</tr>
<tr>
<td>Hard to accommodate multiple service classes</td>
<td>Easy / efficient to design for multiple service classes</td>
</tr>
<tr>
<td>Ring-constrained routing</td>
<td>Shortest-path routing</td>
</tr>
</tbody>
</table>

Table 1.1 The Ring Mesh Divide

Each of these mechanisms has its own tradeoffs. P-Cycles based networks allow us to utilize the advantages of both rings as well as mesh networks. They are fast, efficient and allow the traffic flow to be routed independent of the protection topology. They have been proven to be most efficient pre-connected protection mechanisms [2]. This paper is organized as follows. The next section gives an overview of the P-Cycle Concept and explains what exactly a p-cycle is. Section 3 surveys various algorithms for P-Cycle based Design. Section 4 surveys the applications of P-Cycles beyond Link protection. Section 5 gives the conclusions of the work.

2. **P-Cycles**

P-Cycles are protection cycles for mesh networks that are created out of spare capacity. The main attributes of a p-cycle are:

- Preconfigured in a network like a ring
- Protect both On Cycle and Straddling Links
- Allow working capacity to be routed using shortest path routing schemes
- Efficiency similar to that of mesh and restoration speed similar to that of link.
- Easy to Configure
- Bridge the divide between the ring and mesh.
2.1 How are P-Cycles Formed [2]?

Fig 2.1 illustrates the formation of a P-Cycle in a network.

In a network with nodes and links there is a capacity $C_{ij}$ associated with each link $ij$. Also Each Link Carries a traffic $W_{ij}$ which is the working capacity of the link. This working capacity is always less than or equal to the capacity of the link $W_{ij} \leq C_{ij}$. Spare Capacity $S_{ij}$ of a link is the difference between the total and the working capacity of the link. P Cycles are made out of this spare capacity. These P-cycles are used to protect the working capacity $W_{ij}$ in case a link $ij$ fails. P -Cycles protect both on cycle and straddling links.

2.2. Working of P-Cycle [2]

Working of P-Cycle is simple to explain. Figures 2.2 (a), (b), illustrate the working of a P Cycle. Fig 2.2(a) shows a case in which an on cycle link breaks the cycle provides one restoration path this is similar to how a ring would provide restoration. In Fig 2.2(b) a straddling link breaks. In this case the cycle provides two restoration paths. Any of which may be used. To increase efficiency of the cycle both paths can be used simultaneously if the straddling link carries twice the traffic of the on cycle links. This way the entire traffic on the straddling span can be protected by two different parts of the same p-cycle hence increasing the capacity efficiency of the P-Cycle.

Figure 2.2(a) Protection against and on cycle Failure (b) Protection Against a straddling link failure adapted from [5]
The secret to the efficiency of a p cycle is straddling link protection which is effectively obtained free of cost and the speed of restoration is inherited from their ring like structure.

### 2.3 P-Cycles based Design: Algorithms [6]

P-Cycles based design can be done in a variety of ways. The selection of an algorithm depends on the topology of a network and the type of failure to protect against. Most P-Cycle Design methods are ILP based. ILPs are used to obtain a set of optimal p-cycles by maximizing or minimizing an objective function subject to a set of constraints. Two most common and simple objective functions are minimizing the spare capacity used for 100% restorability and maximum restorability for a given spare capacity. These methods are explained further. More advanced algorithms utilize heuristic efficiency metrics used to limit the no of cycles to be enumerated. There are some algorithms such as [3] that do not use ILP based methods for P-Cycle selection.

#### 2.3.1 ILP Based Methods [6]

**Some Parameters**

- \( x_{ij} \): No of Useful paths a P Cycle J provides in case span I fails
- \( S \): a set of Network Spans
- \( P \): a set of Elementary Cycles in a graph
- \( \partial_{ij} \): 1 Cycle i includes span j
- \( W_j, S_j \): No of working channels and spare channels on span j
- \( R_j \): the number of available protection paths in the design in excess of those required for span j (defined as a slack variable)
- \( U_i \): No of un-restorable working channels on span i
- \( N_j \): the number of unit capacity copies of a cycle j in the design

