An examination of single link failure in an optical network with full grooming

by

David Lastine

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Engineering

Program of Study Committee:
Arun K. Somani, Major Professor
Akhilesh Tyagi
David Fernandex-Baca

Iowa State University

Ames, Iowa

2007

UMI Number: 1443161

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This thesis considers a survivable optical network with link-based protection that can survive one fault, has links each with a fixed total bidirectional capacity, and full grooming at each node. An upper bound on the total primary capacity the network can provide is derived from conditions that are necessary but not sufficient to guarantee restoration. An ILP formulation is developed and presented that achieves this bound when possible for a given topology, and tends to spread out backup capacity by insisting on using small cycles to form backup routes. The blocking probability of connections in a network thus protected is compared against the networks blocking probability if protected by a p-Cycle. Simulations showed that the more spread out backup capacity had lower blocking probability than a Hamiltonian p-Cycle. P-Cycles are supplemented by an additional pre-configured structure. The capacity used by non-simple p-cycles and lines are compared against non-pre-configured resource reservation. Examples where lines can be useful are easy to construct. However ILP solutions for protecting the Cost 239 network did not include them.

CHAPTER 1. INTRODUCTION

The need to send information long distances quickly is doubtless as old as civilization itself. Four thousand years ago or more [1, 2] the Egyptians had a postal service. Such systems have always had outside adversaries threatening to cause them to fail. In the fifth century B.C. the Greek Herodotus commented on this when he said the mail carriers of his day in the Persian system were not stopped by snow, rain, heat, or darkness. In fact, despite these problems they accomplished their task with "all speed."

Of course, it isn't always possible to protect a message carrier from all possible dangers. When information is important it is worth using multiple redundant messengers to get the information to its destination. This strategy was used during the American Revolution by Dr. Joseph Warren, who sent a message via two men traveling separate routes at separate times [3].

Today computer networks have taken over the role that message carriers or mailmen have traditionally played. Network routers and computers have taken over the jobs of post office and mailbox. Fiber optics, wire, coaxial cable and in some places wireless links have taken the place of message carriers. Fiber optics based networks are a carrier of choice for much of the traffic going between cities. Like the message carriers of old, snow or rain could be a problem, but now so can accidental cuts during digging for construction, losing a support pole from a vehicle hit, being chewed on by rodents, or by being hit by a tree.[4] Also like messengers of old, redundancy is useful in ensuring that information gets to its destination. This is the topic of this thesis.

Specifically this thesis explores the case of a network protected by a link-based protection scheme that has one fiber optic cable fail. The amount of redundancy required by such a system is explored. The effects of how the redundancy is distributed are explored. The effects of pre-configuring on available primary capacity are explored.

CHAPTER 2. BACKGROUND AND LITURATURE REVIEW

2.1 Nuts and Bolts of Optical Networking

An optical network is a computer network where information is transmitted between computers via light traveling along a cable. They replaced networks using copper wire because of their lower bit error rate, 10^{-9} vs. 10^{-5} for copper [5], and because of the high bandwidth of optic fibers.

Physically the bandwidth on a fiber can be split up a couple of ways. One way is to take advantage of the fact that light is an electromagnetic wave. As such it is possible for light of different frequencies to be sent down the same fiber at the same time and later be separated by a receiver. The technique of using several frequencies of light to transmit information on a fiber optical cable at the same time is called wavelength-division multiplexing (WDM).

A second way bandwidth can be split on a fiber is by the different users of a fiber taking turns transmitting. This technique is known as time-division multiplexing (TDM). Of course TDM and WDM can be done at the same time, in which case it's the users of a frequency that take turns using the frequency.

A number of times it is the case that computers are connected by more than one fiber. Thus bandwidth between computers in a network can be divided by fiber, wavelength, and time slot.

Often a computer in a network may need to communicate with a computer not directly connected to it. The intermediate computers can forward the information in one of two ways. One way is to convert the information from the optical domain back into electrical signals, process it, and then resend the information as light on another link. This is known as Optical Electronic Optical (OEO) conversion. In this approach electronic processing of the information is the bottleneck.

The second way to forward the information is to redirect the light out a different fiber than it entered in. This is accomplished by an Optical Cross Connect (OCX) that uses tiny mirrors to bounce the light from one fiber to another. This leads to the concept of a circuit switched network. When two computer systems want to communicate, the OCXs between them can create a light path between the two computers. This allows an optical network to be protocol independent since only the two communicating systems need understand the protocol they are using.

While circuit switched networks have great speed advantages they suffer from blocking. Blocking is when two computers request a given amount of bandwidth and the connection is not established since the resources are not available. Resources already dedicated to maintaining other circuits are considered to not be available. Also, resources reserved as backup are not considered available.

Blocking is affected by the grooming capabilities of various nodes. Grooming refers to the ability of a network node to switch traffic passing through it from one wavelength to another or one time slot to another, or if multiple fibers go between two computers, to select which fiber a signal travels on. In a network it may be the case that some nodes can only switch wavelengths, some time slots, others fibers. In this thesis, it is assumed all nodes have full grooming; so grooming can be done for fibers, wavelengths or timeslots by all nodes.

2.2 Overview of Fault Tolerant Strategies

The survivability techniques fall into one of two categories [6, 7]: protection and restoration. Restoration is conceptually a very simple idea, when a link fails, the networks centralized control attempts to find new routes for all the connections that have been disrupted. This scheme has the advantage of requiring no computation time or any other network resource when there is no fault in the network. It has the downside that recovery from failure is slow, computationally expensive, and there is no guarantee that interrupted connections can be restored.

Protection is a more proactive approach where some of the networks resources, such as link bandwidth, are reserved for use in the event of a failure. This strategy has the advantage that restoration of a connection can be guaranteed. The approach has the drawback that the probability that a connection request gets blocked is increased since there are fewer resources available to form connections. Many of the specific strategies in this approach have turned out to be NP-hard.

Protection approaches fall into four categories: global rerouting, path-based, link-based, and segment based.

Global Re-routing: In global re-routing some or all of the connections are re-routed after a link failure has occurred. This is the best any strategy can do. This is the only strategy that does not explicitly reserve network resources. An example of this strategy is a sub-graph-based routing [8], where a connection request is accepted only if the connection can routed on all possible graphs of the network that result due to any of the possible failures.

Path-Based Approach: When the path a connection is using fails, a new path is provided for the connection without effecting un-failed connections. The original route is called the primary path and the one used after the failure is called the backup path. How the path based approach plans a new route for a connection has many varieties. The backup route may share no resources with the initial route (dedicated backup), it could have minimal overlap with the primary (primary-backup multiplexing), or several backup routes could have resources reserved with one being chosen based on which network component failed (shared backup). Backup paths can be link disjoint, or link and node disjoint. A resource can be reserved by multiple backup paths if the network is only designed to survive one link failure and the primary routes are disjoint.

Normally path based strategies are required not to re-route connections that did not use the failed resource. However some authors don't include this in the definition and may classify global rerouting as a special case of path based protection. For the rest of this paper, it will be implicit that protection strategies will not reroute connections that do not use failed components.

Figure 1(a) shows a path-based approach in action. The dotted line in the figure is a primary path that failed when the link between nodes 2 and 3 failed. The solid line shows the backup path that protected the primary path.

