ADAPTIVE AND FLEXIBLE CIRCUIT SURVIVABILITY IN OPTICAL NETWORKS

BY

VISHAL SUNDARRAJAN
B.E. ANNA UNIVERSITY, CHENNAI, INDIA (2014)

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE
COMPUTER ENGINEERING
UNIVERSITY OF MASSACHUSETTS LOWELL

Signature of Author: ______________________ **Date:** __________

Signature of Thesis Chair: ______________________
Vinod M. Vokkarane

Signatures of Other Thesis Committee Members

Committee Member Signature: ______________________
Yan Luo

Committee Member Signature: ______________________
Jay A. Weitzen

ADAPTIVE AND FLEXIBLE CIRCUIT SURVIVABILITY IN OPTICAL NETWORKS

BY
VISHAL SUNDARRAJAN

ABSTRACT OF A THESIS SUBMITTED TO THE FACULTY OF THE
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF
MASTER OF SCIENCE
COMPUTER ENGINEERING
UNIVERSITY OF MASSACHUSETTS LOWELL
2017

Dissertation Supervisor: Vinod M. Vokkarane, Ph.D.
Professor, Department of Electrical and Computer Engineering

ABSTRACT

Modern and emerging telecommunication networks employ optical technology in the backbone. Providing high-speed connectivity, high bandwidth and reliable networks, connection failures within backbones may lead to social and economic damages which may be catastrophic. Therefore, survivability is a critical aspect for the success and the delivery of data in achieving end to end connectivity requirements in optical networks. The primary focus of this research is to investigate the effectiveness of two traditional survivable link disjoint routing algorithms, Bhandari's algorithm and iterative Djikstra's algorithm in order to compare them for performance. The performance metrics include connection blocking, connection failure and hop counts. This research also proposes a novel heuristic link avoidance routing heuristics which is presented as an alternative to the others. Stochastic models resembling the realistic large scale traffic characteristics is used to quantitatively and qualitatively evaluate the survivability approaches under various load conditions. Independent single link failures are injected in order to examine the relative performance benefits in terms of survivability on imperfect networks.

In this research, Department of Energy's Energy Science Network (ESnet) is considered. ESnet contributes scientific researchers from more than 40 institutions and facilitates all DOE application networking needs. ESnet can support up to 100 Gbps nationwide and move datasets equal to 20 billion books every month. The network is managed by a centralized circuit reservation controller On-Demand Secure Circuits and Advance Reservation System (OSCARS) v1.0 prototype. OSCARS invokes appropriate Path Computation Engine (PCE) which computes paths based on traditional and tailored routing algorithms to schedule end to end circuit resources in advance of reservation establishment. This controller is extensively used in this research to analyze the survivable algorithms for the considered performance metrics.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER 1
INTRODUCTION

This chapter briefly introduces optical network and how logical end to end IP connection can be achieved over optical networks. Further, it explains connection provisioning concepts and connection management.

1.1 Introduction to Optical Networks

The application in social media, mobile devices, and cloud computing increases the growth of Internet traffic in recent years. In order to serve these high data traffic, the medium through which data travels must support huge amount of data with high speed. There are two options available that offer the means to transport data- one is copper wire and other is optical fiber. The latter is far superior to the former.

An Optical network is a data communication network built with optical fiber technology. Optical network uses fiber optic cable as the medium of transmission for converting and passing data in the form of light pulses between sender and receiver nodes. Since optical network uses light as a medium, the data in the form of light travels faster and supports greater bandwidth than the electrical signals traveling through copper wire. The data loss is also lower and therefore reduces the intermediate signal boosters. Because of these reasons, Optical network is largely employed to carry traffic in high speed backbone networks.

Optical networks transmit data in the form of light. Light comes in many different wavelengths and these different wavelengths can be combined on the same fiber. The goal is to prevent signals from interfering with each other. To achieve this, light can be generated at various wavelengths and switched between different wavelength channels. This is called Wavelength-Division Multiplexing (WDM). In an all-optical network, signals remain in the optical domain from sender to receiver node without being converted to electrical, thus eliminating the electro-optic bottleneck[1]. The optical network consists of optical wavelength routing nodes interconnected by optical fiber links. This architecture allows a lightpath(s) to be established between any two nodes. A lightpath represents a direct optical connection between any two routing nodes. Normally, same wavelength is used along the lightpath. This limitation is called wavelength continuity constraint. This constraint can also be relaxed with the use of wavelength converters. However, it is very expansive to have an optical network with network nodes that have the capability of wavelength conversion[2].