**Maximum P-Cycle Restorability with Given Spare Capacity**

Minimize the total number of un-restorable spans \( u_i \) in the network.

\[
\text{Minimize } \sum_{i \in S} u_i 
\]

**Constraints**

- Spare Capacity on each span can support the p-cycles that cross it
  \[
  s_k \geq \sum_{j \in P} \partial_{k,j} \cdot n_j \quad \text{for all } k \text{ that belong to set of Span } S
  \]
- Every Span accesses enough protection coverage for all its working capacity
  \[
  u_i + \sum_{j \in P} x_{ij} \cdot n_j = w_i + r_i \quad \text{for all } i \text{ that belong to set of Span } S
  \]
- Un-restorable portion of any span cannot be more than its working capacity
  \[
  0 \leq u_i \leq w_i \quad \text{for all } i \text{ that belong to set of span } S
  \]
- Numbers of each p-cycle copy and slack variable are non negative.
  \[
  n_j \geq 0; r_j \geq 0 \quad \text{for all } i \& j \text{ that belong to a set of span } S
  \]
Minimum Spare Capacity for 100% Restorability

Minimize

\[ \sum_{k \in S} c_k \cdot s_k \]

Constraints

- Spare Capacity on each span can support the p-cycles that cross it
  \[ s_k \geq \sum_{j \in P} \delta_{k,j} \cdot n_j \text{ for all } k \text{ that belong to set of Span } S \]
- The Number of useful paths provided for each span supports 100% restorability.
  \[ w \leq \sum_{i \in P} x_{ij} \cdot n_j \text{ for all } j \text{ that belong to a set of span } S \]
  \[ n_j \geq 0 \text{ for all } j \in P \text{ and } s_k \geq 0 \text{ for all } k \in S \]

2.4 Efficiency of a P-Cycle

There are three important metrics for describing the efficiency of a P-Cycle namely Topological Score TS [4], Apriori Efficiency AE [4] & Demand Weighted Efficiency [6]. These metrics help us gauge the efficiency of using a particular P-Cycle in a design.

The First basic Metric is the Topological Score of a candidate p-cycle (j) defined as

\[ TS(j) = \sum_{i \in S} x_{ij} \]

where \( x_{ij} \) as already defined can be either 0 or 1 or 2 depending upon if the link is on cycle, straddling or not a part of the cycle.

The Topological Score is basically a measure of the total number of protection relationships a cycle provides to the network. However it fails to take into account the cost of constructing that cycle in terms of the cost of using the spare capacity on that link.

The Second Metric is the Apriori Efficiency it is an improvement over the basic TS metric. It gives the potential efficiency of constructing a cycle in terms of the protection relationship as well as the cost of constructing the cycle.

AE is defined as
AE (j) = \sum_{i \in S} x_{ij} / \sum_{k \in S} \partial_{kj} \cdot C_k

\text{i and k } \in \text{S which is a set of all links}

\text{Ck is the cost of using the link k}

\partial_{kj} = 1 \text{ or } 0 \text{ depending on the fact if the link k is part of the cycle j or not.}

It can be observed that the more number of straddling links a cycle has, the higher its AE metric will be because \(x_{ij}\) for a straddling link is maximum (2) and its contribution to the total cost of using the cycle is 0 hence. AE is a good metric for judging the actual efficiency of a cycle. However AE and TS both only give a potential efficiency of a cycle. Actual efficiency depends on the actual set of demands of the network.

The third metric Demand Weighted Efficiency DWE takes this into account and gives the actual demand set on each links and calculates the protection provided by the cycle to the network in presence of the demands. For example a cycle may protect n straddling links in a network, but if there is no actual traffic flowing through those n links, the actual protection provided by the cycle to the network should not factor in those n links.

\[
\text{DWE (j)} = \sum_{i \in S} \min (x_{ij}, w_i) / \sum_{k \in S} \partial_{kj} \cdot C_k
\]

\text{i and k } \in \text{S which is a set of all links}

\text{Ck is the cost of using the link k}

\text{w_i is the working capacity of the link}

Fig 2.3 Calculation of AE and DWE metrics for an example network

Based on these efficiency metrics more efficient algorithms for p-cycle selection have been proposed to reduce the inherent hardness in computation of the above explained ILP methods in [ER] these algorithms are explained in the subsequent sections.
2.5 Score based Design Assembly Algorithm [6]

This algorithm is based on two very simple observations.