Link-Based Approach: In the link-based approach, the connections are rerouted around the failed link. An example of this is shown in Figure 1(b). The dotted and solid lines have the same meanings as in part (a) of the figure.

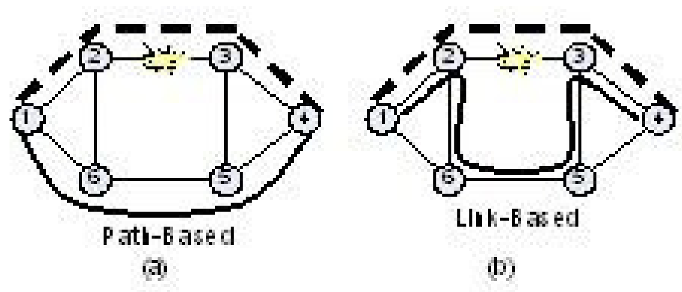

Figure 2.1 Example Path and Link Based Protection

Segment-Based Approach: Segment based protection is protecting a path where two links are routed around if a link in a path fails. It is mentioned only for the sake of thoroughness and will not be discoursed further. The techniques used are similar to the path based methods. However, it is claimed that a better efficiency can be achieved due to reuse of resources.

Speed Considerations:

The speed of restoration of connections is important to minimize the impact of a fault. For historical but technically questioned reasons the target time for restoration is 50 ms [4]. To reach this speed, resources reserved for backup can be pre-configured. This consists of creating circuits in the network to be used to carry connections before a failure occurs. When a link failure occurs, nodes can break into this circuit and send the traffic that used the failed link. The major bottlenecks of contacting intermediate nodes to route the connections through and verifying continuity of connections at intermediate nodes are thus eliminated.

In the late 90's arbitrary pre-configured patters where explored via genetic algorithms and by use of ILP[19]. The ILPs involved did not calculate any pre-configured structures but merely selected from a set of candidates they were provided. This work was done as a network design problem with the assumption that backup capacity could be added as required. The general conclusion of this work is that cycles were a very good pre-configured structure to use. Therefore work in preconfigured structures then narrowed to considering only how to select and place the best simple cycles.

Recently the variety of pre-configured structures being considered has begun to increase again. The motivation is a shift in what is being assumed about the underlying network. When a network has constraints on how much backup capacity can be added as in [10] or has a dynamically configured fixed total capacity as in this paper, then there are cases where simple cycles can not protect a network but other preconfigured structures can. The major pre-configured structures previously considered by researches are described below.

A p-Cycle is a preconfigured structure in a network where every functioning link used to create the structure is used to protect any link protected by the p-Cycle. The initial proposed structure of a p-Cycle is a simple cycle [9]. That is a cycle that does not revisit a node. It is allowed for links not part of the cycle to begin and end on nodes in the cycle. These links are termed straddling links. Links on the cycle are termed on-cycle links. A p-Cycle protects all of its on-cycle links and the straddling links that start and stop on its nodes.

p-Cycles are a link based protection mechanism since recovering a connection consists of routing connections around a failed link while leaving the rest of the path intact. As shown in Figure 2 (a) p-Cycles provide protection for on cycle links by re-routing traffic all the way around the cycle. Figure 2 (b) shows that straddling links are protected by splitting its traffic and using the p-cycle to provide two paths.

Finding the best way to protect a network with p-Cycles is a very complex problem. There are many known ways to go about selecting p-Cycles to use. All possible cycles can be enumerated and an ILP selects the best ones [9, 14]. The number of cycles in a network can grow exponentially with the number of nodes, so heuristics have been explored to select a "high merit" set of candidates [4] and only these cycles are given to an ILP. Alternately an ILP

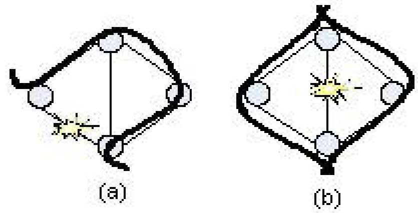

Figure 2.2 p-Cycles

can compute the p-Cycles directly from the connectivity of the network [14] or from a set of fundamental cycles in the network [11]. Both joint routing of connections and p-Cycles have been considered as well as non-joint [9-13].

The non-simple p-Cycle is a variation on p-Cycles where a p-Cycle is allowed to use nodes and or links more than once. Non-simple p-Cycles where the maximum allowed number of reuse of nodes or links is varied are studied in [10].

An advantage of non-simple p-Cycles is their ability to use links as straddling links that simple p-Cycles can not. For an example consider Figure 3. It shows a non-simple p-Cycle listing vertices starting and ending at 1, the cycle is 1-4-3-2-4-5-1. Link 1-2 is a straddling link. No simple p-Cycle could have made 1-2 a straddling link, since its removal would result in the network graph being disconnect-able by removing of only one node.

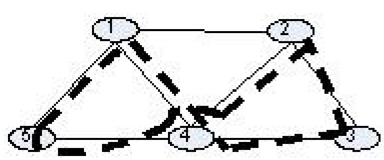

Figure 2.3 Network Protected by Non-Simple p-Cycle

Recently p-Paths have been proposed [16]. The structure of a p-path is that of a p-Cycle with one link removed. It can protect the removed link and any straddling links, but can only do so with one restoration path not two as a p-Cycle can. Also on-line links can not be protected. Only the method of placing p-lines using an ILP was been considered so far in the literature.

CHAPTER 3. LIMITS ON PRIMARY CAPACITY

This section establishes upper bounds on maximum primary capacity any network can service for the fault scenario described in each subsection. It does so to provide a standard against which the capacity usage of an algorithm can be measured. All such bounds establish conditions that are necessary, but not sufficient to guarantee protection in any given network. In other words, the topology of a network may prevent these upper bounds from being achievable. Examples to illustrate this point will be presented in section 3.2

3.1 Link-based Protection: All Traffic Recoverable

We first define a link to be *incident* on a node, if the node is one of the two end nodes of the link

Lemma: Links incident on a node allocate at least C units to backup between them.

On any given node, n, with L_n incident links, the primary capacity on a given link i must be equal to or less than the sum of the backups on the other links. Otherwise there is some link that if it fails, there is not enough backup capacity in the other links to cover it.

Algebraically these requirements can be stated as:

$$\forall i \in \{1, \ldots L\}$$

$$P_{i,n} + B_{i,n} \leq C \tag{3.1}$$

$$P_{i,n} \leq \sum_{i \neq j} B_{j,n} \tag{3.2}$$

Figure 3.1 shows the scenario where link i has failed and links $1 \ldots, i-1, i+1, \ldots L_n$ must use their backup capacity to pick up the primary traffic that was on link i.

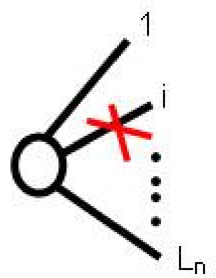

Figure 3.1 Link i fails and its traffic uses other links

The fact that a node reserves at minimum C units of capacity can be seen by turning the constraints into a linear programming problem by adding the objective function.

$$Maximize \sum_{i=1}^{L_n} P_{i,n} \tag{3.3}$$

Since decreasing the value of any given B_i increases the corresponding P_i, the objective function is maximized when the second constraint becomes the strict equality of

$$P_{i,n} = \sum_{i \neq j} B_{j,n} \tag{3.4}$$

From this it follows that the maximized objective function is

$$P_{i,n} = \sum_{i=1}^{L_n} \sum_{i \neq j} B_{i,n} \tag{3.5}$$

$$= (L_n - 1) \sum_{1}^{L_n} B_{i,n} \tag{3.6}$$

$$= (L_n - 1) B_n \tag{3.7}$$

Noting that capacity is conserved for all links incident on a node we see

$$L_n C = P_n + B_n \tag{3.8}$$

$$(L_n - 1)B_n + B_n \tag{3.9}$$

$$= L_n B_n \tag{3.10}$$

$\therefore B_n = C$ as claimed.

