

1-1-2002

## A single-fault recovery strategy for optical networks using subgraph routing

Michael Todd Frederick  
*Iowa State University*

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

---

### Recommended Citation

Frederick, Michael Todd, "A single-fault recovery strategy for optical networks using subgraph routing" (2002). *Retrospective Theses and Dissertations*. 19850.  
<https://lib.dr.iastate.edu/rtd/19850>

This Thesis is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact [digirep@iastate.edu](mailto:digirep@iastate.edu).

**A single-fault recovery strategy for optical networks using  
subgraph routing**

by

**Michael Todd Frederick**

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  
Mani Mina  
Spehen Vardeman

Iowa State University

Ames, Iowa

2002

Copyright © Michael Todd Frederick, 2002. All rights reserved.

Graduate College  
Iowa State University

This is to certify that the master's thesis of  
Michael Todd Frederick  
has met the requirements of Iowa State University

Signatures have been redacted for privacy

*To my parents,  
who have never done anything that wasn't for my brother and me.*

## TABLE OF CONTENTS

Table of Contents	iv
List of Figures	vi
List of Tables	vii
Abstract	viii
1. Introduction	1
2. Literature Review	2
2.1. Backup Multiplexing	2
2.2. Primary-Backup Multiplexing	3
3. $L+1$ Fault Tolerance Routing Strategy	4
3.1. The Key Characteristics of $L+1$ Fault Tolerance	4
3.2. The Key Assumptions of $L+1$ Fault Tolerance	5
3.3. What is $L+1$ Fault Tolerance?	5
3.3.1. A Mathematical Description	5
3.3.2. Connection Request Servicing in an $L+1$ Network	6
3.3.3. Fault Tolerance in an $L+1$ Network	9
4. Performance Evaluation	9
4.1. The Simulator	9
4.2. Performance Metrics	11
4.2.1. Blocking Probability	11
4.2.2. Average Path Length and Average Shortest Path Length	11
4.2.3. Effective Utilization	12
4.2.4. Probability of Path Reassignment	13
4.2.5. Link Load	13
4.3. Network Structures	14
4.4. Simulation Parameters	16
5. Simulation Results and Comparison	16
5.1. Blocking Probability	16
5.2. Average Path Length	19
5.3. Effective Utilization	21

5.4. Comparison Between Network Topologies	23
5.5. Probability of Reassignment	24
6. Conclusion	25
6.1. Summary of Results	25
6.2. Future Work	26
6.3. Closing Remarks	27
References	28
Appendix	30
A. Confidence Intervals	30

## LIST OF FIGURES

Figure 1. A 3x3 mesh base network and its $L$ subgraphs.	6
Figure 2. The $L$ subgraphs while attempting to service connection requests.	7
Figure 3. The base network after servicing the connection requests.	8
Figure 4. The subgraphs after servicing the connection requests $A-C$ and $D-C$ .	8
Figure 5. The anatomy of a fault recovery.	9
Figure 6. 14 Node, 23 Link NSFNET	14
Figure 7. 16 Node, 32 Link 4x4 Mesh Torus	15
Figure 8. 23 Node, 26 Link ARPANET-2	15
Figure 9. Blocking Probability vs. Link Load for NSFNET	18
Figure 10. Blocking Probability vs. Link Load for ARPA-2	18
Figure 11. Blocking Probability vs. Link Load for 4x4 Mesh Torus	19
Figure 12. Average Path Length vs. Link Load for NSFNET	20
Figure 13. Average Path Length vs. Link Load for ARPA-2	20
Figure 14. Average Path Length vs. Link Load for 4x4 Mesh Torus	21
Figure 15. Effective Utilization vs. Link Load for NSFNET	22
Figure 16. Effective Utilization vs. Link Load for ARPA-2	22
Figure 17. Effective Utilization vs. Link Load for 4x4 Mesh Torus	23
Figure 18. Blocking Probability vs. Link Load	24
Figure 19. Probability of Reassignment vs. Link Load	25

## LIST OF TABLES

Table 1. NSFNET 95% Confidence Intervals for Average Blocking Probability	30
Table 2. ARPA-2 95% Confidence Intervals for Average Blocking Probability	30
Table 3. 4x4 Mesh Torus 95% Confidence Intervals for Average Blocking Probability	31
Table 4. NSFNET 95% Confidence Intervals for Average Path Length	31
Table 5. ARPA-2 95% Confidence Intervals for Average Path Length	31
Table 6. 4x4 Mesh Torus 95% Confidence Intervals for Average Path Length	32
Table 7. NSFNET 95% Confidence Intervals for Normalized Ave. Effective Utilization	32
Table 8. ARPA-2 95% Confidence Intervals for Normalized Ave. Effective Utilization	32
Table 9. 4x4 Mesh Torus 95% Confidence Intervals for Normalized Ave. Effective Utilization	33
Table 10. NSFNET 95% Confidence Intervals for Ave. Probability of Reassignment	33
Table 11. ARPA-2 95% Confidence Intervals for Ave. Probability of Reassignment	33
Table 12. 4x4 Mesh Torus 95% Confidence Intervals for Ave. Prob. of Reassignment	34



## ABSTRACT

The data transmission potential inherent in optical networks is enormous, and because of this great potential, optical networks are in dire need of fault tolerance. Recently, various fault tolerance techniques exploiting the special properties of optical data transmission have been presented, one of the most interesting being backup-multiplexing. Backup multiplexing comes in a couple of different flavors, but it basically enables connections to be backed up by allocating system resources for fault recovery upon the occurrence of a single fault in a network. Backup-multiplexing attempts to limit the amount of system resources it utilizes up by allowing backup connections to share a particular wavelength, given that their associated primary paths are link disjoint. This way, in the event of a link failure, each primary routed on that link can be assured of finding an available backup connection.

Backup multiplexing is certainly a viable form of fault tolerance, but is there another way of assuring that a network can recover from a link failure, while not tying up valuable system resources in backup connection allocation? The goal of this research is to present an alternative method, known as  $L+1$  fault tolerance, and to compare the performance of that alternative to that of the backup multiplexing strategy. Each network has a set of subgraphs associated with it such that one of the links in the original network is removed. Connections in the newly proposed strategy are accepted if they can be routed in all of the subgraphs. That way, in the event of a link failure, the network state can be restored to that of the corresponding subgraph, where all connections are guaranteed restoration.

## 1. INTRODUCTION

A physical backbone of fiber optic strands allows data to not only travel at the speed of light, but also allows multiple different data streams to travel along a physical strand at the same time. The concurrent transmission of multiple streams of data using the unique properties of fiber optics is called wavelength division multiplexing (WDM) [1]. WDM networks offer users the ability to transport massive amounts of data at high speeds over large distances. The potential of optical networks for data transmission is enormous, but what can optical networks offer in terms of service guarantee?

Fault tolerance is a very important aspect of any computing system. Virtually all computer systems incorporate some form of passive, active or hybrid fault tolerance [2]. Optical networks are no different, as they typically require some permutation of signal regeneration, retiming and reshaping, all in an effort to maintain the integrity of the data that is carried along its constituent lightpaths [3]. Managing the integrity of the signal is very important, but what if a fault serious enough to render an entire link inoperable occurs? Tolerance of a single link failure is the focus of this research.

Catastrophic link failures in optical networks are, in fact, quite common. A variety of factors can lead to link failure including optical fiber, transmitter, receiver, amplifier, router and converter faults. A fiber cut, possibly the result of an errant excavation, has been estimated to occur, on average, once every four days by TEN, a pan-European carrier network [4]. It has been shown in [5] that detection, location and isolation of all of these fault scenarios is both very important and very possible. A link fault can be detected as easily as the receiver node detecting a loss of light on the link, and invoking a network management algorithm to first notify and then recover from the fault without causing network failure.

There have already been several approaches to link fault tolerance laid forth in literature. Three such strategies, presented in [6] and [7], require the usage of network resources to provide backup lightpath routing so that, when a fault occurs, there is an alternate path for the connection to use. Service in such an approach is only interrupted briefly to allow restoration to occur. The major drawback of this approach is the allocation of valuable system resources on typically unused backup lightpaths.

The major design approach taken in the development of the fault tolerance approach presented is the elimination of system bandwidth for the establishment of backup paths.  $L+1$  fault tolerance, as it is referred to in this paper, works by routing connections on network subgraphs. There are  $L$  links in each network. That means that there are  $L$  subgraphs in which one link is missing from the original network configuration. A connection is only serviced if it can be routed on all subgraphs so that, in the event of a link fault, the network is restored to the state given by the corresponding subgraph with the faulty link removed. In this strategy there is no capacity sacrificed to the routing of backup connections.

This paper first overviews, in Section 2, several approaches that have previously been developed to handle link faults in optical networks. Section 3 presents  $L+1$  fault tolerance as a strategy for dealing with link faults in optical networks. To assess the effectiveness of  $L+1$  fault tolerance, network simulation has been conducted according to the guidelines laid forth in Section 4. Simulation results are analyzed and compared to other link fault tolerance strategies in Section 5. This paper concludes with an overall summary in Section 6.

## 2. LITERATURE REVIEW

### 2.1. Backup Multiplexing

Developed in [7], backup multiplexing enlists both proactive and reactive fault tolerance. Backup multiplexing is proactive in that, when a request enters the network, measures are taken to reserve system resources for recovery after a link failure. It is a reactive approach in that it requires the network state to be altered on the fly during connection restoration.

Initially, guaranteeing a connection required that system resources be allocated, and, if a link failure did not occur, those resources went unused. That's where the idea of multiplexing backups comes in. If two primary connections were routed along link-disjoint paths, why couldn't their backup paths have common links and wavelengths? In the event of a link failure, only one of the primary connections would be in need of its backup connection and the other would continue as if nothing happened. Thus, many backup paths can be multiplexed together on a wavelength, provided their associated primary connections are

routed on link-disjoint paths. Multiplexing backup connections reduces the amount of system resources that are effectively unused, except in the event of a link failure.

Backup multiplexing assumes the single-link failure model, and uses a 100% restoration guarantee. It is a path based restoration technique, meaning that the whole connection path changes in the event that link failure occurs and the backup connection is used. The multiplexing algorithm comes in two different styles, primary dependent backup wavelength assignment (PDBWA) and primary independent backup wavelength assignment (PIBWA). PDBWA requires that both the primary and backup connection must use the same wavelength, while PIBWA has no such requirement.

Backup multiplexing has the advantage of offering a 100% connection restoration guarantee upon the occurrence of a link fault. There is also the chance that it could tolerate a second link fault, although there is no guarantee. It also attempts to optimize network resource allocation by allowing backups to share capacity rather than require each backup reserve its own link capacity. Although backup multiplexing does reduce the capacity used for backup connection reservation, it still requires at least some network capacity to be rendered useless if a fault does not occur and the backup connection is unnecessary.

## 2.2. Primary-Backup Multiplexing

An extension of the backup multiplexing technique is primary-backup multiplexing, also presented in [7]. Primary-backup multiplexing allows a primary connection to actually use a light path reserved by a backup connection if necessary. In the event of a primary connection failing and finding that its backup connection has been compromised by another primary connection, the connection fails. Thus there is not 100% connection guarantee as in backup multiplexing. Primary-backup multiplexing works on the premise that three things must happen concurrently for a connection restoration to fail:

- 1) A link fails during normal network operation.
- 2) The failed link is being used by a primary connection.
- 3) A channel on its backup connection is being used by another primary connection.