1.1.1 Routing in Optical Network

In optical network there are two problems to establish a lightpath. The first is to determine a physical route along which the lightpath can be established. The second is to assign a wavelength to the selected path based on whether the node has the capability to convert wavelengths. The existing lightpaths cannot be re-routed to accommodate the new lightpath until they release the wavelength. Hence, some lightpath requests cannot be established if network have no free wavelengths. There are two common approaches to solve this routing problem in WDM, fixed and adaptive routing. In fixed, the routing problem is first solved between each pair of nodes before assigning a wavelength. This is simpler but ineffective approach. This is because a fixed route is predetermined and if its in use, the future request for lightpaths may not be established if there are no free wavelengths although an alternative path exists. A slight modification to fixed path routing is fixed alternate. Instead of one fixed path between nodes, fixed alternate calculates $'n'$ number of paths between a pair of nodes. This increases the probability of success in setting up the lightpath. The major drawback of both these approaches is that neither of them considers the current state of the network. If the predetermined fixed path are not available, then the lightpath will not be set up. An adaptive routing approach considers the current state of the network. It is an unconstrained routing scheme that considers all the paths between any two nodes and a shortest path can be obtained by dynamic shortest path algorithms based on link costs at the time of lightpath requests. This thesis will cover more about adaptive routing in the upcoming chapters.

1.1.2 Advance Reservation

In advance reservation, network operator knows the exact serving time of lightpath set up. This thesis considers that the requests for lightpath arrive in advance. This means that if a request for lightpath arrives, then the serving time is considered to be somewhere in future. Reserving requests for lightpath in advance allows us to remodel the request parameters. For example, if a request for lightpath cannot be reserved for a particular request specifications, then it is possible to alter the request parameters such that it can be reserved.

1.2 IP Over Optical Networks

Emerging use of optical network for transport technologies lead to innovations in optical core networks. This moved Internet transport infrastructure towards modeling of high-speed optical routers interconnected by optical switches [3]. Equipments for WDM such as reconfigurable optical cross connects (OXC) have emerged to build very high capacity networks. Also, WDM enables multiple OC-48 (2.5 Gbps) and OC-192 (10 Gbps) communication channels in the form of wavelengths or frequencies to operate in parallel over a single fiber optic cable. Several architectures have been introduced in the past to integrate IP and WDM technologies [4]. This research now explains one of the very common architectures: Interconnection models.

1.2.1 Interconnection Models

The Interconnected model consists of IP (Internet Protocol) routers attached to an optical core network. IP router is a device capable to route adaptively between two nodes based on IP routing protocols[5]. There can be more than one IP routing approach as specified in [4]. On the other hand, the optical core network consists of multiple optical cross connects interconnected by fiber optic links. These OXCs are capable of switching a data stream, controlled by properly configuring the cross connect table[3]. Thus, a switched optical path is established between IP routers. This research will discuss more about these routing (IP routing) in Chapter 2.

1.2.2 Logical Connection in IP Over Optical Network

In the previous section, it is clear that IP routing is possible in optical networks. In IP networks, setting up the path between two nodes can be realized logically rather physically. For example, a logical connection spans multiple physical nodes and links that are part of separate physical network. That means, a logical connection is a virtual representation but appears as a separate and self-contained network circuit even though it might physically constitute a small portion of a large shared network. To achieve this virtual logical connection set up, each connection has been given its own connection resources. This resources include bandwidth availability in fiber links, ports, and VLAN tags[5] availability. VLAN tag is label used to identify a virtual connection. Each end to end connection is backed by a VLAN tag.