Given this lemma the K nodes in a network would collectively see KC units of capacity being reserved at minimum. Since each link is incident on exactly two nodes this figure double counts the capacity needed, thus the minimum required backup capacity is $KC/2$. Or equivalently the maximum possible primary is $LC - KC/2$.

3.2 Achievability of Bounds

As noted above, the bound given in Section A may or may not be achievable, depending on the network topology. In this section, we give examples to illustrate this point.

The bound given in section A can be achieved by the network in Figure 3.2. The network configuration that can achieve this bound allocates primary capacity $C/2$ to links 1-2,2-3,3-4,4-1 and primary capacity C to link 1-3. This gives a total primary capacity of $4 * C/2 + C = 3C$. Using the bounding expression from section A the value calculated is $5C - 4C/2 = 3C$ which is exactly the total primary capacity achievable by the network configuration.

The bound given in section A cannot always be achieved, and this will be illustrated using the topology in Figure 3.3. In this topology, the links are effectively grouped into pairs where each link in the pair protects the other link in the same pair. For instance if link 1-2 fails, the only link that could pick up its traffic is link 2-5. The reverse is also true, if link 2-5 fails only link 1-2 can back it up. Link pairs (1-3, 3-5) and (1-4, 4-5) behave in a similar fashion. From this it can be clearly seen that the way to maximize the total primary capacity on the network is to set all link pairs to $C/2$ primary capacity. This gives a total of $6 * C/2 = 3C$ primary capacity. However the bound computed using the formula from section A is $6 * C - 5 * C/2 = 3.5C$. This is $C/2$ greater than is possible given the topology of the network.

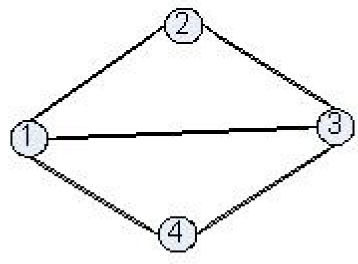

Figure 3.2 Can reach bound of capacity argument

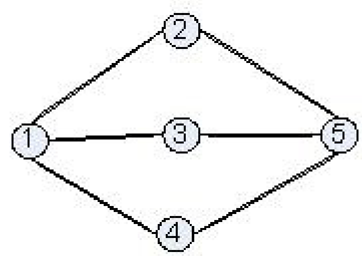

Figure 3.3 Topology prevents achieving capacity based bound

3.3 Relevance to p-Cycles

When a network contains a Hamiltonian cycle, it can be used as a p-Cycle to protect the network. Assuming the Hamiltonian p-Cycle uses $C/2$ on every on-cycle links, it provides the maximum possible primary capacity any link-based scheme can achieve.

This is not an original result in that [15] shows a p-Cycle where the straddling links have twice the primary capacity as the on-cycle links achieves the minimum redundancy. The idea is emphasized here since it provides motivation for chapter 4 where Hamiltonian p-Cycles are used as a capacity optimal solution when examining the effects of backup capacity distribution on blocking probability.

3.4 Link-based Protection: Partial traffic recoverable

The above bound can be generalized to the case of only guaranteeing protection to a fraction f of the traffic on a given link. In this case, the constraints become

$\forall i \in \{1, \ldots L\}$

$$P_{i,n} + B_{i,n} \leq C \tag{3.11}$$

$$f * P_{i,n} \leq \sum_{i \neq j} B_{j,n} \tag{3.12}$$

Once the protection factor has propagated though the equations the end result is a network's primary capacity that is upper bounded by:

$$EC - \frac{K}{2} \frac{L_n C f}{L_n - 1 + f} \tag{3.13}$$

3.5 Multiple Link Failures

The results from section 3.1 can be generalized to the case of allowing m link failures. To see this, let the set of links on a node be divided into two sets, one set D represents an arbitrary set of m links that might be destroyed. The other set R represents the rest of the links and contains all the links not in D.

From link capacity being conserved the following hold:

$$\sum_{i \in D} P_{i,n} + \sum_{i \in D} B_{i,n} = mC \tag{3.14}$$

$$\sum_{i \in R} P_{i,n} + \sum_{i \in R} B_{i,n} = (L_n - m)C \tag{3.15}$$

Since the links in D must be protected a third constraint is:

$$\sum_{i \in D} P_{i,n} \leq \sum_{i \in R} B_{i,n} \tag{3.16}$$

To find the maximum possible useful network load:

$$Maximize \sum_{i=1}^{L_n} P_{i,n} \tag{3.17}$$

This sum is maximized when the third constraint becomes a strict equality. It can then be substituted into the first constraints to get:

$$\sum_{i \in R} B_{i,n} + \sum_{i \in D} B_{i,n} = mC \tag{3.18}$$

Similar to Section 3.1, this is minimum backup capacity seen by a node. Removing the total backup capacity from the total capacity and remembering to divide by two to avoid over counting, one can see an upper bound on the total primary capacity as:

$$EC - \frac{mKC}{2} \tag{3.19}$$

3.6 Similar Bound

In [4] bounds on backup resources have also been investigated. In [4], Grover derived the quantity bound is the redundancy which is defined as:

$$\frac{\sum B_{i,n}}{\sum P_{i,n}} = redundancy \tag{3.20}$$

The bound on a network given in terms of redundancy is given as

$\frac{1}{\bar{d}-1}$ where $\bar{d}$ is the average degree of a node.

At first glance, the reader may be tempted to conclude the results of section 3.2 are a previously known bound derived in an overly complicated way. With a little algebraic manipulation we see:

$$\frac{\sum B_{i,n}}{\sum P_{i,n}} = \tag{3.21}$$

$$\frac{\frac{cn}{2}}{EC - \frac{cn}{2}} \tag{3.22}$$

$$\frac{\frac{cn}{2}}{EC - \frac{cn}{2}} * \frac{\frac{2}{cn}}{\frac{2}{cn}} \tag{3.23}$$

$$\frac{1}{\frac{2E}{n} - 1} \tag{3.24}$$

$$\frac{1}{\bar{d} - 1} \tag{3.25}$$

And this seems to match the previous result. However, the bound in [4] is derived under much different conditions than those made in this thesis. Table 1 gives a breakdown of the differences between the networks to which the bound has been proven applicable.

	This paper	Else and Grover paper
Primary Capacity	May vary per link. Values on any given link can range from zero to the total capacity of the link	All links have the same primary capacity introduced as a constant.
Total Link Capacity	All links have same total capacity introduced as constant.	Not addressed in proof. Assumes the ability to add capacity to links as needed.
Explicitly Required Network Attributes	All links have same total capacity.	All links have same primary capacity. All nodes have same degree.