Primary-backup multiplexing takes advantage of the relatively low probability of all three of the aforementioned events occurring at the same time. It is important to note that primary-backup multiplexing does not offer a 100% guarantee of connection restoration in

the event of a link failure. While the primary-backup algorithm is effective, the lack of a 100% guarantee is a disadvantage. Primary-backup multiplexing does have the advantage of allowing primary connections to use backup connection capacity that would never have been used without a link failure.

### 3. $L+1$ FAULT TOLERANCE ROUTING STRATEGY

#### 3.1. The Key Characteristics of $L+1$ Fault Tolerance

The proposed routing strategy attempts to provide a passive form of redundancy to optical networks in the event of a single-link failure. It is passive in that, before a connection is established, it is subjected to the constraints of  $L+1$  routing and is thus guaranteed in the event of a single link failure. The end user experiences nominal interruption in service due to network state restoration. The key characteristics of the  $L+1$  strategy are as follows:

- 1) No additional system transmission resources are used to provide connection redundancy.
- 2) Fault recovery network states are maintained throughout the operation of the network.
- 3)  $L+1$  fault tolerance provides a 100% guarantee that any single link fault can be recovered from.
- 4)  $L+1$  is a path-based fault tolerance strategy.

The first characteristic highlights one of the most important aspects of the strategy; it doesn't require the allocation of system transmission resources to ensure recoverability after the detection and location of a link fault. Simply put, there is no link capacity lost due to the routing of backup connections because no backup connections exist in this strategy. The second characteristic is important because, upon the occurrence of a fault, the network restores itself to a state that eliminates the defective link from consideration, and the network operates as if it never existed. Third, the strategy provides a 100% guarantee that any single link failure is recoverable. This becomes important when comparing  $L+1$  to other fault tolerant strategies. Fourth,  $L+1$  is a path based recovery strategy because it does not guarantee that any of the same links are used to reroute a connection upon the occurrence of a fault. A disadvantage of  $L+1$  fault tolerance is that it can potentially require complete

reconfiguration of the network. Not all connections may be affected by a network reconfiguration, but no connection is guaranteed to be unaffected by a link fault recovery.

### 3.2. The Key Assumptions of $L+1$ Fault Tolerance

In order for the  $L+1$  strategy to be a viable alternative in optical network fault tolerance, certain assumptions need to be made. The assumptions of the  $L+1$  fault tolerance strategy are as follows:

- 1) A single link failure scenario is assumed.
- 2) Each node in the network has comprehensive knowledge of the status of the network.
- 3) There is adequate storage capacity at each node to store network subgraph state information.
- 4) The occurrence of a link fault can be detected, located and isolated.

First off,  $L+1$  fault tolerance assumes that the probability of suffering a double link failure during network operation is very low. The strategy guarantees the recovery of the network from any single link failure at any given time. Second, it is assumed that each node knows the entire network state at any given time. This is key because all nodes need to know when and where a fault has occurred so that they can appropriately retask themselves to adopt the backup network state. The ability to conform to a backup network state goes hand in hand with the third assumption, that each node has adequate storage space to maintain all subgraph network state information. Lastly, it is assumed that any link fault that occurs in the network will be detected, located and isolated, and that all nodes receive this information.

### 3.3. What is $L+1$ Fault Tolerance?

#### 3.3.1. A Mathematical Description

Networks consist of a set of nodes and links that correspond to the various servers, routers, switches and cables that make up its physical implementation. These nodes and links can be viewed as a set of vertices and edges in a graph. Each graph,  $G$ , is defined as a set of  $V$  vertices and  $E$  edges or, in mathematical terms,  $G = (V, E)$ . That being said, there exists a set of subgraphs of  $G$ , denoted as  $G_i$ , where  $e_i$  is removed from the graph, or mathematically,

$\exists \forall i \leq L G_i \mid G_i = G - \{e_i\}$ , where  $L$  is the cardinality,  $|E|$ , of the set of edges in graph  $G$ .

Therefore there exist  $L$  subgraphs of graph  $G$ , each one missing link  $e_i$ .

The set of  $L$  subgraphs of  $G$  represents all possible single-link failures in the network. The original full link graph is called the *base network*. The base network's constituent subgraphs are not considered networks because they only maintain a state of the base network, and are not real networks until a link fault occurs. For example, let a standard 3x3 mesh be the base network. Such a base network contains 9 vertices and 12 edges. Therefore, there are 12 subgraphs of the base network, as shown in Figure 1. For the purposes of this example, assume that each edge in the base network (and its constituent subgraphs) has a capacity of one and that the distance between any pair of adjacent vertices is one.

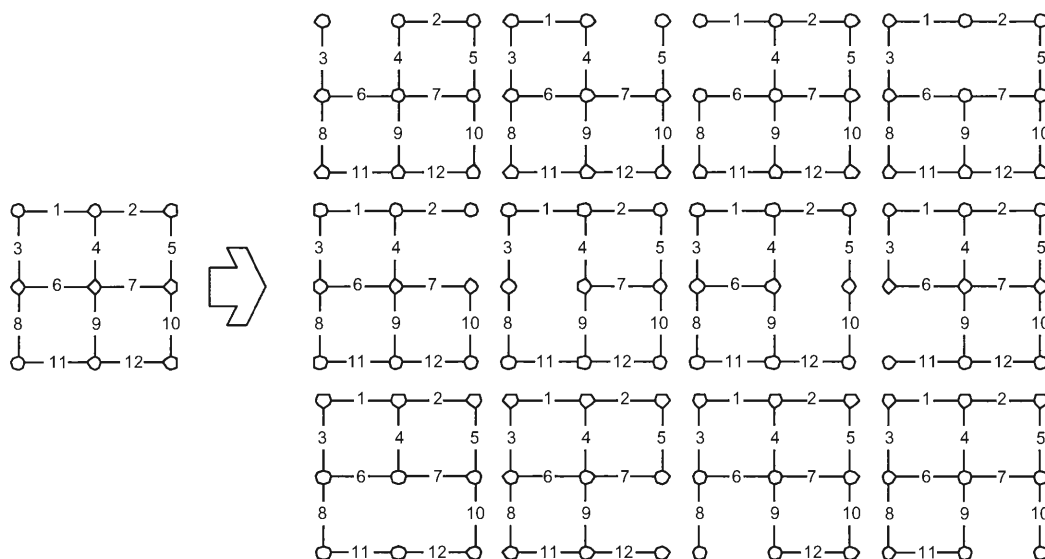


Figure 1. A 3x3 mesh base network and its  $L$  subgraphs.

### 3.3.2. Connection Request Servicing in an $L+I$ Network

Let there be a request issued by vertex  $A$  to connect with vertex  $C$ . This connection attempts to find a path from  $A$  to  $C$  on all of the  $L$  subgraphs of the base network. Figure 2 depicts the connection from vertex  $A$  to vertex  $C$  as the black line. Note that in all subgraphs the connection can be routed successfully, resulting in a connection between  $A$  and  $C$  in the base network, as shown in Figure 3.

Additionally, let there be a request issued by vertex  $D$  to connect with vertex  $C$ . An attempt is made to find a viable path on all  $L$  subgraphs, but this attempt fails, as there is no possible path between vertices  $D$  and  $C$  in subgraphs  $G_2$  and  $G_5$ . Figure 2 shows the attempted routing of the connection between vertices  $D$  and  $C$  as the lighter colored line. Notice the cases of  $G_2$  and  $G_5$ , where the connection between nodes  $D$  and  $C$  fails because there is no free path to go from  $D$  to  $C$ . In  $G_2$ , the connection is blocked by an already full edge 5, and in  $G_5$  it is blocked by an already full edge 2.

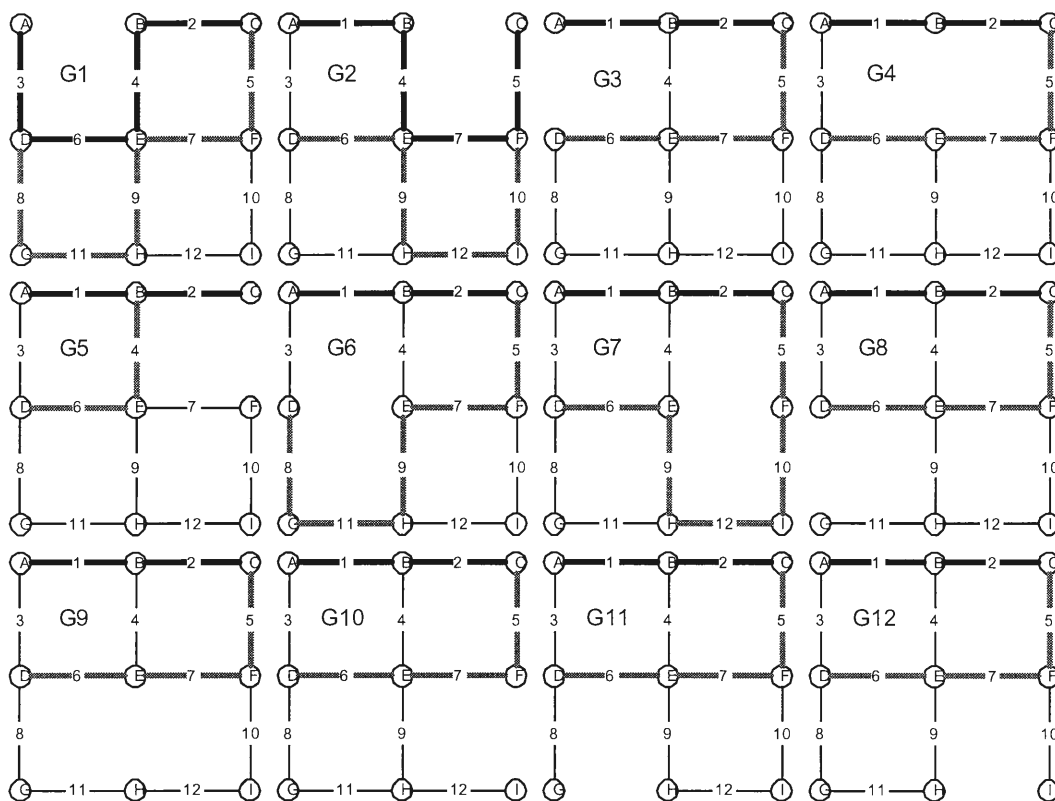


Figure 2. The  $L$  subgraphs while attempting to service connection requests.

The connection between vertices  $D$  and  $C$  is not accepted, and consequently not routed in the base network, as shown in Figure 3, nor the  $L$  subgraphs of  $G$ , as shown in Figure 4. After the attempted routing of two requests, only the connection between vertices  $A$  and  $C$  is accepted and routed on the base network as well as the  $L$  subgraphs. No other edges are consumed through backup connection routing and all subgraph network state information is saved at each node to assist in a fault recovery situation. Additionally, the



shortest path between vertices  $A$  and  $C$  is chosen as the connection path in the base network, although this may not always be the case.