1.3 Introduction to Survivability

Since optical networks carry high volume of data, any disruption in connection can cause both economic and social damages. Fiber links are likely to carry large amount of data because of the high capacity they offer. It is possible to have a data loss up to 50 TB every 8 seconds if a fiber link goes down. Hence, it is extremely important to ensure that communications crossing these links and networks should be properly protected against catastrophic failure events. The physical layer of optical network is unfortunately vulnerable to a variety of failures. These failures may be planned or unplanned. Almost 25 percent of failures in networks are planned due to maintenance [6]. However, unplanned failures may lead to logical connection disruption because its unpredictable and beyond the network operator's knowledge. These failures may happen due to disasters, digging works, terrorist attacks. This research considers unplanned independent single link failures. Independent single link failure poses a great deal of challenge to protected connections. Double link or node failures may lead to unnecessary blocking of survivable connections. Hence, a realistic single link failure distribution is employed. Chapter 3 explains more on how these accidental failures are distributed and their effects in the network.

Survivability is the ability of protecting connections from failures. Extensive research has already been done on the design of survivable algorithms for different transport technologies [7]. These survivability algorithms can be applied to any network not just optical. This thesis will discuss these algorithms briefly in Chapter 2. Survivability in optical networks can be achieved by providing two paths: one primary and one backup in advance of potential failures. These two paths are typically disjoint, meaning they do not share any common fiber links. The idea is to make routing as diverse as possible so that failure in the primary will not affect the backup paths. If the primary path fails, the traffic can be easily switched to the corresponding backup path, such that the overall connection can be protected. This approach is called path protection. One of the prime motives of this research is to perform a comparative analysis of three path protection algorithms, two existing iterative Dikstra, Bhandari and one proposed, risk aware link avoidance in terms of its connection blocking probability, connection failure probability and effect of path lengths. Connection blocking probability is the ratio of the number of connections that cannot be allocated resources for both primary and backup path to the total number of connection requests. This value describes the rate at which circuit establishment is unsuccessful due to resource availability and contention. Connection failure is the interrupted connections due to link failures after it has been successfully established. Note that the survivable connection fail only if both its primary and backup path fails. Thus, failure probability is the ratio of connection failures to the the total number of successful connections. Path length refers to the number of intermediate nodes through which the connection is established from sending to receiving node. Analysis shows that the proposed algorithm balances a decent tradeoff between all these parameters.

1.4 Centrally Controlled Network

In this research, this thesis assumes that the network is centrally controlled. Current trend towards centrally controlled networks relies on the separation of hardware and software. In a centralized architecture, the controller hosts all the logic in the central high performance super computer. This way, it is easy to decouple the software logic from network nodes, thus providing a greater visibility and control over the network [8] .

Relating this centrally controlled approach to this problem, the logically centralized controller (software) handles the process of setting up the connection over the optical nodes (hardware). Also, the controller has the central view and hence it is easy for the controller to locate the failure in a reasonable time [9]. Chapter 2 explains more briefly about the operation of this central controller.

In summary, survivability in optical network is an important constraint. This thesis considers two existing survivability algorithms, Iterative Dijkstra and Bhandari's link disjoint and compared in terms of its relative performance measures. This thesis also proposes one algorithm. To achieve this, a realistic independent single link failure is introduced in the network. This thesis discusses existing survivability algorithms in Chapter 2 and introduces realistic single link failures in Chapter 3. Simulation assumptions are explained in Chapter 4 and comparative analysis of existing survivability algorithms in Chapter 5. The proposed algorithm is discussed in chapter 6. Finally, the Conclusion along with future scope is discussed in Chapter 7.

CHAPTER 2
BACKGROUND

This chapter explains how the logical connection has been achieved in the centrally controlled optical network. Further, it details how this central controller can be exploited to achieve the survivability algorithms.

2.1 OSCARS v1.0

The On-Demand Secure Circuits and Advance Reservation System (OSCARS) developed by ESnet, is a central software tool designed to provision logical connections with end to end network resources. OSCARS serves as the central controller in ESnet. Previous versions of OSCARS could only provision a point to point logical connection in IP-over-WDM. Recently, it has been entirely reworked in collaboration with Advanced Communication Networks Laboratory (ACNL) at the University of Massachusetts Lowell to support a vastly more flexible framework that can dynamically select enhanced services based on connection request needs. In this research, OSCARS v1.0 has been extensively used as a deployable framework on which to perform comparative analysis of survivable networking approaches on reliable network scenarios.