- Calculating all P-Cycles with a hop or a circumference limit is a hard operation however for a given network it is done only once initially.
- Evaluation and Re Evaluation of DWE for a cycle can be a simple \(O(n)\) operation in a suitable data structure Fig 2.4.

<table>
<thead>
<tr>
<th>P Cycle Number (p)</th>
<th>List of Uncovered On cycle spans with their uncovered (w_i) ((i,w)\rightarrow(i,w))</th>
<th>List of Uncovered Straddling Spans with uncovered capacity ((i,w)\rightarrow(i,w))</th>
<th>Current Demand Weighted Score</th>
</tr>
</thead>
</table>

Fig 2.4 Data Structure used for Design Assembly Algorithm [6]

The actual algorithm works as follows:

1. Enumerate all cycles up to a circumference limit and build the data structure

2. If set of all P-Cycles \(P\) is too large we can throw all cycles with an AE score of \((1 + K)\) (\(K\) fixed constant) to reduce the set.

3. For every current entry in the data structure

4. Compute DWE using the above described method

5. If DWE < \((1+ K)\) remove the cycle and delete its entries in the table.

6. Scan the remaining list with the \(p\) –cycle with the highest score

7. Add this \(p\) –cycle to the design

8. \(k=p\)

9. For all \(i\) on cycle list \(k\) \(w_i\leftarrow w_i - 1\)

10. For all \(I\) on straddling list of \(k\) \(w_i\leftarrow w_i - 2\)

11. Until all \(w_i = 0\)
2.6 Pre-selection of Candidate p-Cycles [6]

The Score based design assembly algorithm is greedy in nature i.e. it chooses the best looking p-cycle at each stage and adds it to the p-cycle set. However it fails to analyze the efficiency of the collective set. A basic algorithm proposed by Grover et al, forming the basis for many heuristic based p-cycle selection algorithm pre selects a set of p-cycles based on a metric and gives this selected set as an input to a MIP solver to calculate the protection capacity provided by this set. This can be done as follows.

For a given large network,

- Select top 5000 AE ranked Cycles
- Add top 2000 cycles ranked in terms of maximum straddling links.
- Add 2000 Longest cycles
- Add 1000 Random cycles.

Remove duplicate cycles among the above set and present this elite cycle set to a MIP solver like CPLEX and solve calculate the protection provided by this set of cycles.

This may not generate 100 % protection with a sure shot guarantee; however this algorithm is expected to work well as long as the underlying methods select a good set of cycles.

2.6 Non ILP Based Algorithms

2.6.1 Straddling Span Algorithm [3]

In contrast with the heuristic based approaches the SSA avoids the computationally hard step of cycle enumeration at each step.

SSA is based on the observation that a p-cycle can be seen as a combination of two node disjoint paths between a straddling span. A basic condition for this to happen is that both the end nodes have a degree of 3 at least. The algorithm is a very simple and generates a set of p-cycles which are guaranteed to have at least 1 straddling span. Also the number of cycles found is limited to the number of links in the graph at the maximum. The algorithm can be explained by the following pseudo code and example.

```plaintext
For each span s {
    Find two node disjoint paths between the end nodes, if such paths exist set ndpaths=2 if only one node disjoint path exists set ndpaths=1.
    
    Case {
        If ndpaths=2 and the degree of end nodes >=3
            Form a p-cycle with both the node disjoint paths found in step 1 that exclude the span.
        If ndpaths=1 and the degree of end nodes>2
            Form a p-cycle with the node disjoint path and the initial span
        If degree of end nodes <2
            No action, rely on coverage being provide to super-spans
    }
}
```

2.6.2 SP Add and Expand and Grow Algorithms [7]

The methodology from the above explained is combined with add and join operations of a p-cycle to yield a high efficiency p-cycles.

The SP add algorithm [7] works in a simple manner. It takes each span on a cycle and constructs a new cycle is by removing the span and replacing it with the shortest path between its end nodes that is node disjoint from the original cycle, if such a path exists. This makes the initial span a straddling span and the resulting cycle has a higher efficiency than the original cycle.