Table 3.1 Assumptions Used In Bounding Primary Capacity

Besides the above there is an implicitly network attribute required to reach the bound in that the Link capacity's must have infinite granularity and thus can be modeled as real numbers or if finite granularity, then the sum of the primary bandwidth across all links must be a multiple of the node degree minus one.

Thus our result is a generalization to bounds in [4]. The lemma of section 3.1 and the resulting limit on redundancy in this section were recently shown in [16]. The derivation in [16] is however different than the ones presented here. The derivation shown in this paper easily generalized to the results for partial protection and multiple link failures.

CHAPTER 4. LOWERING BLOCKING PROBABILITY

One approach for minimizing the probability that a connection request is blocked in a fault tolerant network is to reserve as little capacity for backup as possible. For a number of networks p-Cycles have reached the fundamental limits of this approach. This chapter examines how to achieve further improvement by distributing backup capacity in a more even fashion.

The intuition for evenly distributing backup capacity is: links that have a high percentage of their capacity reserved for backup can become completely used much more quickly than links which only devote a small amount of their capacity to backup. When a link becomes full later requests can be blocked, or take longer routes which will increase the speed at which other links reaching capacity faster and causing blocking.

This chapter examines distributing backup capacity in three steps. The first step is to give an ILP in section 4.1 that finds the maximum possible primary capacity a network can provide. This ILP has the disadvantage that a backup route could be potentially long, which is undesirable since lengthy paths can lead to signal degradation. This issue is addressed by section 4.2 where the ILP is expanded to include a concept of distributed p-Cycles link protection (DPLP). In section 4.2, for each link failure scenario the ILP finds a set of p-Cycles that can be dynamically configured from the reserved backup capacity. These cycles are computed outside of the ILP and range in size up to some fixed maximum length. This is how short backup route lengths are ensured.

The results of the ILP from section 4.1 can be used to calculate how close the ILP from section 4.2 got to the optimum using the cycle set it was given. Section 4.3 examines the difference between p-Cycles and DPLP in terms of network blocking probability. Blocking probabilities are determined by simulation.

4.1 Baseline ILP

To maximize the primary capacity of a network with L links, the ILP in this section divides the task into L sub-problems where each sub-problem is that of finding the backup route(s) to use to protect one of the L links. It is assumed that the flow of information across a link may be split into several paths when it is re-routed because of a link failure.

Each sub-problem is formulated as a flow problem on a network where the capacities of the links are the backup capacities, except a unique link is omitted in each sub-problem. In each sub-problem the nodes that were connected by the omitted link act as a source or sink. One of the nodes acts as a source with a supply equal to the primary capacity of the omitted link, and the other acts as a sink with demand equal to the primary capacity of the omitted link.

The ILP maximizes the sums of the primaries, while requiring the total space allocated between primary and backup on a link to be less than the total capacity of a link. All links were assumed to have the same capacity but this assumption could easily be removed. Link capacity is assumed to be only finitely dividable, if this assumption were to be dropped, the problem would reduce to an LP.

The ILP has three sets of variables, one set P_e represents primary capacity allocated on the links, one set B_e represents backup capacity allocated on the links, and the third set $F_{e,g}$ represents the flows. Links are numbered in [1, L] and the index 'e' on a variable is in reference to the number given a link. Flow sub-problems are also numbered [1, L] where the number given a flow sub-problem corresponds to the link that has been omitted. The index 'g' is used to indicate the number of a flow sub-problem. The index 'n' indicates the node number numbered [1, K].

The mechanism of removing a link from a flow sub-problem is by multiplying every instance of it by zero, effectively causing the variable not to exist. The sign of the flow variable determines the direction of the flow in a link. The values of $d_{n,e}$ is zero if link e is not connected to node n, and for the two times link e is incident on a node n, $d_{n,e}$ takes on the value of one for the lower node number and negative one for the higher node number.

Objective:

$$Maximize \sum_{e=1}^{L} P_e \tag{4.1}$$

Subject to:

The first equation in the ILP says that flow is conserved except for the source or sink. That is the flow into a node is equal to the flow out of the node. The chronic delta in the equation causes a node to be a source or sink with supply or demand equal to the capacity of the link omitted from the network.

$$\sum_{e=1}^{L}((1-\delta_{e,g})d_{n,e}F_{e,g} - \delta_{e,g}d_{n,e}P_e) = 0 \tag{4.2}$$

The second and third inequalities ensure that the backup capacity on a link is large enough for the flow that wants to cross the link. This is required since the flow represents a backup path to be used if the link fails.

$$F_{e,g} \leq B_e \tag{4.3}$$

$$-F_{e,g} \leq B_e \tag{4.4}$$

The combined primary and backup capacity allocated on a link cannot be greater than the total capacity on the link.

$$B_e + P_e \leq capacity \tag{4.5}$$

Variables are restricted in their range by application demand.

$B_e, P_e \in [0, capacity]$

$F_{e,g} \in [-capacity, capacity]$

$d_{n,e} \in \{-1, 0, 1\}$

4.2 DPLP ILP

This section explains the DPLP ILP that selects backup capacities, so upon link failure p-cycles of a fixed maximum length can be formed to protect the failure.

The DPLP ILP has been formulated as an expanded version of the one in section 4.1. In the expansion the ILP sub-problems have had the additional constraint added that each unit of backup flow is constrained to use one of a number of pre-computed cycles that have a fixed maximum length. These cycles become the DPLP cycles. To prevent the ILP from selecting all possible cycles in a given sub-problem, a penalty is added for each instance of a cycle selected. The reward for increasing primary capacity is set much higher than the penalty for adding more backup cycles to ensure increasing primary capacity always more than offsets the penalty for adding an extra cycle.

Objective:

$$Maximize \;\; capacity^2 \sum_{e=1}^{L} P_e - \sum_{g=1}^{L} \sum_{c=1}^{cycles} Cy_{c,g} \tag{4.6}$$

Subject to:

Equations (4.7)-(4.10) perform the same function as equations (4.2)-(4.5) in the basic ILP.

$$\sum_{e=1}^{L} ((1-\delta_{e,g}) d_{n,e} F_{e,g} - \delta_{e,g} d_{n,e} P_e) = 0 \tag{4.7}$$

$$F_{e,g} \leq B_e \tag{4.8}$$

$$-F_{e,g} \leq B_e \tag{4.9}$$

$$B_e + P_e \leq capacity \tag{4.10}$$

Equation (4.11) ensures that the backup capacity on a link is large enough to form all the DPLP cycles that may exists at the same time.

$$B_e \geq (1 - \delta_{e,g}) \sum_{c=1}^{cycles} BIC_{c,g} LIC_{e,c} Cy_{c,g} \tag{4.11}$$

Equations (4.12)-(4.14) cause a variable $f_{e,g}$ to perform the role of a per link capacity for the various sub-problems. In a sub-problem, the capacity of a link is the sum of all the DPLP cycles that become active upon the failure of the link being protected in the sub problem. This is enforced by equation (4.12). Equations (4.13) and (4.14) ensure $F_{e,g}$ sees $f_{e,g}$ as a capacity limit.

$$f_{e,g} \leq (1 - \delta_{e,g}) \sum_{c=1}^{cycles} BIC_{c,g} LIC_{e,c} Cycles_{c,g} \tag{4.12}$$

$$f_{e.g} \geq -F_{e,g} \tag{4.13}$$

$$f_{e.g} \geq F_{e,g} \tag{4.14}$$

Variables' limits and values as dictated by the problem constraints and formulation.