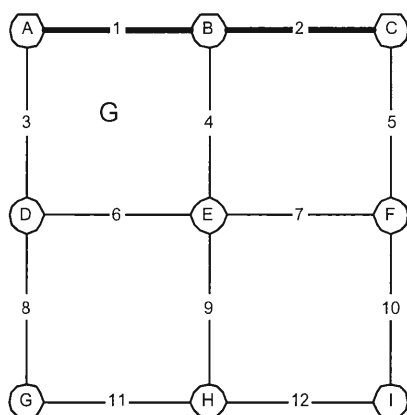


Figure 3. The base network after servicing the connection requests.

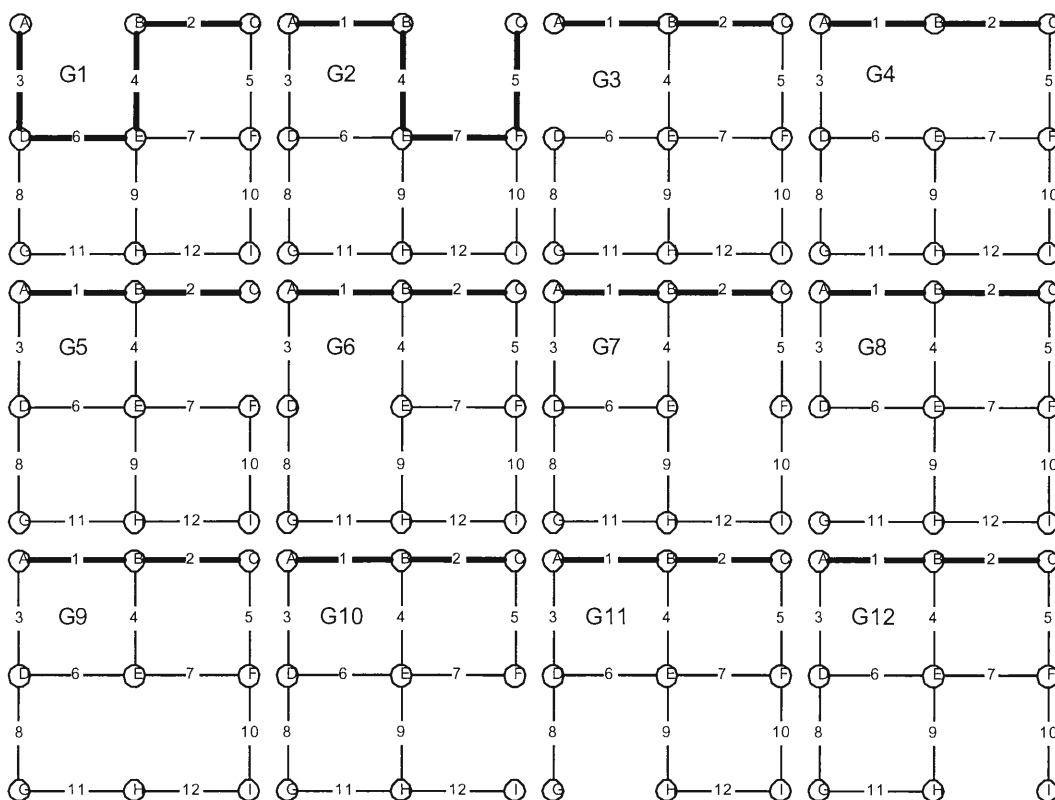


Figure 4. The subgraphs after servicing the connection requests  $A-C$  and  $D-C$ .

### 3.3.3. Fault Tolerance in an $L+1$ Network

In the event of a fault, the network can fully recover by accepting the subgraph network state corresponding to the located edge failure. For example, assume that there is an arbitrary failure in edge 2. For whatever reason, the edge is left inoperable. To recover, the network reroutes all current connections to reflect the network state depicted by subgraph  $G_2$ . The fault occurrence and recovery cycle is illustrated in Figure 5.

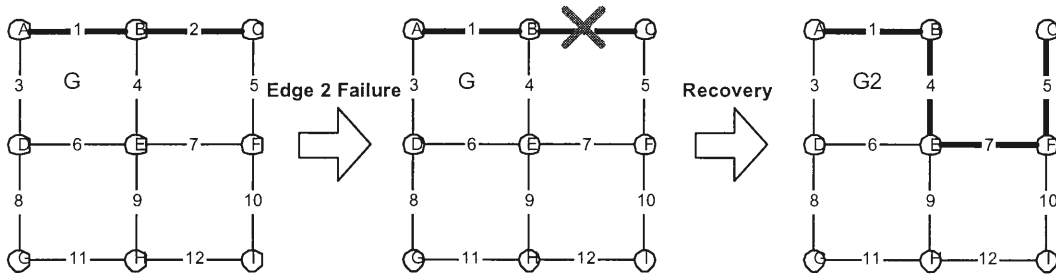


Figure 5. The anatomy of a fault recovery.

While the above examples using the 3x3 mesh base network are, to say the least, basic, they illustrate the key points of this particular fault tolerant connection routing strategy stated in section 3.1. Namely, no additional network transmission resources are sacrificed to form backup connections, and all network state information is saved to assist in fault recovery.

## 4. PERFORMANCE EVALUATION

### 4.1. The Simulator

The performance evaluation is based on simulation results obtained by using ISTOS, the Iowa State Optical Simulator, developed in the Dependable Computing and Networking Laboratory (DCNL), based on [8], and written in C++. ISOTS is capable of simulating heterogeneous network topologies using a wide array of primary connection routing strategies. The scope of the simulation is limited to homogeneous network topologies where nodes are capable of fiber and time slot conversion.

The shortest path length in terms of hops routing strategy has been utilized to evaluate the effectiveness of  $L+1$  fault tolerance. Shortest path length routing attempts to dynamically route connections along the path with the least number of links between source

and destination nodes. Each link is known as a hop. Wavelength assignment within a link is done at random. Connections are routed dynamically in that each request is routed based on the network state at the time it enters the network, as defined in [9]. This differs from fixed path and fixed alternate path routing, as presented in [10], in which the path between a pair of nodes is predetermined. Dynamic routing and wavelength assignment typically performs better than fixed path routing, although it requires higher control overhead because each node must maintain network state information.

The no backup and backup multiplexing routing strategies will also be simulated to provide a reference to measure the effectiveness of  $L+1$  fault tolerance. No backup routing will use the shortest path in terms of hop strategy, while backup multiplexing will use shortest cycle routing. The results laid forth in [7] for backup multiplexing are based on the selection of the shortest path for the primary connection. After a shortest length primary path is chosen, the links on the primary path are then removed from the network and then the shortest path on the new network becomes the backup path. Shortest cycle routing is used to increase the performance of backup multiplexing by guaranteeing both a potential primary and backup path are found while attempting to establish a connection and that the primary/backup path pair is the shortest pair of paths from source to destination.

Connection requests are created using a library of random number generating functions. The following request parameters use a negative exponential probability distribution:

- 1) Arrival Rate – the user defines the expected arrival rate.
- 2) Hold Time – an expected hold time of 1.0 is used.

The arrival rate parameter is varied to control the load on the network for each simulation. The expected hold time is set to 1.0 because there is no sense of how long an interval of time is within the simulator. One unit of hold time could be 1 nanosecond, 1 second or 1 hour; whatever the user deems it to be.

A uniform probability distribution is used to generate random numbers for the following connection request parameters:

- 1) Source Node
- 2) Destination Node

## 4.2. Performance Metrics

Several metrics, described in [11], are used to evaluate the effectiveness of the  $L+I$  *fault tolerance* scheme. These metrics are designed to measure the both the efficiency and the feasibility of such a scheme and are only measured in the base network. They are be described in further detail in sections 4.2.1 through 4.2.5 and are as follows:

- 1) Blocking probability
- 2) Average path length
- 3) Average shortest path length
- 4) Effective used network capacity
- 5) Probability of path reassignment
- 6) Link load

### 4.2.1. Blocking Probability

Blocking probability is the most common indicator used to assess network routing and fault tolerance strategies. It is the probability that a request entering the network is rejected. Blocking probability is defined in Equation [1], where  $B$  is defined as the total number of blocked requests and  $R$  is the total number of requests.

$$P(\text{Connection Blocked}) = \frac{B}{R}$$

[1]

### 4.2.2. Average Path Length and Average Shortest Path Length

Average path length and average shortest path length are used in tandem to compare how metrics perform within a network. As the ultimate goal in routing a connection is usually to use the shortest path between two points, average shortest path length provides a way to compare how effectively a routing strategy performs in a given network configuration. Both average path length and average shortest path length are calculated in terms of number of “hops,” or number of links along the path using Equations [2] and [3], where  $L_i$  is the length of the path of request  $i$ , and  $R$  and  $B$  are the same as in Equation [1].

$$P_{ave} = \frac{\sum_{i=0}^{R-B} L_i}{R-B}$$

[2]

$$SP_{ave} = \frac{\sum_{i=0}^{R-B} SPL_i}{R-B}$$

[3]

### 4.2.3. Effective Utilization

Network utilization metrics are characterized by their inclusion of the ideas of connection and link capacity. They are an indication of how much of the network is being used over the course of operation and whether there are enough resources available to handle the request load demands

Effective utilization (Equation [4]) refers to the minimum amount of system resources needed to service all accepted connections if they were to have been routed along the shortest path.  $BW_i$  is the bandwidth of accepted connection request  $i$ ,  $R$  and  $B$  are as in Equation [1],  $L_i$  is as in Equation [2], and  $SPL_i$  is as in Equation [3].

$$U = \sum_{i=0}^{R-B} BW_i * SPL_i$$

[4]

In order for the effective utilization metric to be useful, it first has to be normalized. The first step is to normalize it to the time duration of the simulation so that data obtained at different arrival rates can be compared. Dividing by the time duration of the simulation normalizes utilization. The simulation time is known only to the network, and can either be attained by knowing the time that the last request enters the network, or by calculating it as a function of  $\frac{R}{\lambda}$ , where  $R$  is as in Equation [1] and  $\lambda$  is the arrival rate of requests into the network.