2.1.1 Working of OSCARS

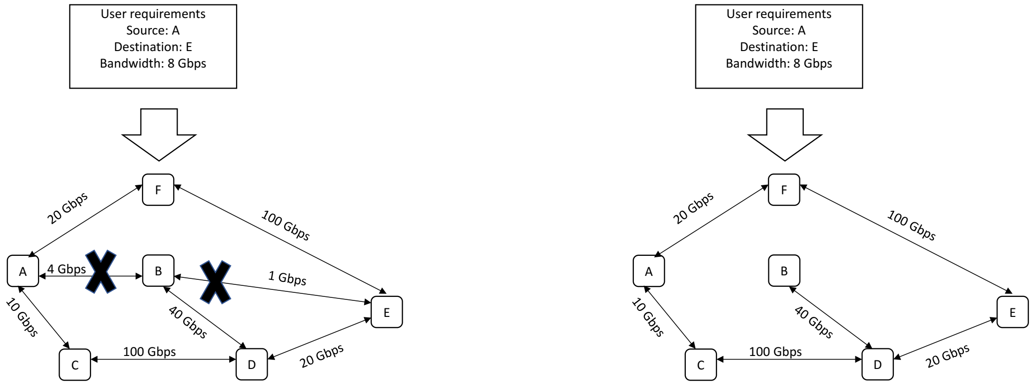
The path reservation of OSCARS is a two-step process,

1. Constraint-driven adaptive network resource pruning
2. Path finding based on availability of resources (bandwidth, VLAN tags)

2.1.1.1 Pruning

OSCARS does adaptive routing, meaning that it considers the current state of the network before path computation. When a request for a connection arrives, OSCARS removes the links and ports that cannot support the requested resources. These resources include link bandwidth, VLAN tags. This is called resource pruning.

Figure 2.1 shows how the pruning service in OSCARS may remove links from the topology that cannot support the bandwidth requirements in both forward and reverse directions. Assume the user makes a bandwidth request for 8 Gbps from A ->E. From A ->E, the pruning component removes any edges in the network link topology graph that have less than 8 Gbps of available capacity during the requested scheduling period. Figure 2.1 (b) shows the updated pruned network without insufficient bandwidth links.



(a) Pruning

(b) Pruned network before path computation

Figure 2.1. Pruning Service in OSCARS.

2.1.1.2 Path Computation

The Path-Computation Engine (PCE) in OSCARS v1.0 is built on a uniquely flexible framework that allows distinct path computation elements to be composed in an arbitrary workflow sequence. There are more PCE components than previous versions of OSCARS, along with their offered service enhancements and algorithms to compute the solution path. Prior to pathfinding, the pruning services should be employed to remove the links that cannot support requested bandwidth and VLAN tags. Then these PCE modules employ low level pathfinding algorithms to compute the least-cost path between source and destination nodes. This research is going to utilize these low level pathfinding algorithms for our survivable connection set up.

2.1.2 Services Offered by PCE

The following subsection only discusses the services offered by PCE components that are applied in this research. Other services can be found at [10].

2.1.2.1 Unicast Service

Unicast routing is the process of finding a path between two nodes, sender and receiver. This is the only service offered by all previous version of OSCARS and continues to be supported in the OSCARS v1.0. It is one of the basic services offered by the PCE to compute the shortest path between two nodes in the pruned network. The shortest path can be computed by invoking the Dijkstra [8] path finding module. Note that this is not the shortest route on the network but rather the shortest working solution after adaptive topology pruning. The shortest path is computed according to link cost, which has been precomputed using traffic engineering information by ESnet engineers. In Figure 2.2, the shortest path from A to E is calculated with Dijkstra.