The Expand Algorithm [7] is similar to SP-Add except that we do not stop at the first SP-Add operation. Instead, the search is continued, moving on through the spans on the initial cycle, seeking yet another span that can be transformed into a straddle in the same manner. This process is continued until all cycles are visited on the initial cycle

Each time the cycle is expanded the cycle, it is ensured that it is not only disjoint from the original cycle, but also to every previous routes that have been added.

The Grow Algorithm [7] algorithm is more than both the above algorithms. In this algorithm we start with an initial set of p-cycles, A SP –Add operation is performed on each cycle to give an intermediate set $B$. Then, for each cycle in $B$, we visit the first span in the cycle and find the shortest route that connects its end nodes but is also node-disjoint from the cycle itself. Until such a route is found we keep iterating through the spans of the cycle. If such a route is found, the span is removed and the route is added in its place, to create a new cycle. The operation does not stop here. The operation is restarted with the first span of this new cycle, continuing as above. The process continues until no further route additions can be made.

The basic idea behind this algorithm is to come up wide range of cycles, including the very small, large and medium length; on each replacement a span becomes a straddle thus generating a new cycle with higher AE efficiency.

3. Applications of P –Cycles

We have seen how p-cycles can be used to protect against link failures. This section takes a look at applications of P-cycles beyond protection against single link Failure. The end of this section discusses some recent extensions.

3.1 Multiple Link Failure Protection [8]

As explained in the earlier sections single link failure is easy to protect using p-cycles. However with the increasing size of networks today, simultaneous double failures are not uncommon. Designing P-cycle based networks to protect against such failures is gaining importance [8]. The algorithm explained [8] is simple and based on a series of table lookups. It is summarized here as follows:

1. A network maintains a Link –Cycle table which has protection path of every link in the network. This is called as protection topology of the link.
2. For every s-d pair in the network we calculate additional shortest paths, these are joined and stored in the Link –Cycle table as additional p-cycles.
3. The additional P-cycles are ranked based on their topological score and a P-cycle pool table is established and each node maintains its adjacent nodes PPT.
4. When a link fails highest ranked cycle from a PPT is chosen that would service that link.
5. If now a second link should fail, The PPT is looked up again for a p-cycle that would service both the failed links if such a Cycle exists, it is chosen and the earlier cycle is returned to the cycle table.
The Algorithm was simulated on the EON Network and NFS Network to find the probability of surviving a double link failure as having 3 or more simultaneous failures is a very rare occurrence. The algorithm proved to be efficient and provided promising results. Also another key point here is that protection against multiple failures is provided without adding extra capacity to the network so basically for the same cost we are getting more protection. The algorithm can be run both distributed online manner as well as a centralized offline computation. An important observation by the authors of [8] was networks with higher degree of connectivity tend to survive multiple link failures better than those with lower degree.

3.2 SRLG Failure Protection [9]

A SRLG is a set of links that share a common resource, whose failure will cause a failure of all the links in the set. Upon SRLG Failure; all links in the SRLG are gone. Thus every failed link must be protected by some p-cycle. Also if multiple failed links happen to be on the same cycle then that p-cycle is broken hence cannot provide any protection.

A simple algorithm given in [9] is used to predict the protection provided by a p-cycle to a set of SRLG links is summarized here.

```
1. int CYCLE_LINK_SRLG(cycle i, link j, SRLG k)
2. if  j does not belong to k
   return 0    //As j is not a part of SRLG that failed.
3. If j’s end nodes are not both on the cycle
   return0   . //Link j cannot be protected by cycle
4. Remove the links in SRLG k from the Cycle I
5. If cycle i remains a cycle
   return 2
6. else if link j has both end nodes on the same segment of the cycle
   return 1
7. else
   return 0
```

The value of protection obtained for a SRLG by each cycle is put into a score metric and solved for an optimal set of p-cycles that give maximum protection against all SRLG failures of a network.