Normalizing the utilization with respect to time yields a value that is bounded on the low side by 0 and on the high side by the total available capacity in the network. The second step to normalization is to normalize it by dividing it by the total available capacity of the network, given by  $L*C$ , where  $L$  is the total number of links in the network,  $C$  is the total

available capacity per link. Mathematically this is denoted as  $0 \leq \frac{\lambda}{R} \cdot U \leq L \cdot C$ , where  $U$  is the utilization as calculated by Equation [4]. Dividing through by  $L \cdot C$  yields Equation [5].

$$0 \leq \frac{\lambda \cdot U}{R \cdot L \cdot C} \leq 1$$

[5]

#### 4.2.4. Probability of Path Reassignment

$L+1$  fault tolerance depends on the network state's ability to be changed to that of a subgraph during a recovery. This potentially requires all connections in the network to be reassigned to different paths. In order to quantify the amount of path reassignment taking place, path reassignment probability has been measured. Upon the occurrence of a link failure, it is the probability that a connection's path on the base network will have to change when the subgraph state corresponding to the link failure is incorporated. This does not account for the possibility that the connection will have to change its wavelength or timeslot. Probability of path reassignment is calculated using Equations [6] and [7], where  $R$  and  $B$  are the same as in Equation [1] and  $P_j(R_i)$  is 1 if the base network path of request  $R_i$  is the same as the path in subgraph  $j$ , and 0 if it isn't.  $L_{phys}$  is the number of links physically routed together. Equation [7] calculates the probability of reassignment of a backup multiplexed network and is expressed as the probability that a particular path contains the faulty link, thus necessitating the use of a backup connection.

$$P(L + 1 \text{ Path Reassigned}) = 1 - \frac{\sum_{i=0}^{R-B} \sum_{j=0}^L P_j(R_i)}{L_{phys} \cdot (R - B)}$$

[6]

$$P(\text{Backup Multiplexed Path Reassigned}) = \frac{\sum_{i=0}^{R-B} L_i}{L_{phys} \cdot (R - B)}$$

[7]

#### 4.2.5. Link Load

Link load is a measure of the load placed on each node in the network at any given time. It is useful in providing a baseline for the comparison of the effectiveness of routing

strategies across different network topologies. Link load, or  $\gamma$  is calculated using Equation [8], where  $L_{total}$  is the total number of links in the network where each duplex link is treated as 2 links,  $N$  is the total number of nodes in the network and  $\lambda_n$  is the arrival rate per node.  $\bar{H}$  is the expected length of a primary connection in the topology in hops calculated by simulating the topology at very low arrival rate, very high link capacity and very low request hold time. Link load is expressed in units of Erlangs and will be used to compare results between networks.

$$\gamma = \frac{N \cdot \lambda_n \cdot \bar{H}}{L_{total}}$$

[8]

### 4.3. Network Structures

Three standard network structures are used to assess and compare the performance of the  $L+I$  strategy. The simulated networks are shown in Figure 6, Figure 7 and Figure 8.

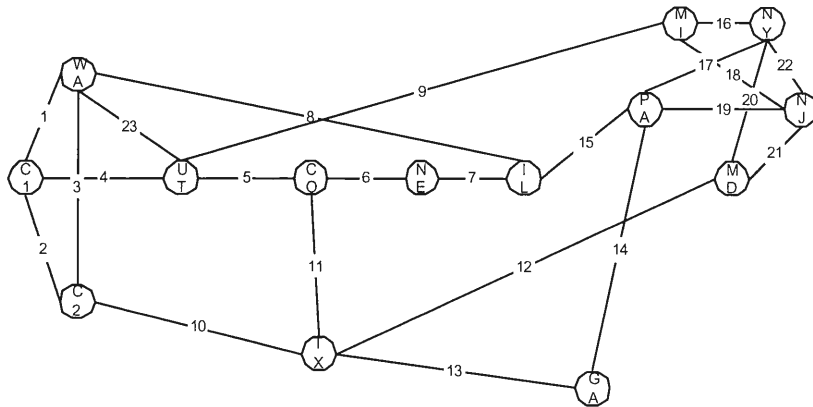


Figure 6. 14 Node, 23 Link NSFNET

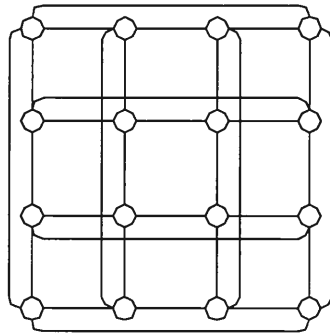


Figure 7. 16 Node, 32 Link 4x4 Mesh Torus

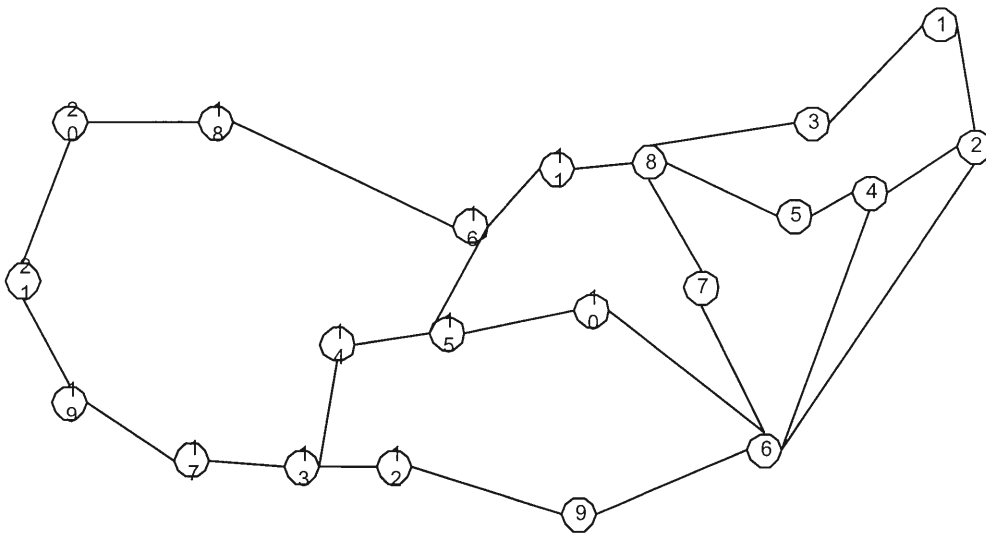


Figure 8. 23 Node, 26 Link ARPANET-2

The NSFNET and APRA-2 networks were simulated to show how the  $L+I$  strategy performs in real world topologies. The 4x4 mesh torus network was chosen because it possesses a high level of connectivity. Each node in a 4x4 mesh torus has a degree of 4, resulting in many potential paths between a node pair. This is especially important in  $L+I$  *fault tolerance* because of the nature of removing links to derive subgraphs. In 2 of the  $L$  subgraphs, a node with degree of 2 becomes a node with degree of 1 as exhibited by node  $GA$  depicted in Figure 6.

All nodes in each network have fiber and time slot conversion capabilities, although the number of fibers per link and timeslots per wavelength are set to 1. The number of timeslots used is unimportant as it only adds additional capacity to a link. If a fault were to



occur, the entire wavelength and all the timeslots on the wavelength would be reassigned as a whole to reflect the new network state. The number of wavelengths per fiber is simulated at 16. In general, the capacity of a link is a direct function of the number of wavelengths on each fiber in the link.

The wavelength continuity constraint has been applied to every request made throughout all simulations. No node used has wavelength conversion capability, and thus the same wavelength must be used along the entire length of a connection path.

In the event that a network incorporates a duplex link (as all of the tested networks do), the link is simulated as two simplex links operating in opposite directions. Each simplex link will have the capacity 16 as previously stated. It is assumed in the real world that the two simplex links that comprise a duplex link are physically routed together. Therefore, in the event of a link failure such as a fiber optic cable being severed, both simplex links are severed. The simulation therefore only maintains one subgraph for each duplex link.

#### 4.4. Simulation Parameters

Simulation data will be collected by simulating each network topology for 11 rounds. The first round begins with an unloaded network topology. The following 10 rounds begin with an already loaded network state. Data is taken according to the performance metrics put forth in Section 4.2 at the end of every round. The results of the first round are not used and data analysis occurs on the measurements taken at the end of rounds 2 through 11.

## 5. SIMULATION RESULTS AND COMPARISON

### 5.1. Blocking Probability

Blocking probability was evaluated using Equation [1]. It gives an indication of how well the network and routing strategy accommodate the needs of the network users. In this case, loads that lead to blocking probabilities greater than 0.1 are not used, as anything higher than 0.1 would render the network useless to its users. As the performance of  $L+1$  fault tolerance is at issue, the interval is established by ascertaining what value of the link load yields a blocking probability of approximately 0.1 in the  $L+1$  scheme. The following figures all concur that  $L+1$  fault tolerance performs much better than backup multiplexing under the

same parameters. In the best case, the NSFNET, the blocking probability of backup multiplexing is roughly 3.5x that of the  $L+1$  strategy, and is approximately 3x higher than  $L+1$  in the ARPA-2 and mesh torus topologies.

The results for  $L+1$  fault tolerance and backup multiplexing and no backup strategy for the NSFNET, ARPA-2 and 4x4 mesh torus topologies are shown in Figure 9, Figure 10 and Figure 11, respectively. These figures show that  $L+1$  fault tolerance performs reasonably well in terms of blocking probability, and exceptionally well in comparison to the backup multiplexing strategy.  $L+1$  fault tolerance outperformed backup multiplexing throughout the entire arrival rate interval for all topologies.

One factor in the increasing the blocking probability of  $L+1$  fault tolerance is the presence of nodes with degree of 2 present two of the topologies. For example, there are 2 nodes of degree two in the NSFNET topology, and when subgraphs are formed, these two nodes become nodes of degree 1 in four of the 23 subgraphs. These isolated nodes are much more difficult to route connections to because of the severely limited capacity in and out of the nodes. A node with degree of 2 in the base network is referred to as a *dead-end node*. Dead-end nodes account for 14 of the total 21 nodes in the ARPA-2 topology, and consequently the performance of  $L+1$  in that topology is slightly worse when compared to the NSFNET and mesh torus.