$B_e, P_e \in [0, capacity]$

$F_{e,g} \in [-capacity, capacity]$

$f_{e,g} \in [0, capacity]$

$d_{n,e} \in \{-1, 0, 1\}$

$BIC_{c,g} \in \{1, 0\}$

$LIC_{e,c} \in \{1, 0\}$

$Cy_{c,g} \in [0, capacity]$

$BIC_{c,g}$ is short for $BothInCycle_{c,g}$

This takes the value of 1 if both ends of the omitted link are in cycle c, and zero otherwise.

$LIC_{e,c}$ is short for $LinkInCycle_{e,c}$

This takes the value of 1 if link e is in cycle c, and 0 otherwise.

$\delta_{e,g}$=1 iff e=g, otherwise $\delta_{e,g}$=0

The value of $Cy_{c,g}$ is amount of backup flow following this particular cycle. The maximum value is arbitrary. Setting it to a low value can encourage the ILP to protect a link with multiple cycles. Of course there is no guarantee even with a low value that every link will carry primary capacity greater than the limit and thus need to be split.

It should be noted that when two different choices of maximum cycle length result in the same total primary capacity used, and there are multiple equivalent solutions, changing the maximum cycle length may result in a different solution being found by the ILP.

4.3 Solutions by Cycle Length

The performance of the DPLP ILP was evaluated for Arpanet, NSF net, and Cost 239. All cycle lengths were tested for Arpanet and NSF net, but only a few were tested for Cost 239 since the size and running of the ILP where much greater for this network than the other two. In some cases such as cycle maximum length 3,4,5 or 6 in NSF net the ILP finds solutions that waste link capacity by not allocating it as primary or backup capacity. However as cycle length is increased, the ILP finds the capacity optimal solutions for all networks. All links have a capacity of 10.

Figure 4.1 shows at a cycle length of eight DPLP cycles can reach 96.6% of the maximum possible primary. The maximum primary is achieved at cycle length fourteen.

Figure 4.2 shows a DPLP cycle length of seven can reach 97.3% of the maximum possible primary capacity. The theoretical maximum is reached at cycle length fourteen.

Figure 4.3 shows that with only a cycle length of three over half the maximum possible primary is available and that only a cycle length of five is required to reach the maximum possible primary.

These results in general show there is a threshold cycle length, where increasing the allowed cycle length beyond this point does not result in more primary capacity.

Thresholds have occurred in other context. In [17] a network was protected with backup routes with a fixed maximum length. Results in that paper showed that past a threshold, increasing the allowed backup path length did not lower the backup resources required. This work differs from [17] in that [17] fixed primary capacities and then allowed backup capacity to be independently added. In [18] a threshold limit is shown for p-cycles where capacity could be added and the placements of primary and backup capacity were jointly done.

Since this work assumes a total fixed capacity, the primary capacities are in fact functions of the backup capacities and visa versa. Thus it is interesting to see that a threshold limit exists in this context also.

4.4 Example Solution

Figure 4.1 shows primary capacities found for Cost 239 when the maximum cycle length is five. Table 4.1 gives statistics on how capacity is distributed over the links. While each link is protected by a set of small DPLP cycles that will be dynamically configured upon failure, the amount of primary capacity in the network reaches the theoretical maximum.

Table 4.1 Summery of link primary capacities

Primary Capacity	Number of Links
9	4
8	17
7	3
6	2

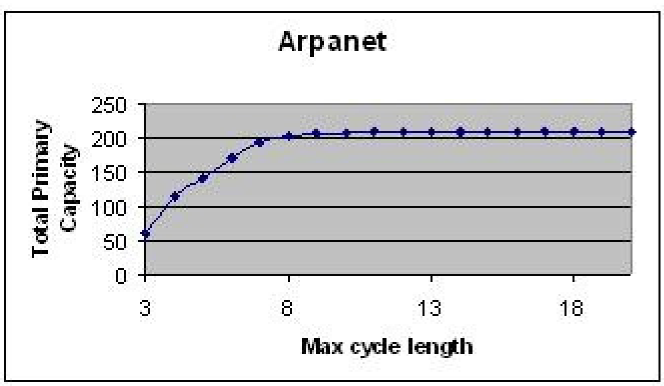

Figure 4.1 Arpanet primary capacities vs longest DPLP cycle length

Listing all the p-Cycles the ILP found to protect each link would be long and tedious. However a much smaller set gives a good illustration of the type of solutions found the ILP. For the three examples shown below, the first column in the table is the number of units of capacity reserved for that cycle. Notice that the link, which is being protected on the cycle, does not reserve any backup capacity on itself.

Link 3 to 5

Cycle Capacity	Cycle
2	11-5-3-4-10
2	1-3-5-6-7
1	2-3-4-8-5
1	2-3-6-5

Link 3 to 4

Cycle Capacity	Cycle
2	11-5-3-4-10
2	1-3-2-8-4
1	3-4-7-9-6

Link 8 to 9

Cycle Capacity	Cycle
1	11-8-5-6-9
2	1-2-8-9-7
1	4-8-6-9-10
1	11-8-4-10-9

4.5 Blocking Probabilities

To examine how blocking is affected by varying the distribution of primary capacity each network was repeatedly simulated using different distributions of the primary capacity. Some of these distributions were p-Cycles that used Hamilton cycles. Others were results generated by the ILP for various fixed maximum cycle lengths.

The first step of the simulation was for a given arrival rate to process a 100k request to get the network into a random state. After this a number of rounds were simulated, where

10k events occurred without any statistics being gathered, then 90k events were simulated and blocking probability was calculated. The blocking probability for each round was stored in a text file, and data analysis tools in Excel were used to calculate the average blocking probability and the confidence with alpha being set to 0.05. The number of rounds simulated was one hundred.

Requests for the network were randomly generated with a bandwidth requirement of one to four. Routing was done by finding the shortest route in the network between source and destination only considering links that had enough bandwidth free to accommodate the requested bandwidth. Results of these simulations are given below. Hamiltonian cycles were numbered simply to distinguish between them.

Figure 4.5 depicts that DPLP achieves a lower average blocking probability than a p-Cycle using one Hamiltonian. The difference between the two increases as arrival rate increases. This is because as links reach capacity, connections have to take longer routes; which consumes the remaining primary capacity faster.

Figure 4.6 depicts that the choice of Hamiltonian cycle can greatly affect the difference in performance between p-Cycle and PDLP cycle approaches. This is because some links may tend to be used more on shortest routes than others, and thus are more important that they provide a high amount of primary capacity.

Figure 4.7 shows that due to high connectivity and low blocking probability of Cost 239 network the difference between p-Cycle and DPLP is not significant till much higher arrival rates in comparison to NSF or Arpanet.