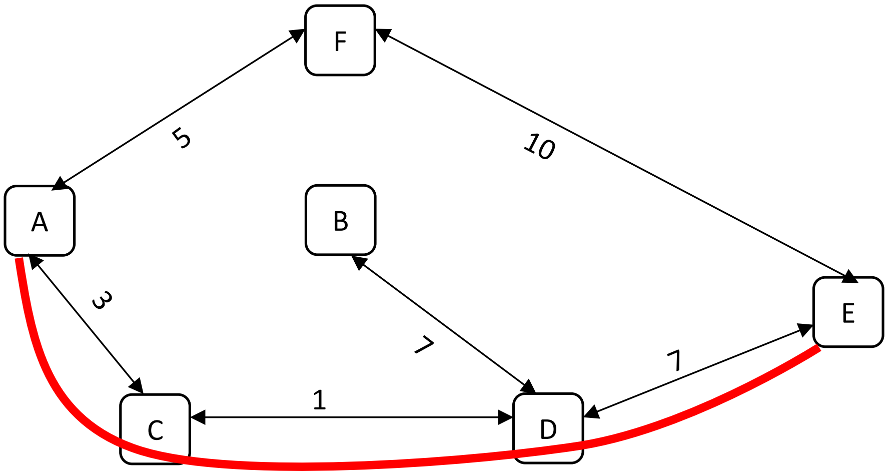


Figure 2.2. Unicast Routing.

2.1.2.2 Anycast Service

Anycast routing is very similar to unicast routing except that the sender node has the option of setting up the connection with any one of the specified receiver nodes. OSCARS treats all the candidate destinations equally and adaptively supports destination selection using path weights. OSCARS starts by finding a least-cost paths to each candidate node one by one. Then, OSCARS compares the cost of each and picks the one with overall least-cost path. The choice of receiver nodes can be decided by anycast degree. OSCARS supports anycast to increase the probability of setting up the connection over unicast by increasing the anycast degree. In Figure 2.3, the Sender can set up a connection with either E or C. It selects C over E because the cost of path established is lower. Anycast also employs Dijkstra to find the shortest path.

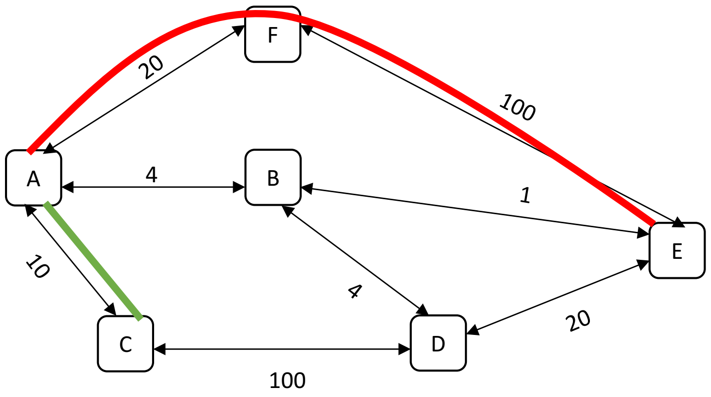


Figure 2.3. Anycast Routing.

2.1.2.3 Non-palindromic Service

Palindrome, as the name implies, should have both the forward and reverse paths the same. Previous versions of OSCARS supports only palindrome routing. However, OSCARS v1.0 introduces ’Non-palindrome’ routing that allows the forward and reverse path to be different. Figure 2.4 discusses this difference with an example.

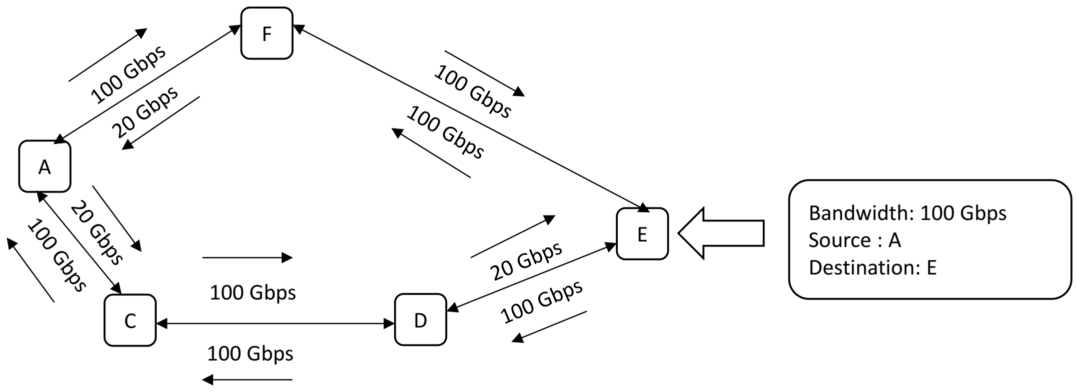


Figure 2.4. Palindrome.