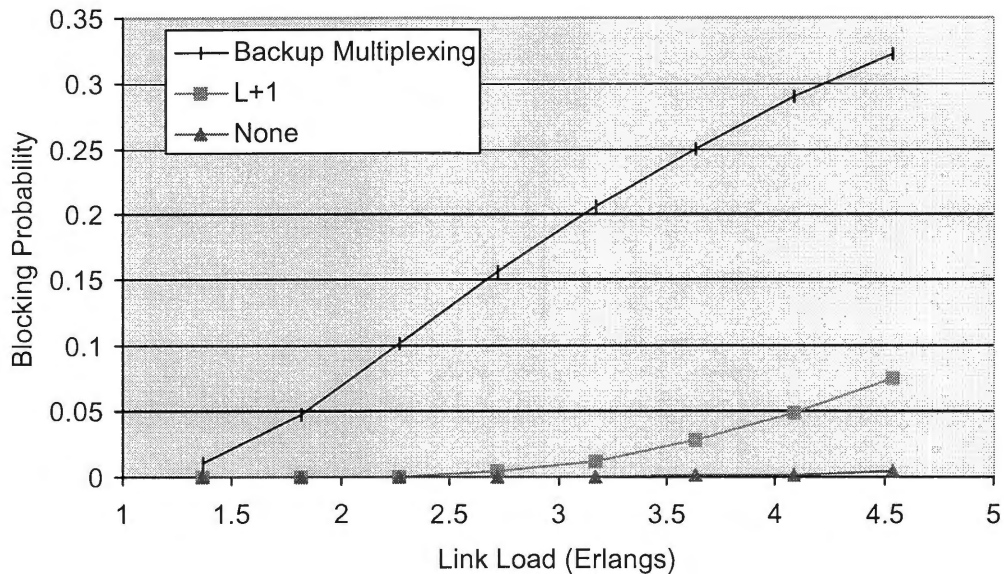


Figure 9. Blocking Probability vs. Link Load for NSFNET

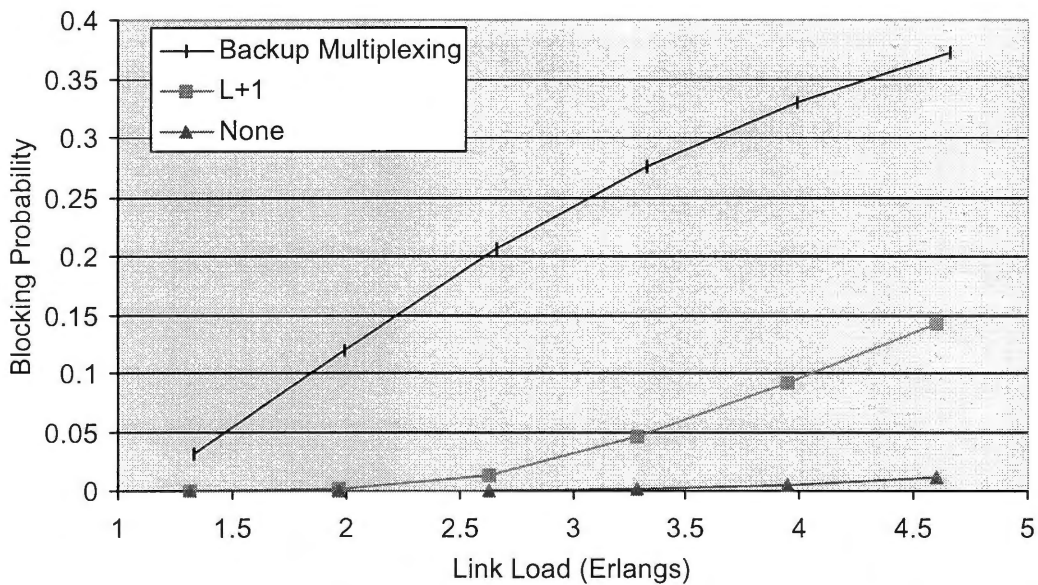


Figure 10. Blocking Probability vs. Link Load for ARPA-2

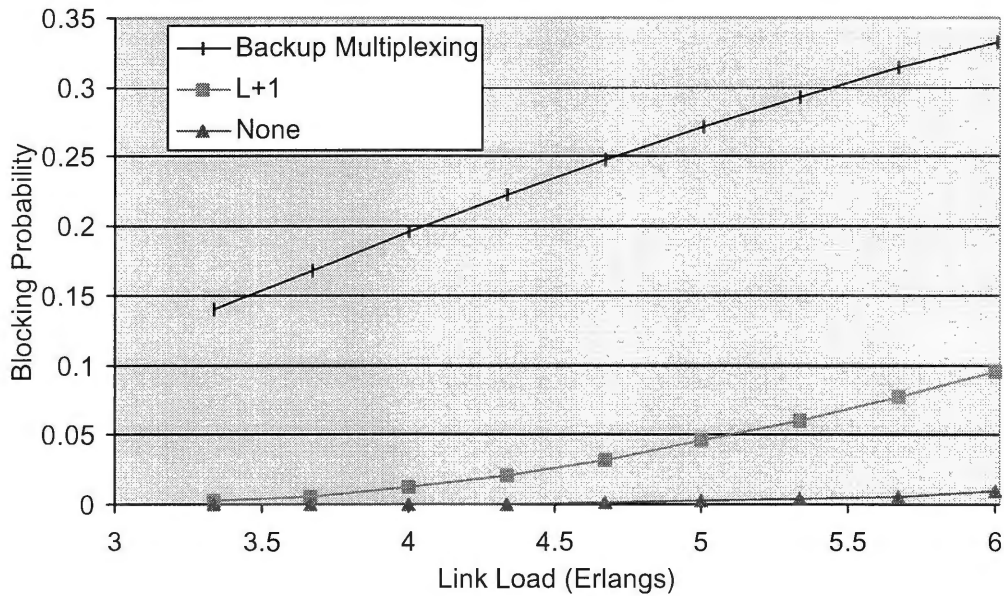


Figure 11. Blocking Probability vs. Link Load for 4x4 Mesh Torus

## 5.2. Average Path Length

Average path length is evaluated according to Equation [2] and indicates the average number of hops a connection must contain. The link load interval of each figure remains the same as the interval for blocking probability used for each respective topology. In general the average path length decreased as link load increased. As more requests entered the network at a time, the network becomes more congested and more requests are blocked. The result is that the requests that are accepted as connections are able to find shorter paths to route on.

The backup multiplexing path lengths are higher across all topologies because the paths are based on shortest cycle routing, and the primary paths are not necessarily the shortest path between two nodes. Shortest cycle routing actually improves performance because, although primary path lengths are longer, a primary-backup pair is almost always found and the total length of the primary-backup pair is the shortest possible.

The results for  $L+1$  fault tolerance and backup multiplexing and no backup strategy for the NSFNET, ARPA-2 and 4x4 mesh torus topologies are shown in Figure 12, Figure 13 and Figure 14, respectively. These figures show that average path length decreases as link

load increases. This large decrease in path length can be attributed to higher blocking probability and the consequent lower number of connections in the network for a request to have to route around. The main purpose of measuring average path length is to illustrate how both fault tolerance strategies compare to using no fault tolerance at all.

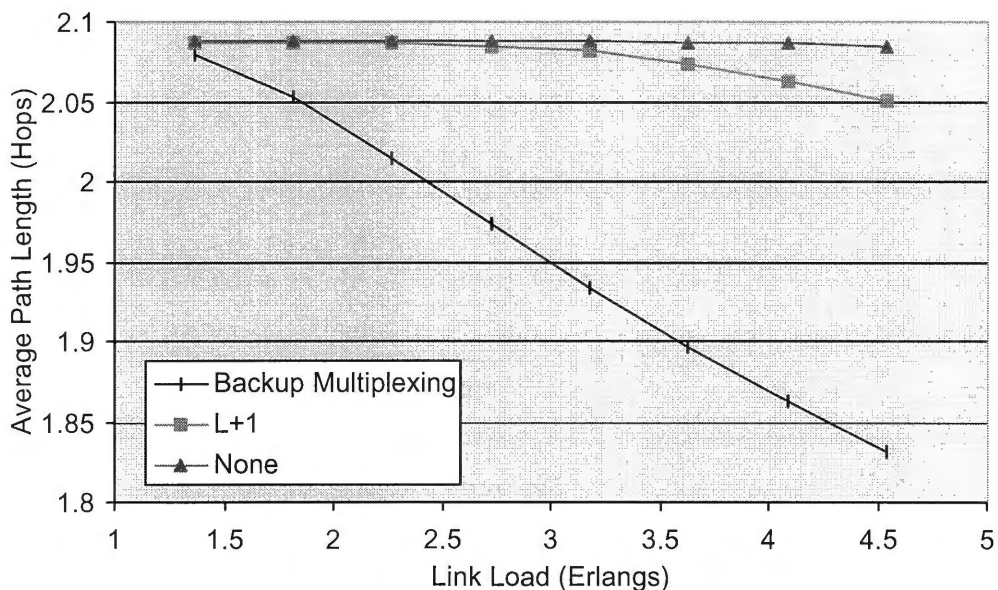


Figure 12. Average Path Length vs. Link Load for NSFNET

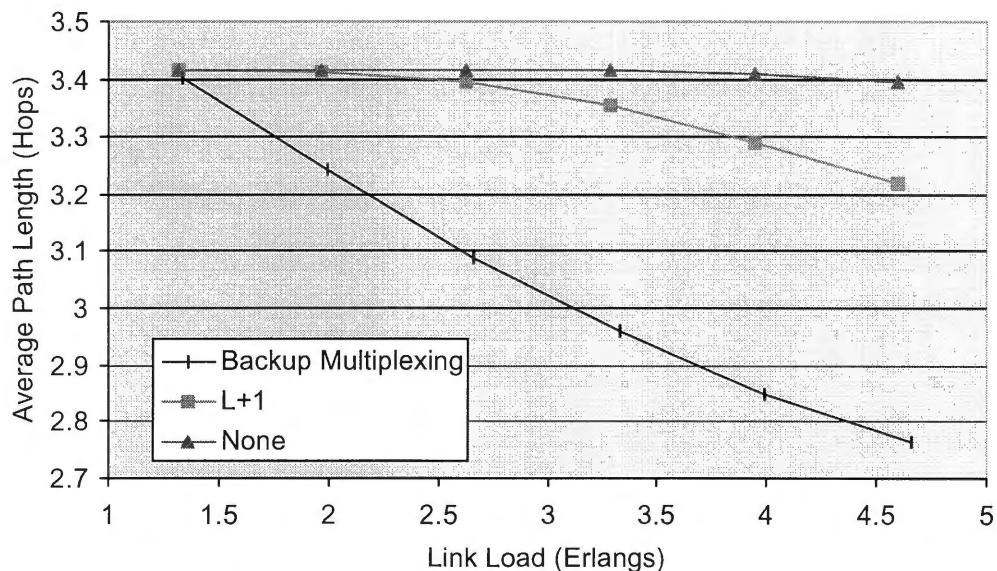


Figure 13. Average Path Length vs. Link Load for ARPA-2

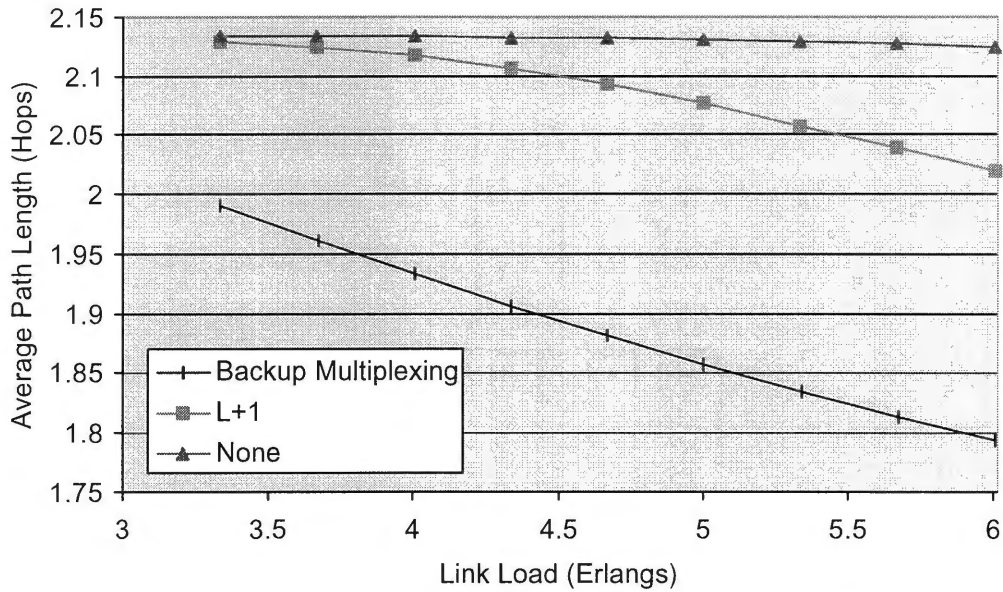


Figure 14. Average Path Length vs. Link Load for 4x4 Mesh Torus

### 5.3. Effective Utilization

Effective utilization measures the minimum amount of network resources needed to accept the connections in the network at any given time for any given link load. In other words, in order to accept the requests that the network did, the network had to provide a minimum amount of resources to the establishment of the connections. This minimum amount of resources is calculated using Equation [4] and normalized to simulation duration and total network capacity as shown in Equation [5]. The effective utilization figures shown below are taken over the same arrival rate interval of the respective topologies' blocking probability figures.