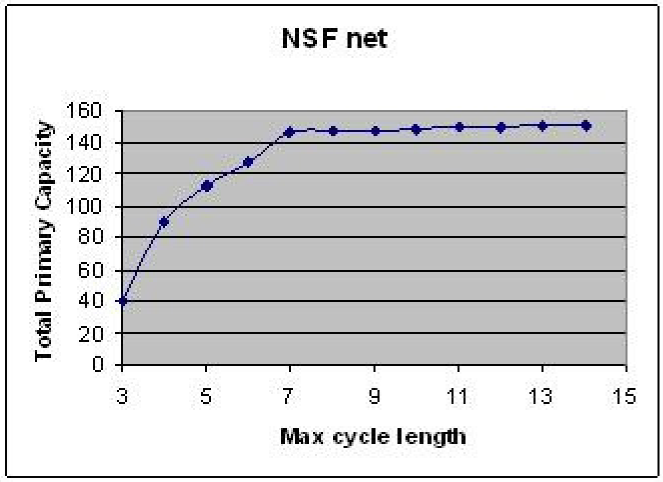

Figure 4.2 NSF net primary capacities vs longest DPLP cycle length

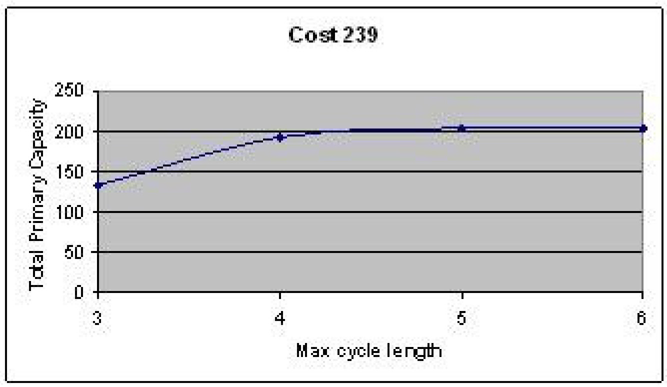

Figure 4.3 Cost 239 primary capacities vs longest DPLP cycle length

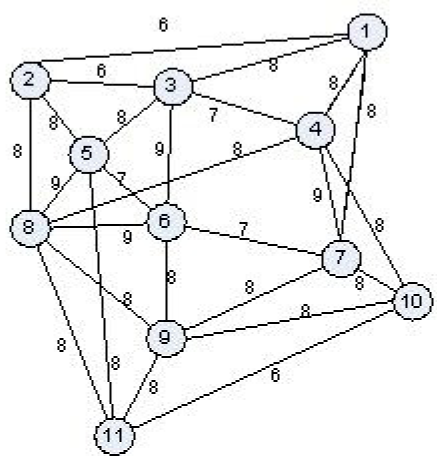

Figure 4.4 Primary Capacities on Cost 239 Network

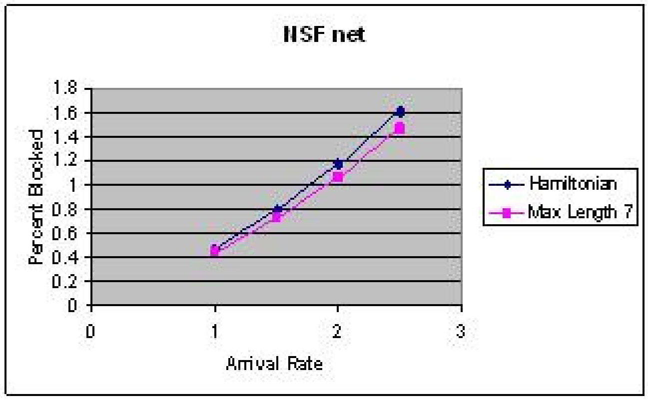

Figure 4.5 Blocking percentages NSF net

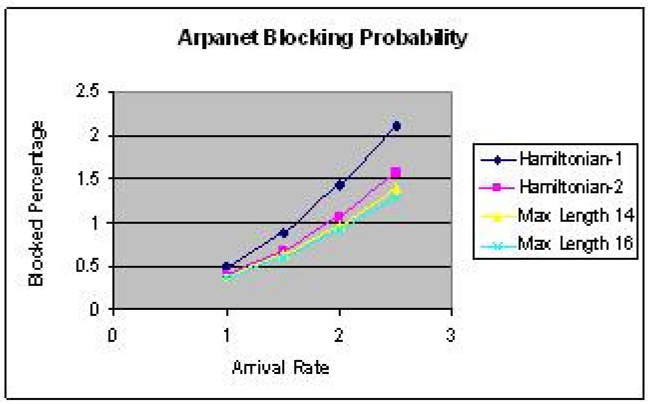

Figure 4.6 Blocking percentages Arpanet

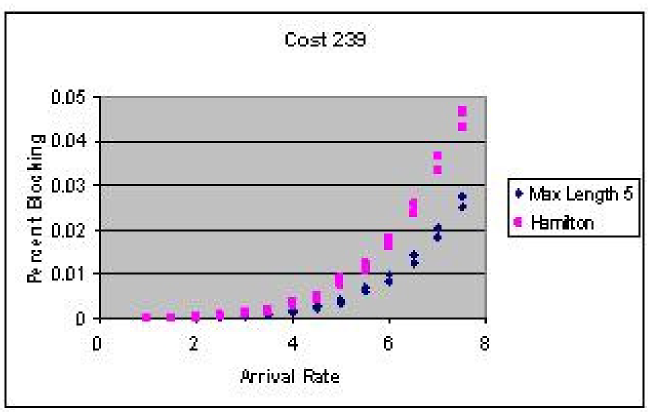

Figure 4.7 Blocking percentages Cost 239

CHAPTER 5. SUMMARY AND DISCUSSION

This chapter examines the effect of pre-configuration on available primary capacity. First an ILP that can compute a couple different types of preconfigured structures is motivated in section 5.1. The ILP itself is then given in section 5.2. Section 5.3 examines the resulting primary capacities produced by the ILP.

5.1 Motivation

Consider the case where the primary capacity on the on-cycle links, not the straddling links, is the dominating factor in determining the amount of backup capacity that needs to be reserved on the on-cycle links. Then every on-cycle link in a p-Cycle must reserve backup capacity equal to the primary capacity of the on-cycle link with the most primary capacity. If the maximum required primary occurs only on one link, then every link in the p-Cycle reserves bandwidth that will only be used in one particular failure scenario.

In this case, some of the backup capacity reserved on the link requiring the most primary capacity will be wasted since no failure scenario where the link is functioning requires the entire reserved backup capacity. If the difference in primary bandwidth required between the link requiring the most and the next largest amount could be handled outside of the p-Cycle, backup bandwidth reversed in the network could be potentially lowered.

This handling of this extra capacity can be done by adding a pre-configured line that starts at the beginning of one end of the link needing the most protection follows the p-Cycle except it uses straddling links as shortcuts and reaches the other end of the link getting supplemental protection. Each time a straddling link is used as a shortcut at least two on-cycle links are bypassed thus reducing the number of links responsible for covering the difference in capacity.

Two straddling links may be sufficient to bypass all on-cycle links provides the two links and the one being protected form a cycle.

Besides reducing backup capacity used, adding pre-configured lines can allow for traffic demands in networks to be protected that could not be before. An example of this is the network shown in Figure 5.1 with traffic demands as given in Table 5.1. Figure 5.1 shows the non-simple p-Cycle that can protect the network if it is supplemented with a line. The p-Cycle uses two units of bandwidth and the line use one unit of bandwidth. All links have a capacity of five. Even with joint optimization demands use the link directly connecting communicating nodes.