In Figure 2.4, assume a request arrives for 100 Gbps to establish a connection between A and E in a bi-directional network. Once the links are pruned, there are no paths between A and E. This is because OSCARS prunes A-F and A-C and therefore, it is not possible for A to communicate with node E. With Non-palindrome option, only the forward link of A-C and reverse link of A-F need to be pruned. This still allows A to establish a forward delivery path through A-F-E and reverse response path through E-D-C-A.

2.1.2.4 Survivability Service

This service incorporates link-disjoint survivable routing of backup paths. The connection request may be for any number of disjoint paths and the obtained solution paths are assigned identical bandwidths. The only pathfinding algorithm in OSCARS v1.0 for survivability is Bhandari's link disjoint algorithm. This will be extensively discussed in the section 2.3.

2.1.3 Request and Response Specifications

OSCARS v1.0 allows user to interact to obtain information regarding the connection requirements through an API. The research makes use of this API [11] to submit connection requests to OSCARS. On receiving the requests, OSCARS processes them and responds with successful or unsuccessful reservation objects. The response object also returns the associated path(s) for the connections if successful. These request and response objects are briefly explained in the Table 2.1, 2.2.

Parameter	**Description**
Connection ID	Unique ID for a connection request.
Start time	Beginning time of a connection request.
End time	This is the time that reserved circuits for this connection are released. End time must be later than start time.
Source Node	Sending node for this connection.
Destination Node	Receiving node for this connection.
Source Port	Ports in a selected source node.
Destination Port	Ports in a selected destination node.
AZ Bandwidth	Forward delivery path bandwidth for the end to end connection request.
ZA Bandwidth	Reverse response path bandwidth for the end to end connection request.
Palindromic	Decides if forward and reverse paths are the same binary value.
Blacklist	Nodes and ports to be pruned explicitly from topology before path computation. Note that this is different from the pruning due to insufficient link or port bandwidth and VLAN tags.
Number of Paths	The number of disjoint backup paths for the connection.This thesis assumes a fixed value of 2 for all survivable scenarios.

Table 2.1. Input request specifications.

Parameter	Description
Connection ID	Identification of a connection.
Status	Status of connection request (Successful or Unsuccessful).
Path	Forward and Reverse paths for the connection. Returned as iterable sequence of node/port connections.

Table 2.2. Response parameters.

2.2 Survivability Algorithms

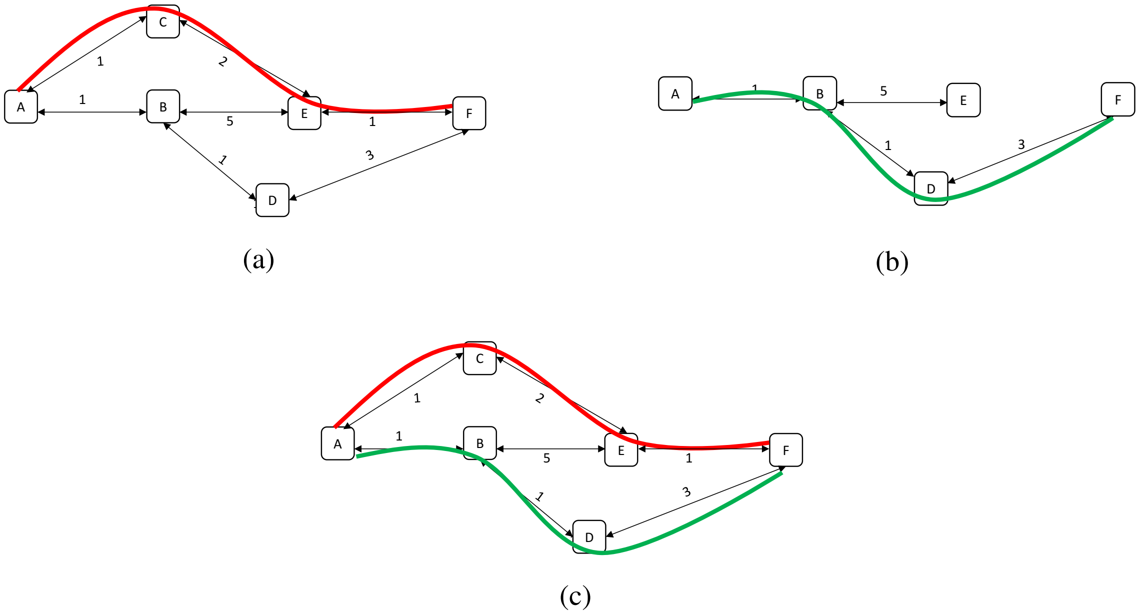
The survivability of a connection can be realized through physical diversity. This can be achieved by setting up a connection with two or more disjoint paths. The term disjoint in itself can be either node or link-disjoint. Fortunately, there are algorithms that can run in polynomial time to find node/link-disjoint paths (K$\geq$1) where, K is the number of backup paths. This research considers two common link-disjoint algorithms, iterative Dijkstra and Bhandari for their performances relative to considered performance metrics. This section is going to discuss how they can be implemented for survivable paths.