Effective utilization for the NSFNET, ARPA-2 and 4x4 mesh torus topologies are shown in Figure 15, Figure 16 and Figure 17, respectively. For the most part, the effective utilizations of each strategy mirror each other, the exception being under high link loads. The difference is again attributed to fewer requests being accepted as link load increases. Utilization is directly proportional to the sum of the products of the capacity and path length of each accepted connection, and it consequently decreases as fewer requests are accepted.

Again effective utilization is used primarily to compare the performance of both fault tolerance strategies to a no fault tolerance strategy.

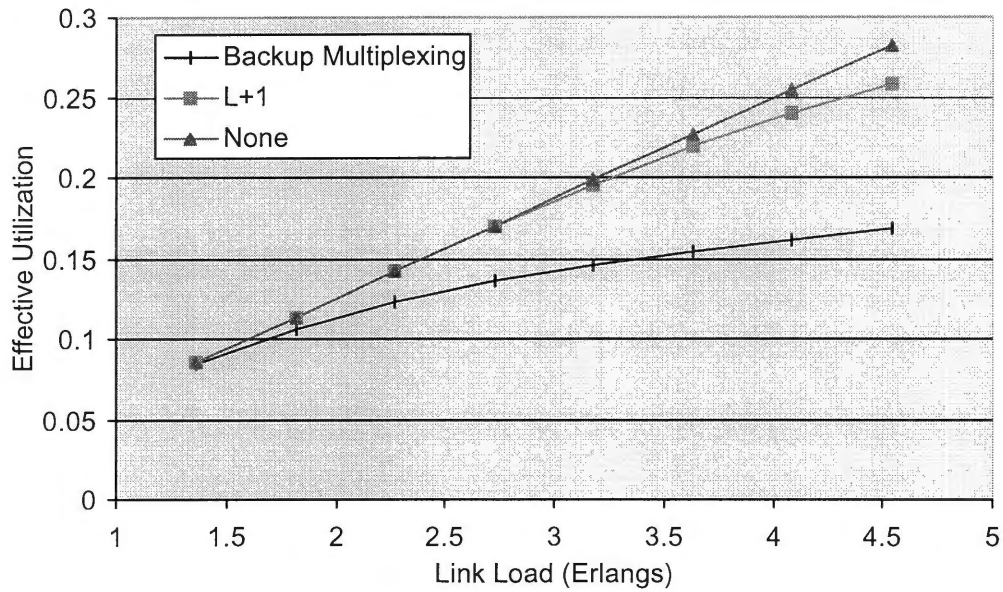


Figure 15. Effective Utilization vs. Link Load for NSFNET

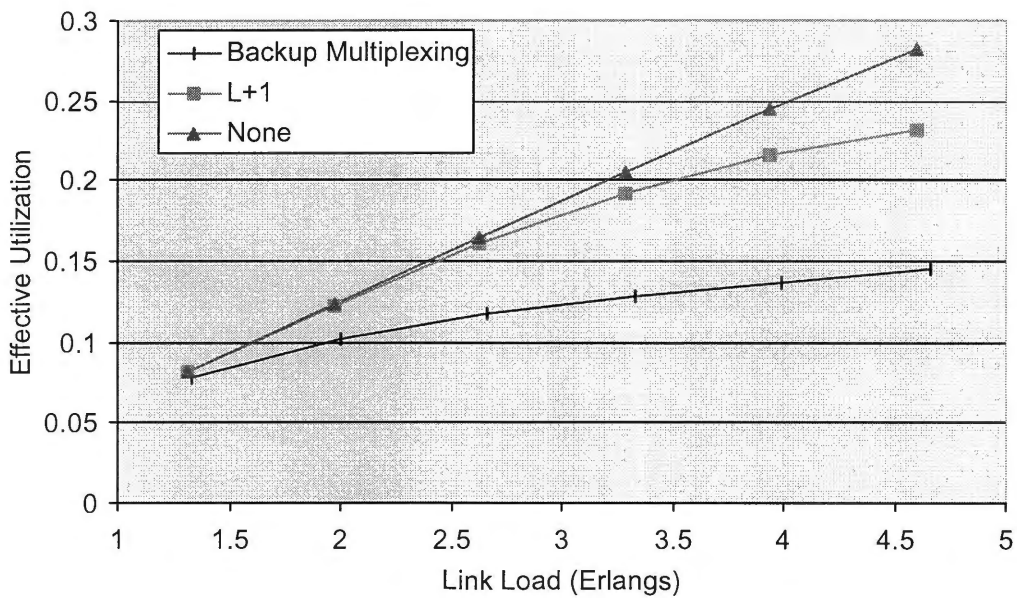


Figure 16. Effective Utilization vs. Link Load for ARPA-2

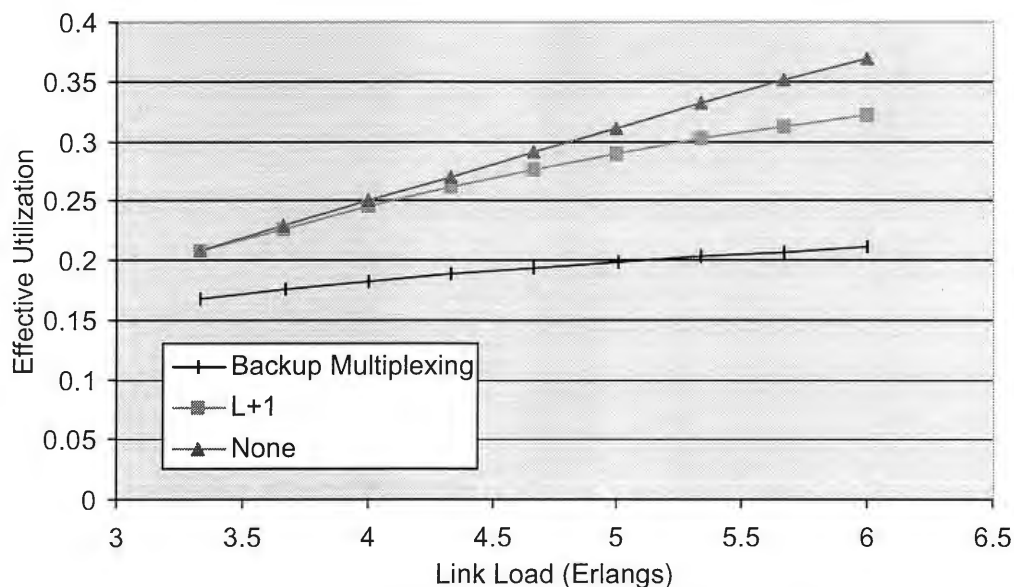


Figure 17. Effective Utilization vs. Link Load for 4x4 Mesh Torus

#### 5.4. Comparison Between Network Topologies

An indication of the connectivity of a topology is the number of dead-end nodes each of the subgraphs of that topology contains. Along that reasoning, the ARPA-2 topology has the worst connectivity with 14 dead-end nodes out of 21, the NSFNET next with 2 of 14 and the mesh torus best with no dead-end nodes. The blocking probabilities confirm this, as the blocking probability is higher per Erlang of link load for the ARPA-2 topology, followed by the NSFNET and lastly by the mesh torus. The comparatively lower blocking probability per link load of the mesh torus indicates that  $L+1$  fault tolerance performs much better in topologies with higher connectivity.



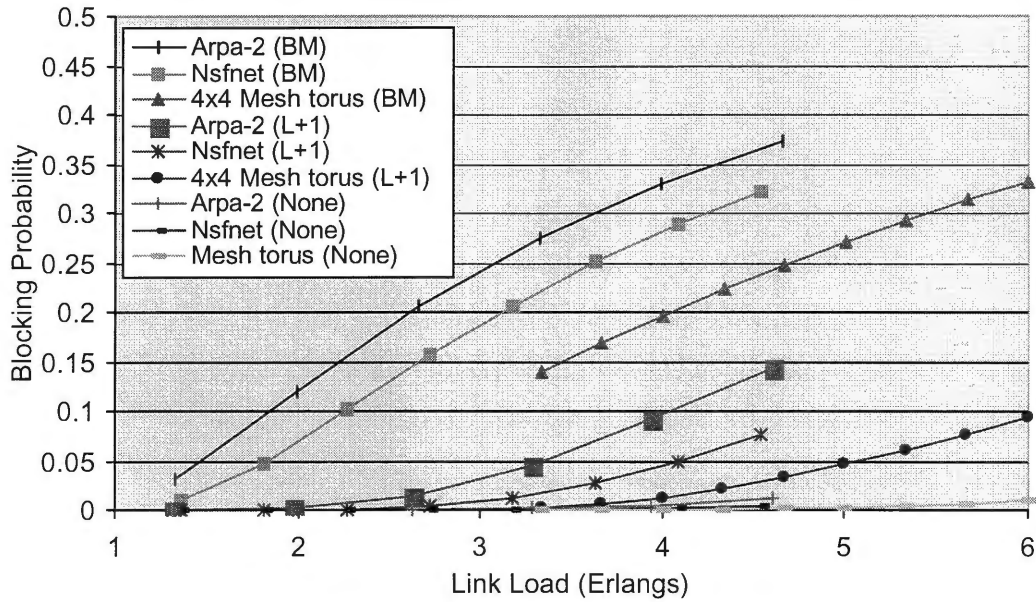


Figure 18. Blocking Probability vs. Link Load

### 5.5. Probability of Reassignment

In a network recovery situation,  $L+1$  fault tolerance requires the network to reconfigure to take the state given by the subgraph corresponding to the link failure. This could potentially require all connections in the network to change how they are routed. The probability of reassignment, given by [6], indicates how likely a connection's path will change during recovery. The probability of reassignment does not apply to the possibility that a connection will have to be routed on a different wavelength, only that it will have to change its logical path along links.

In all three simulated topologies, as the link load increases, the probability of reassignment decreases. This observed decrease is due to the higher probability of a request being blocked as link load increases. As requests enter the network at a higher rate, fewer connections are established. This makes routing the connections that do get accepted on each subgraph easier. Thus there is a much greater probability that a subgraph routes the connection exactly the same as the base network does.

Figure 19 shows the probability of reassignment for the NSFNET, ARPA-2 and 4x4 mesh topologies, respectively. As mass connection reassignment is unique to  $L+1$  fault

tolerance, it is important to show how much more reassignment it requires than backup multiplexing. As the link load increases, the probability of reassignment decreases, indicating that there is less of a chance of a connection having to be rerouted during a network recovery.