Node to Node	Bandwidth requested
1 to 2	5
2 to 3	2
3 to 4	3
1 to 5	2
1 to 4	2
2 to 4	2
4 to 5	2

Table 5.1 Traffic Demand For Figure 5.1

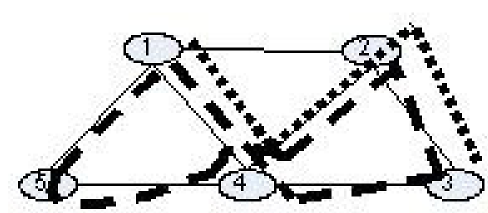

Figure 5.1 A Line And A Non-simple p-Cycle

5.2 Non-Simple Cycle and Line ILP

The naive way to compute a non-simple p-Cycle would involve keeping track of the order in which links are used to visit a node. This complexity can be avoided by assuming nodes on a non-simple p-Cycle only have two paths between them. With this assumption, when selecting links to form a non-simple p-Cycle / line it is sufficient to ensure the links are selected in such a way that they can form a Euler tour. The links selected by the ILP can then be used to compute a non-simple p-Cycle or line in linear time.

An Euler tour is a tour of a graph that crosses each edge exactly once. It is necessary and sufficient that a graph have at most two nodes of odd degree for there to be an Euler tour. Or in the case of selecting links for use in p-Cycles or lines only two nodes can have an odd number of links selected for the construction.

The fact cycles and lines satisfy many of the same requirements locally is reflected in the notation of the ILP in that variable names often appear twice in the formulation with different capitalization and with one changed index. When a variable appears in both capitalized and lowercase form, the capitalized variable is being used to compute a cycle and the lower case variable to compute a line.

Due to the number of indexes and variables, it is hard to give each one an intuitive name. The following chart explains the purpose of each index and variable.

Index	Set it indexes	Range
L	The set of all links in the network. This term is used interchangeable with edges.	[1 , number of links in the graph]. In the ILP the second is represented by the constant "edges"
c	The set of possible cycles	[1 , the largest number of cycles the ILP is allowed to calculate]
L	The set of possible lines	[1 , the largest number of lines the ILP is allowed to calculate]
S	All edges used in a particular cycle / line have the same capacity reserved. The index s ranges over the set of capacities a cycle/line is allowed to use.	[1 , log_2(total link capacity)] The possible capacities are increasing consecutive powers of two starting at 2^0.
n	The set of nodes in the network.	[1 , number of nodes in the network]

Table 5.2 Non-simple p-Cycle / line indexes

$$Maximize \sum_{l=1}^{edges} P_l \tag{5.1}$$

A cycle has an even number of edges incident on a node.

$$\sum_{l=1}^{edges} C_{l,c}E_l = 2H_{l,c,s} \tag{5.2}$$

Both sides of a link must be incident on nodes in the cycle to be protected.

$$C_{n,l}S_{l,c,s} \leq H_{l,c,s} \tag{5.3}$$

Connectivity of the cycle is checked by (4) – (9)

All nodes sink some flow proportional to the number of links incident on them, but one also acts as a large sources. Flow into a node is the same as out of a node except for the amount it sinks or sources.

$$\sum_{l=1}^{edges} D_{l,n}F_{l,c,s} = T_{l,c,s} - H_{l,c,s} \tag{5.4}$$

The amount to be sourced by the one node is calculated to just meet the demands of all the sinks.

$$\sum_{n=1}^{nodes} H_{l,c,s} = \sum_{n=1}^{nodes} T_{l,c,s} \tag{5.5}$$

One node is selected to be the node that produces the source to all nodes in the cycle.

$$\sum_{n=1}^{nodes} R_{l,c,s} \leq 1 \tag{5.6}$$

Links used by the cycle are seen as having a capacity larger than the largest possible flow. Links not used are seen as having capacity zero.

$$C_{l,c}F_{l,c,s} \leq 2 * edges * H_{l,c,s} \tag{5.7}$$

$$-C_{l,c}F_{l,c,s} \leq 2 * edges * H_{l,c,s} \tag{5.8}$$

Only the selected source node may have a non-zero source. This inequality does not affect the maximum possible value of the source at the one node.

$$T_{l,c,s} \leq 2 * edges * R_{l,c,s} \tag{5.9}$$

Now in a similar fashion to cycles, the equations for lines are:

A potentially non-simple line has an even number of edges incident on a node, except when the O variable allows it to have an odd degree.

$$\sum_{l=1}^{edges} c_{l,L} e_l = 2h_{l,L,s} + O_{n,L,s} \tag{5.10}$$

Both sides of a link must be incident on nodes in the line to be potentially protected.

$$C_{n,l} S_{l,c,L} \leq h_{l,c,L} \tag{5.11}$$

All nodes sink some flow proportional to the number of links incident on them, but one also acts as a large sources. Flow into a node is the same as out of a node except for the amount it sinks or sources.

$$\forall n, \sum_{l=1}^{edge} D_{l,n} f_{l,L,s} = t_{l,c,L} - h_{l,L,s} - O_{n,L,s} \tag{5.12}$$

The amount to be sourced by the one node is calculated to just meet the demands of all the sinks.

$$\sum_{n=1}^{nodes} h_{l,L,s} + O_{n,L,s} = \sum_{n=1}^{nodes} t_{l,L,s} \tag{5.13}$$

One node is selected to be the node that produces the source to all nodes in the cycle.

$$\sum_{n=1}^{nodes} r_{l,L,s} \leq 1 \tag{5.14}$$

Links used by the line are seen as having a capacity larger than the largest possible flow. Links not used are seen as having capacity zero.

$$C_{l,c} f_{l,c,L} \leq 1 * edges * h_{l,c,L} \tag{5.15}$$

$$-C_{l,c}f_{l,c,L} \leq 1 * edges * h_{l,c,L} \tag{5.16}$$

Only the selected source node may have a non-zero source. This inequality does not affect the maximum possible value of the source at the one node.

$$t_{l,L,s} \leq 2 * edges * r_{l,L,s} \tag{5.17}$$

Only two nodes in the line are allowed to have an odd degree. These will be the starting and ending nodes of the line. It's allowed for the number of end nodes to be zero to allow the line not to be used, or form a circle. In the later case, despite the fact it's actually a cycle it will still only be counted as giving the protection a line gives.

$$\sum_{n=1}^{nodes} O_{n,L,s} \leq 2 \tag{5.18}$$

A link must have all its primary capacity protected. When both ends of a link are on a cycle it gets protection twice the reserved cycle bandwidth for protection, unless its an on-cycle link then it only gets the reserved bandwidth. When both ends of a link are on a line it gets protection equal to the reserved capacity, unless it's in the line in which case it gets none.

$$P_l \leq \sum_{s=1}^{log(capacity)} \sum_{c=1}^{cycles} (2W_sS_{l,c,s} - w_sE_{l,c,s}) + \sum_{s=1}^{log(capacity)} \sum_{L=1}^{lines} (w_ss_{l,L,s} - w_se_{l,L,s}) \tag{5.19}$$

A nodes backup capacity is equal to the amount of capacity reserved on it by all p-Cycles and links.

$$B_l = \sum_{s=1}^{log(capacity)} \sum_{c=1}^{cycles} w_sE_{l,c,s} + \sum_{s=1}^{log(capacity)} \sum_{L=1}^{lines} w_sE_{l,L,s} \tag{5.20}$$

The primary and backup capacity on a link can not exceed its total capacity.

$$P_l + B_l \leq capacity_l \tag{5.21}$$

It should be noted, that in order to make the solution space as large as possible this ILP can consider pre-configured structures it can not use to optimality. If a network can be protected by non-simple p-cycles that provide at most two backup paths and simple lines, the ILP will always find the solution.