3.3 P-Cycles in IP Layer Protection [10]

IP Layer protocols such as OSPF have some inbuilt fault tolerance but usually restoration at IP Layer is time consuming. This can lead to disconnection of sessions and the fault may be noticed at higher layers also. Authors in [10] have provided methods to minimize this restoration time using Virtual P-Cycles. The process explained in [10] is summarized here. There are 3 basic steps.

- Encapsulation.
- Deflection.
- Re introduction

Some IP addresses are explicitly reserved to form virtual circuit p-cycles covering the entire network with these cycles, these virtual circuits are embedded into the routing tables. If a packet arrives at a router and the destination address in the packet pointing towards a router that is known to have failed. The packet is **encapsulated** with a VC number and routed along the virtual p-cycle path. At each next router the payload of the VC packet is examined. If the cost of reaching the destination indicated in the payload is cheaper through a normal path, it is routed stripped of the VC packet and **reintroduced** to be routed using IP. However if cost of reaching the destination is cheaper through the Virtual P-cycle path, it is unchanged and forwarded along the VC path, this is known as **deflection**.
3.3.1 Node Failures

Node failures in a network are usually protected against by using node redundancy thus there is not much research in the area of node protection using P-Cycles. However the most common failure in today’s IP networks is Router Failure. This is a kind of node failure so it is important to look at this aspect of P-Cycles.

We can protect against node failures by encircling all the neighbors of the node to be protected by a node encircling p-cycle.

In case the node fails all connections passing through that node effectively become straddling spans of the cycle hence are protected. However it may not be possible to use simple cycles for node encircling in such a case, assuming the graph is bi connected we can use a non simple cycle to do node encircling.

The following diagrams adapted from [11] explain the working of node encircling P-Cycles.

3.4 Failure Independent Path Protection using P-Cycles

Full pre-connection in p –cycles is the source of their ring like fast speed. This protection strategy has compelling advantages over restoration strategies that employ cross connection on demand when a failure occurs. Additionally in long haul optical networks which have fiber segment impairments, alignment of link segments is largely a manual process[12].Thus in case of a fiber failure in such a network, this process cannot be done on the fly. This makes pre-connection a more suitable choice for optical networks.

FIPP extends the P-cycle concept of pre connected paths but still provides E2E independence against both link and node failures. A commonly used method to provide protection in optical networks is shared backup path protection SBPP. There are however some issues with SBPP, for example when a fault occurs there is signaling time involved to make the required cross connections. Also when the fault occurs there is no way of being sure if the backup path has enough ‘transmission integrity [12]’. FIPP is an improvement over this.
In FIPP all failover paths are predefined hence the integrity of the backup paths is known in advance. Also there is very fast signaling since all backup paths are active at all times and it takes only the end nodes to detect that there has been a fault somewhere in the flow to initiate a switchover. The authors claim that in capacity efficiency FIPP matches SBPP and like SBPP is failure independent.

FIPP simply trades a little capacity efficiency in exchange for increase in restoration speed. The basic principle behind FIPP is that we allow a p-cycle in a network to only protect a group of flows that are vertex disjoint in nature. The authors assume a single failure scenario but also state that FIPP can be used to provide SRLG protection by using a more stringent condition of SRLG disjoint routing. An example similar to that was discussed in [12] is presented here.

![Diagram of network with p-cycles](image)

**Fig 3.3 Failure Independent Path Protection**

Let us assume that there are two p-cycles here in this network one is AGFEDCBA and another is AFDCA.

FIPP is simply states that cycle AGFEDCBA can protect all flows on paths AGF, BHIE, CD since they are mutually vertex disjoint. However if later there is a flow A-I-D, it cannot be protected by the same cycle in spite of the fact that it is straddling on it, since it is not vertex disjoint with the above flows so to protect A-I-D we need another p-cycle which would be AFDCA. In this case assuming single link failure, it is clear that all flows are protected and the failure path is known in advance hence can be tested.

4. Conclusions

In this research report we have presented concepts from the area of P-cycles in network survivability. P-cycles are an important area in network survivability research this report is presents main ideas and sub domains within this research area.

The techniques discussed to select an optimal set of p-cycle to provide protection in optical network ongoing research topics. Various extensions to the p-cycle concept have been proposed. This report has summarized some of those extensions. This report is intended to be a very basic tutorial in the area of p-cycles.
Bibliography