The probability of reassignment for the ARPA-2 and NSFNET topologies remains fairly constant and the probability of reassignment for the mesh torus topology decreases slightly more as the link load increases. The mesh torus also has a much higher probability of reassignment overall. This is probably due to the higher connectivity of the mesh torus. More links means that there are more options to route a path on, and the shortest hop length routing metric uses this to the fullest. Backup multiplexing requires far less reassignment in the event of a fault occurrence.

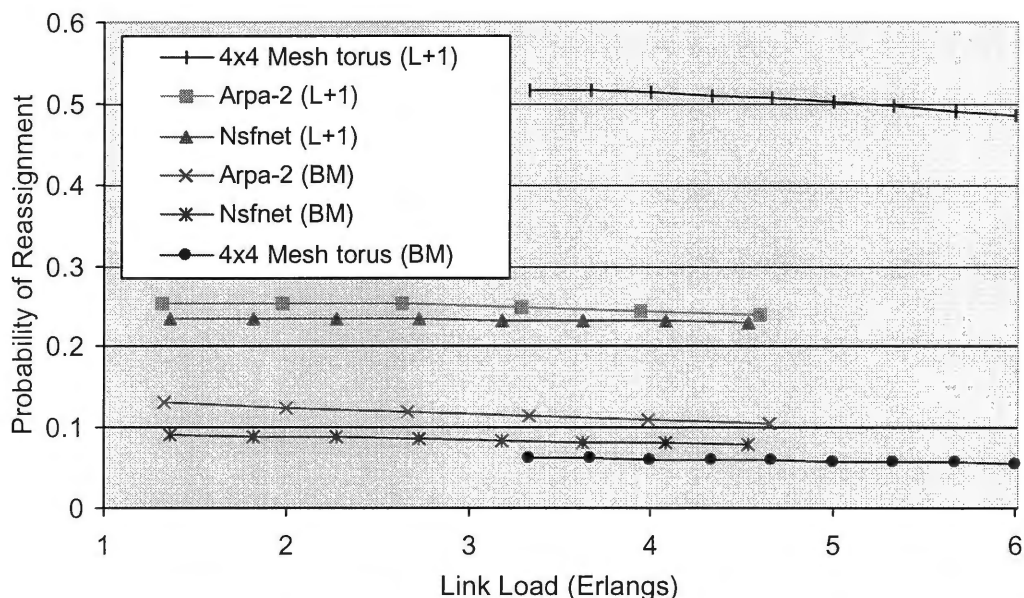


Figure 19. Probability of Reassignment vs. Link Load

## 6. CONCLUSION

### 6.1. Summary of Results

Fault tolerance in optical networks has become increasingly important as reliance on these networks has increased. The ability to provide guaranteed connections in the event of a link failure without adversely affecting the operation of the network is imperative. Backup multiplexing is a strategy designed to allocate capacity on a network to enable a 100%

recovery from a single-link failure. The major drawback of this strategy is that the capacity allocated to backup connections remains unused if there is no link failure.

In this paper, we presented an alternate technique for recovering from single-link failures known as *L+1 fault tolerance*. *L+1 fault tolerance* has the advantage of not requiring a network to allocate capacity for backup connections. As has been shown, the performance of the *L+1* strategy was very good in comparison to that of backup multiplexing. This being said, there are still several ways to possibly improve the performance of the strategy.

## 6.2. Future Work

There are two major areas that will be studied further with respect to *L+1* fault tolerance. The first is how the performance of *L+1* fault tolerance can be improved and the second is the feasibility of such a strategy as a real world implementation.

There are several ways to improve the performance of *L+1* fault tolerance that are to be inspected. *L+1* is flexible in that it is immediately portable to other network strategies. One way to exploit the flexibility of the *L+1* strategy is to assess the operation of different routing strategies such as those published in [12] and further explained in [13]. Possible algorithms include shortest widest path, shortest maximum path, maximum shortest path or any other algorithm designed to more efficiently manage capacity in networks. The goal is to find a routing strategy that is best tailored to operating in the confines of *L+1 fault tolerance*. *L+1 fault tolerance* has the advantage of easily incorporating any desired routing metric.

A second way to possibly improve the performance of *L+1* fault tolerance is to assess how different routing algorithms can be used in conjunction with one another to achieve connection establishment. For example, shortest hop routing, which is used exclusively for this research, can be used to route connections on the base network, while several other metrics such as shortest maximum path can be used to better manage capacity while routing connections on the subgraphs. A subgraph with a dead end node may require the use of a metric that better allocates link capacity, while shortest path length routing may be used on other subgraphs. The routing strategy used on one subgraph can be completely independent of the strategy used on another subgraph.

A third way to improve  $L+1$  fault tolerance is to test it using different network topologies. This includes performance testing of heterogeneous topologies, or networks whose nodes and links each have different operation characteristics. The nodes in these networks can have any combination of fiber, band, wavelength or timeslot conversion capabilities as well as higher link capacities such as 32 available wavelengths. A heterogeneous network can be tailored to take advantage of  $L+1$  fault tolerance. This is done, for example, by limiting the effect of dead end nodes through the allocation of additional capacity or conversion capability on the links and nodes adjacent dead-end nodes.

Lastly, extending  $L+1$  fault tolerance to multiple-link failure models also needs to be investigated. While multiple link failures are much less probable than a single link failure, it is not unheard of. A double link failure is much more probable when the mean time between failures is close to the mean time to repair. A double link failure can also occur if two logically distinct links are physically routed together. With more complex networks incorporating more links becoming more prevalent, addressing multiple link failures needs to be done.

Implementation of  $L+1$  fault tolerance is as important as the how it works. Many times the technological and financial constraints of a design can dictate its feasibility as a real world solution. For example, there must be adequate storage capacity at each node of a network to maintain subgraph state information. Each node in the network must also know the current status of the network and be notified if that status changes. The ability of each node to handle the reassignment of connections both by rerouting and retuning must also to be assessed. A technique is only useful if it can be physically implemented and operate successfully in the real world.

### 6.3. Closing Remarks

$L+1$  fault tolerance has been presented as a means for the recovery of optical networks from single-link failures without the allocation of valuable system resources. While the strategy in its simplest form performs well against already established schemes, the flexibility of  $L+1$  leaves many options to investigate possible ways to further increase performance.

## REFERENCES

- [1] Brackett, C.A. "Dense WDM networks," *Fourteenth European Conference on Optical Communication*, Conference Publication No. 292, Volume 1, 1988, pp. 533-540.
- [2] Pradhan, Dhiraj K. *Fault-tolerant Computer System Design*. Prentice Hall, Incorporated, Copyright 1996. Chapter 1, pp. 1-87.
- [3] Ramaswami, Rajiv and Sivarajan, Kumar N. *Optical Networks: A Practical Perspective*. Academic Press, Copyright 1998. Chapter 8, pp. 329-397 and Chapter 10.4, pp. 430-451.
- [4] Falcao, P. F. "Pan-european multi-wavelength transport networks: Network: design, architecture, survivability and SDH networking," *Proceedings of the 1st International Work-shop on Reliable Communication Networks*, Brugge, Belgium, May 17-20, 1998.
- [5] Li, C. S, and Ramaswami, R. S. "Automatic fault detection, isolation and recovery in transparent all-optical networks," *Journal of Lightwave Technology*, Volume 15, Issue 10, October 1997, pp. 1784-1793.
- [6] Sahasrabudde, L, Ramamurthy, S. and Mukherjee, B. "Fault management in IP-over-WDM networks: WDM protection versus IP restoration," *IEEE Journal on Selected Areas in Communications*," Volume 20, Issue 1, January 2002, pp. 21-33.
- [7] Mohan, G, Siva Ram Murthy, C. and Somani, A.K. "Efficient algorithms for routing dependable connections in WDM optical networks," *IEEE/ACM Transactions on Networking*, Volume 9, Issue 5, October 2001, pp. 553-566.
- [8] Srinivasan, R. and Somani, A.K. "A generalized framework for analyzing time-space switched optical networks," *IEEE Journal on Selected Areas in Communications*, Volume 20, Issue 1, January 2002, pp. 202-215.
- [9] Zang, Hui, Jue, J.P, Sahasrabudde, L, Ramamurthy, R and Mukherjee, B. "Dynamic lightpath establishment in wavelength routed WDM networks," *IEEE Communications Magazine*, Volume 39, Issue 9, September 2001, pp. 100-108.
- [10] Ramamurthy, R. and Mukherjee, B. "Fixed-alternate routing and wavelength conversion in wavelength-routed optical networks," *IEEE/ACM Transactions on Networking*, Volume 10, Issue 3, June 2002, pp. 351-367.

- [11] Fang, J, Srinivasan, R. and Somani, A. "Performance Analysis of WDM Networks with Wavelength Usage Constraint," *Technical Report*, Dependable Computing & Networking Laboratory, Department of Electrical and Computer Engineering, Iowa State University.
- [12] Srinivasan, R. and Somani, A.K. "Request-specific routing in WDM grooming networks," *2002 IEEE International Conference on Communications*, Volume 5, pp. 2876-2880.
- [13] Srinivasan R. "Dynamic routing in WDM grooming networks," *Technical Report*, Dependable Computing & Networking Laboratory, Department of Electrical and Computer Engineering, Iowa State University, August, 2001.
- [14] Wackerly, Dennis D, Mendenhall III, William and Scheaffer Richard L. *Mathematical Statistics and Applications, Sixth Edition*. Wadsworth Group, and division of Thomson Learning, Inc, Copyright 2002. Chapter 8, pp. 365-415.

## APPENDIX

### A. Confidence Intervals

Large sample confidence intervals based on the normal distribution, according to [14], were constructed using the unbiased estimators of the mean of a sample,  $\bar{y}$  and the standard deviation of a sample,  $s^2$ , according to Equation [9], where  $z_{\alpha/2}$  is the z-score and  $n$  is the sample size.

$$\hat{\theta} = \bar{y} - z_{\alpha/2} \sqrt{\frac{s^2}{n}}$$

[9]

Arrival Rate	No Backup		L+1 Fault Tolerance		Backup Multiplexing	
	BP <sub>AVE</sub>	$\theta$	BP <sub>AVE</sub>	$\theta$	BP <sub>AVE</sub>	$\theta$
30	0	0.00E+00	0.000001	1.96E-06	0.01056	5.34E-04
40	0	0.00E+00	0.000105	1.47E-05	0.047591	1.06E-03
50	0.000004	5.99E-06	0.000701	5.28E-05	0.101241	1.44E-03
60	0.000037	1.63E-05	0.003759	1.26E-04	0.156341	1.38E-03
70	0.000216	5.26E-05	0.011981	3.40E-04	0.205883	1.49E-03
80	0.000791	9.48E-05	0.027684	7.17E-04	0.250215	1.31E-03
90	0.002132	1.29E-04	0.048408	5.32E-04	0.288971	1.40E-03
100	0.004778	2.27E-04	0.075712	7.68E-04	0.322454	1.89E-03