There is no fundamental reason why a non-simple p-Cycle need only provide two backup routes. If a non-simple p-Cycle happened to be the union of two link disjoint simple p-Cycles that shared a straddling link, the non-simple p-Cycle could provide four backup paths for this particular link. This ILP would allow such a solution; however the straddling link would only be counted as being backed up by two backup paths.

If the value of $h_{l,L,s}$ is allowed to be an integer instead of a binary variable then the line may be a non-simple line. Some of the links it protects may then end up with more than one backup path, however the ILP will only count the line as providing one backup path, and no backup path if its an on-line link. For the test runs of the ILP discussed in the next section, a line was allowed to cross a node twice, but none of the considered topologies / primary capacity distributions has wide enough variations for a line to appear in the ILPs solutions.

5.3 Pre-Configuration Performance

From section 3.4 one can see that many of the networks historically popular with researchers such as NSF net, Arpanet, and the pan Europe network Cost 239 all have Hamiltonian cycles and can thus achieve the maximum possible primary capacity even when restricted to pre-configured backup strategies.

Besides simple maximizing primary capacity, the effect of protecting a specified traffic distribution was also examined. The ILPs from sections 4.1 and 5.2 where modified to have the primary capacity calculated so that they could support a specified traffic matrix. Then the Cost 239 network with a traffic distribution from [14] was used to calculate the minimum total capacity, which is primary plus backup, needed to protect the network. All links were taken to have a capacity of 30. The non-preconfigured ILP finished quickly with the result of 358 units required. The ILP from section 5.2 reached a value of 361 after a day of running with a gap of 1.45%.

Variable	Indexes	Purpose
E/e	l, c/L, s	Takes the value one if 1 if link l is used on cycle c / line L, reserving backup capacity 2^{s-1}. Zero otherwise.
H/h	n , c/L, s	Used to count the number of times a cycle or line of a given size has visited the node n. This variable can range from 0 to a small integer with one or two being reasonable values. Often multiplied by two, so its value is one half the number of times a cycle visits a node, or one half minus one the number of times a line visits.
R/r	N , c/L, s	All nodes in a cycle / line must be reachable using only selected links from one particular node. This variable takes the value 1 if this node is that node, 0 otherwise.
F/f	l, c/L, s	Reach ability of the R/r node is checked in the ILP by a flow problem where the R/r node is the only source. This variable represents the flow along edges in the problem.
T/t	N , c/L, s	Used to calculate the total amount of flow the R/r node should produces in the cycle/line connectedness check.
O	n,L,s	Takes the value of one if a node is allowed to be visited an odd number of times by the line l with 2^{s-1} units of capacity. Zero otherwise.
S/s	l, c/L, s	One if both ends of link l are on nodes used in a cycle or line. 2S-E is the number of backup paths provided by a cycle. s-e is the number of backup paths provided by a line.

Table 5.3 Non-simple p-Cycle / line variables

Constant	Indexes	Values and Purpose
C	n,l	1 if link l is incident to node n, 0 otherwise.
D	n,l	1 if flow on link l comes into the node n , -1 if flow leaves node n, and zero if the link is not incident on node n. Since flow is allowed to take both positive and negative values, the values of d are the directions if the flow variable is positive.
w	s	The amount of capacity reserved by a cycle or line. Values range from 2^0 to 2^{s-1} in consecutive powers of 2.
capacity	l	The total capacity on link l.

Table 5.4 Non-simple p-Cycle / line constants

BIBLIOGRAPHY

[1] http://en.wikipedia.org/wiki/Mail#Early_postal_systems Accessed Nov 21, 2006.

[2] http://www.nathan.com/projects/current/comtimeline.html Accessed Nov 21, 2006.

[3] http://www.earlyamerica.com/lives/revere/chapt3/ Accessed Nov 20, 2006.

[4] W. Grover. "*Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking*" Prentice Hall PTR, Upper Saddle River, New Jersey, 2004. pp. 105,111,315

[5] Rajiv Ramaswami , "Multiwavelength Lightwave Networks for Computer Communication" *IEEE Communications Magazine* pp. 78-88, February 1993

[6] D.Zhou, S.Subramanian, "Survivability in optical networks," *IEEE Network*, Nov/Dec 2000

[7] G.Mohan, C.S.Murthy, "Lightpath Restoration in WDM optical networks," *IEEE Network*, Nov/Dec 2000.

[8] Michael T. Frederick , Pallab Datta, and Arun K. Somani, "Sub-graph routing: A generalized fault-tolerant strategy for link failures in WDM optical networks", *Computer Networks*, 2005.

[9] W. D. Grover and D. Stamatelakis, "Cycle-oriented distributed preconfiguration: rink-like speed with mesh-like capacity for self-planning network restoration," *In Proc. Of IEEE ICC 1998*, pp. 537-543, June 1998.

[10] Gruber, C.G. "Resilient Networks with Non-Simple p-Cycles", *International Conference on Telecommunications (ICT 2003),* Feb/Mar, 2003.

[11] Lachowicz *et al.* "Joint Optimization in Capacity Design of Networks with p-Cycle Using the Fundamental Cycle Set" *in Proc. of IEEE Globecom 2006,* Nov 2006.

[12] W. D. Grover, J. Doucette, "Advances in Optical Network Design with p-Cycles: Joint Optimization and Pre-Selection of Candidate p-Cycles," *Proc. Of IEEE/LEOS Summer Topicals 2002*, pp. 49-50, July 2002.

.

[13] Schupke, D. A. “An ILP for optimal p-cycle selection without cycle enumeration,” *Eighth IFIP Working Conference on Optical Network Design and Modeling (ONDM 2004) ,* Feb 2004

[14] W. He, J. Fang and A. K. Somani, "A new p-cycle based Survivable Design for Dynamic Traffic in WDM Networks" *In Proc. of IEEE Globecom 2005*, Nov 2005

[15] A. Sack, W. D. Grover, “Hamiltonian p-Cycles for Fiber-Level Protection in Homogeneous and Semi-Homogeneous Optical Networks” *IEEE Network*, 2004

[16] Schupke, D.A.,"Analysis of p-Cycle Capacity in WDM Networks," *Photonic Network Communication*s, July 2006.

[17] M. Herzberg, S. Bye, “An optimal spare-capacity assignment model for survivable networks with hop limits” *Proc. IEEE GLOBECOM 1994* pp. 1601-1607, Nov/Dec 1994

[18] Adil Kodian, Anthony Sack, Wayne D. Grover "p-Cycle Network Design with Hop Limits and Circumference Limits" *BROADNETS 2004* , pp 244-253, Oct 2004

[19] D. Stamatelakis, W.D. Grover “Network Restorability Design Using Pre-configured Trees, Cycles, and Mixtures of Pattern Types” TR*Labs*, Edmonton, Alberta, Canada, Technical Report TR-1999-05, Issue 1.0, October 2000.