Table 1. NSFNET 95% Confidence Intervals for Average Blocking Probability

Arrival Rate	No Backup		L+1 Fault Tolerance		Backup Multiplexing	
	BP <sub>AVE</sub>	$\theta$	BP <sub>AVE</sub>	$\theta$	BP <sub>AVE</sub>	$\theta$
20	0	0	0.000022	8.16E-06	0.031199	7.85E-04
30	0.000005	3.27E-06	0.001395	0.000146	0.120457	1.08E-03
40	0.000133	2.3E-05	0.013122	0.00022	0.206296	1.20E-03
50	0.000992	8.74E-05	0.045268	0.000558	0.274619	9.06E-04
60	0.004312	0.000243	0.091392	0.000599	0.329521	1.73E-03
70	0.011887	0.000329	0.143169	0.001206	0.372922	1.99E-03

Table 2. ARPA-2 95% Confidence Intervals for Average Blocking Probability

Arrival Rate	No Backup		L+1 Fault Tolerance		Backup Multiplexing	
	BP <sub>AVE</sub>	$\theta$	BP <sub>AVE</sub>	$\theta$	BP <sub>AVE</sub>	$\theta$
100	0.000057	1.46E-05	0.002578	1.64E-04	0.139429	1.40E-03
110	0.00013	2.39E-05	0.006296	2.91E-04	0.168468	1.27E-03
120	0.000343	5.15E-05	0.012489	3.32E-04	0.196457	1.61E-03
130	0.000661	5.44E-05	0.021418	4.87E-04	0.222939	1.30E-03
140	0.001328	8.56E-05	0.032646	8.44E-04	0.247825	1.47E-03
150	0.002464	1.06E-04	0.046797	7.35E-04	0.271067	1.74E-03
160	0.004	1.10E-04	0.060898	1.00E-03	0.292598	1.18E-03
170	0.006269	2.23E-04	0.077407	1.12E-03	0.313237	1.18E-03
180	0.009181	2.83E-04	0.094778	1.25E-03	0.331851	1.64E-03

**Table 3. 4x4 Mesh Torus 95% Confidence Intervals for Average Blocking Probability**

Arrival Rate	No Backup		L+1 Fault Tolerance		Backup Multiplexing	
	L <sub>AVE</sub>	$\theta$	L <sub>AVE</sub>	$\theta$	L <sub>AVE</sub>	$\theta$
30	2.087572	1.48E-03	2.0865821	1.39E-03	2.0790342	1.29E-03
40	2.087572	1.48E-03	2.0866061	1.44E-03	2.0534466	1.49E-03
50	2.0875694	1.48E-03	2.0863164	1.51E-03	2.0146823	1.74E-03
60	2.0875643	1.49E-03	2.0846694	1.89E-03	1.97326	1.82E-03
70	2.0873718	1.52E-03	2.0820118	1.59E-03	1.9339341	1.65E-03
80	2.0871289	1.54E-03	2.0739897	8.96E-04	1.8961763	1.82E-03
90	2.0862719	1.40E-03	2.0629532	1.35E-03	1.8626909	1.63E-03
100	2.084284	1.57E-03	2.0502485	1.46E-03	1.831468	2.55E-03

**Table 4. NSFNET 95% Confidence Intervals for Average Path Length**

Arrival Rate	No Backup		L+1 Fault Tolerance		Backup Multiplexing	
	L <sub>AVE</sub>	$\theta$	L <sub>AVE</sub>	$\theta$	L <sub>AVE</sub>	$\theta$
20	3.417037	0.003863405	3.415139	0.002284906	3.4016506	3.01E-03
30	3.417036	0.003856904	3.4129021	0.002796956	3.2438361	2.58E-03
40	3.4167445	0.003888839	3.3966746	0.002953215	3.0878024	3.12E-03
50	3.4151097	0.004066274	3.3539989	0.002473868	2.9587609	2.80E-03
60	3.4092818	0.004210912	3.2919006	0.002695818	2.8494018	2.80E-03
70	3.3956751	0.003714688	3.2207957	0.003717424	2.7624477	4.21E-03

**Table 5. ARPA-2 95% Confidence Intervals for Average Path Length**



Arrival Rate	No Backup		L+1 Fault Tolerance		Backup Multiplexing	
	$L_{AVE}$	$\theta$	$L_{AVE}$	$\theta$	$L_{AVE}$	$\theta$
100	2.1333967	1.52E-03	2.1289086	1.16E-03	1.9901392	1.47E-03
110	2.1333683	1.40E-03	2.1247455	1.49E-03	1.9609114	1.87E-03
120	2.1329546	1.62E-03	2.1169729	1.47E-03	1.9329988	2.10E-03
130	2.1327658	1.57E-03	2.1059826	1.38E-03	1.9061724	2.30E-03
140	2.1313223	1.69E-03	2.0927059	2.65E-03	1.881414	1.53E-03
150	2.1303941	1.40E-03	2.0769708	1.88E-03	1.8570302	2.18E-03
160	2.1286455	1.21E-03	2.0570607	1.43E-03	1.8352661	1.89E-03
170	2.1266156	1.58E-03	2.0389033	1.51E-03	1.8134678	1.84E-03
180	2.1232379	1.83E-03	2.0200209	1.01E-03	1.7932852	2.14E-03

**Table 6. 4x4 Mesh Torus 95% Confidence Intervals for Average Path Length**

Arrival Rate	No Backup		L+1 Fault Tolerance		Backup Multiplexing	
	$U_{AVE}$	$\theta$	$U_{AVE}$	$\theta$	$U_{AVE}$	$\theta$
30	0.08509125	6.04E-05	0.085050815	5.65E-05	0.083848329	5.66E-05
40	0.113455	8.05E-05	0.11339061	7.76E-05	0.106289239	1.57E-04
50	0.141818003	1.01E-04	0.1416341	1.01E-04	0.123010666	2.70E-04
60	0.170175571	1.21E-04	0.16930703	1.50E-04	0.135714293	3.12E-04
70	0.198484334	1.49E-04	0.19564496	1.48E-04	0.146065367	3.56E-04
80	0.226682391	1.68E-04	0.21919281	2.15E-04	0.154535543	3.60E-04
90	0.25457087	1.77E-04	0.24005177	2.21E-04	0.161954633	3.89E-04
100	0.281837636	2.00E-04	0.25747541	2.13E-04	0.168602038	6.45E-04

**Table 7. NSFNET 95% Confidence Intervals for Normalized Ave. Effective Utilization**

Arrival Rate	No Backup		L+1 Fault Tolerance		Backup Multiplexing	
	$U_{AVE}$	$\theta$	$U_{AVE}$	$\theta$	$U_{AVE}$	$\theta$
20	0.082140313	9.28703E-05	0.082092885	5.47412E-05	0.078333005	8.54E-05
30	0.12320982	0.000139283	0.12288969	0.000102908	0.102066274	1.52E-04
40	0.164244712	0.000186731	0.16115884	0.00015886	0.117157115	2.24E-04
50	0.20503137	0.000245433	0.19243796	0.000107469	0.128441166	1.68E-04
60	0.244801587	0.000332224	0.21570056	0.000248758	0.137304231	4.14E-04
70	0.282297716	0.00030369	0.23218477	0.000500293	0.145326815	5.73E-04

**Table 8. ARPA-2 95% Confidence Intervals for Normalized Ave. Effective Utilization**

Arrival Rate	No Backup		L+1 Fault Tolerance		Backup Multiplexing	
	$U_{AVE}$	$\theta$	$U_{AVE}$	$\theta$	$U_{AVE}$	$\theta$
100	0.208327637	1.53E-04	0.20736523	1.05E-04	0.167251855	3.73E-04
110	0.229140635	1.54E-04	0.22680711	1.65E-04	0.175158135	3.83E-04
120	0.249869883	2.04E-04	0.24498444	1.92E-04	0.182021836	5.38E-04
130	0.270582305	2.09E-04	0.26163465	1.67E-04	0.188045254	5.34E-04
140	0.291004766	2.47E-04	0.27677179	4.52E-04	0.193478223	4.87E-04
150	0.311300537	2.31E-04	0.2900064	4.47E-04	0.1982896	6.85E-04
160	0.331270469	2.06E-04	0.30184235	4.40E-04	0.202855313	5.08E-04
170	0.350838164	3.08E-04	0.31228817	4.02E-04	0.206760176	5.36E-04
180	0.36979875	3.05E-04	0.32142833	5.85E-04	0.210618809	7.29E-04

**Table 9. 4x4 Mesh Torus 95% Confidence Intervals for Normalized Ave. Effective Utilization**

Arrival Rate	No Backup		L+1 Fault Tolerance		Backup Multiplexing	
	$PR_{AVE}$	$\theta$	$PR_{AVE}$	$\theta$	$PR_{AVE}$	$\theta$
30	1	0.00E+00	0.23334388	7.17E-04	0.090392791	5.62E-05
40	1	0.00E+00	0.23334332	7.02E-04	0.089280287	6.48E-05
50	1	0.00E+00	0.23328949	5.90E-04	0.087594883	7.56E-05
60	1	0.00E+00	0.23326458	4.91E-04	0.085793913	7.93E-05
70	1	0.00E+00	0.23293856	5.66E-04	0.084084091	7.19E-05
80	1	0.00E+00	0.23209087	4.49E-04	0.082442448	7.93E-05
90	1	0.00E+00	0.23075845	4.97E-04	0.080986561	7.08E-05
100	1	0.00E+00	0.22933054	4.73E-04	0.079629043	1.11E-04

**Table 10. NSFNET 95% Confidence Intervals for Ave. Probability of Reassignment**

Arrival Rate	No Backup		L+1 Fault Tolerance		Backup Multiplexing	
	$PR_{AVE}$	$\theta$	$PR_{AVE}$	$\theta$	$PR_{AVE}$	$\theta$
20	1	0	0.25377363	0.000521	0.130832715	1.16E-04
30	1	0	0.25372514	0.000604	0.124762927	9.91E-05
40	1	0	0.25284294	0.000442	0.118761631	1.20E-04
50	1	0	0.24924945	0.000592	0.113798496	1.08E-04
60	1	0	0.24494323	0.000614	0.109592377	1.08E-04
70	1	0	0.23972717	0.000654	0.106247988	1.62E-04

**Table 11. ARPA-2 95% Confidence Intervals for Ave. Probability of Reassignment**

Arrival Rate	No Backup		L+1 Fault Tolerance		Backup Multiplexing	
	PR <sub>AVE</sub>	$\theta$	PR <sub>AVE</sub>	$\theta$	PR <sub>AVE</sub>	$\theta$
100	1	0.00E+00	0.51638283	5.47E-04	0.06219185	4.60E-05
110	1	0.00E+00	0.51539946	5.20E-04	0.061278481	5.84E-05
120	1	0.00E+00	0.51303294	6.21E-04	0.060406213	6.58E-05
130	1	0.00E+00	0.51022614	4.58E-04	0.059567888	7.18E-05
140	1	0.00E+00	0.50629566	9.86E-04	0.058794188	4.78E-05
150	1	0.00E+00	0.50200147	7.13E-04	0.058032194	6.81E-05
160	1	0.00E+00	0.49604651	5.89E-04	0.057352066	5.89E-05
170	1	0.00E+00	0.49062722	5.59E-04	0.056670869	5.75E-05
180	1	0.00E+00	0.48506415	3.88E-04	0.056040163	6.70E-05

**Table 12. 4x4 Mesh Torus 95% Confidence Intervals for Ave. Prob. of Reassignment**