2.2.1 Iterative Shortest Path

This algorithm runs Dijkstra twice to find two disjoint paths. It is simple and straightforward but not a part of OSCARS v1.0. By making use of input specifier blacklists, which prune the specified links temporarily, it is possible to perform this algorithm with the help of OSCARS v1.0. Figure 2.5 illustrates the implementation steps one by one with the help of OSCARS.

(a) Finds the shortest path with Dijkstra.

(b) Removes the links in first shortest path from the network with the help of input specifier blacklists and runs Dijkstra again.



(a)

(b)

(c)

Figure 2.5. Iterative Dijkstra.

Note that to set up a survivable connections, OSCARS v1.0 has to be contacted twice.

2.2.1.1 Problem with Iterative Dijkstra

Figure 2.6 illustrates the possible scenario where iterative Dijkstra could not find a disjoint-backup path between A and F. Dijkstra greedily tries to allocate minimum resources to the primary path and fails to set up the backup path(s) even there are two paths.

In figure 2.6, iterative Dijkstra can find only one path between A-F but there are two disjoint paths, A-C-E-F and A-B-D-F. Since, it takes the least cost path A-B-E-F for primary, it cannot successfully set up the backup path.

2.2.2 Bhandari’s Link Disjoint

The problem in figure 2.6 can be overcome with another link-disjoint path algorithm, Bhandari’s link-disjoint path algorithm. It is based on Suurballe’s node-disjoint [12] algorithm. Note that Suurballe’s requirement is comparatively stricter than Bhandari’s as it does not

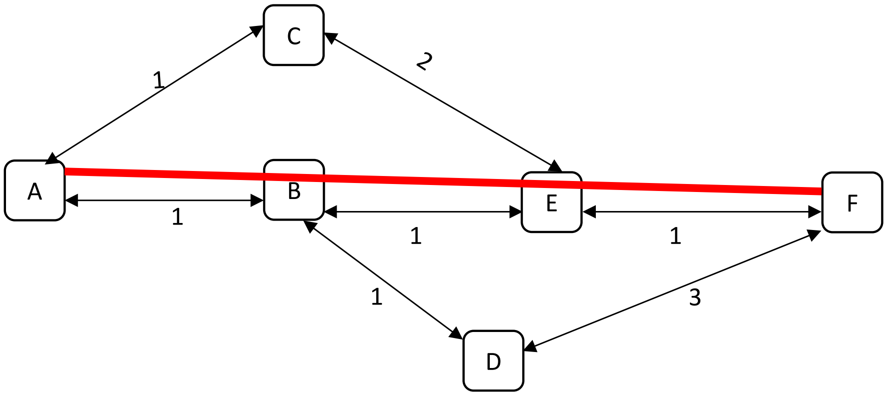


Figure 2.6. Problem with iterative Dijkstra.

allow primary-backup path pairs to share any node. Bhandari is only edge-disjoint. However, Bhandari has some implementation benefits. As discussed, OSCARS v1.0 is packaged with a native Bhandari module to support disjoint solutions.

This algorithm runs on top of single source shortest path algorithms Dijkstra (works only with positive edge weights) and Bellman Ford [13] (works with negative edge weights). These heuristics have been extensively utilized in network route establishments for decades. See Figure 2.7 for two shortest edge-disjoints paths between A and F with Bhandari. Note that this is the same network graph that iterative Dijkstra fails to compute primary-backup path pair.

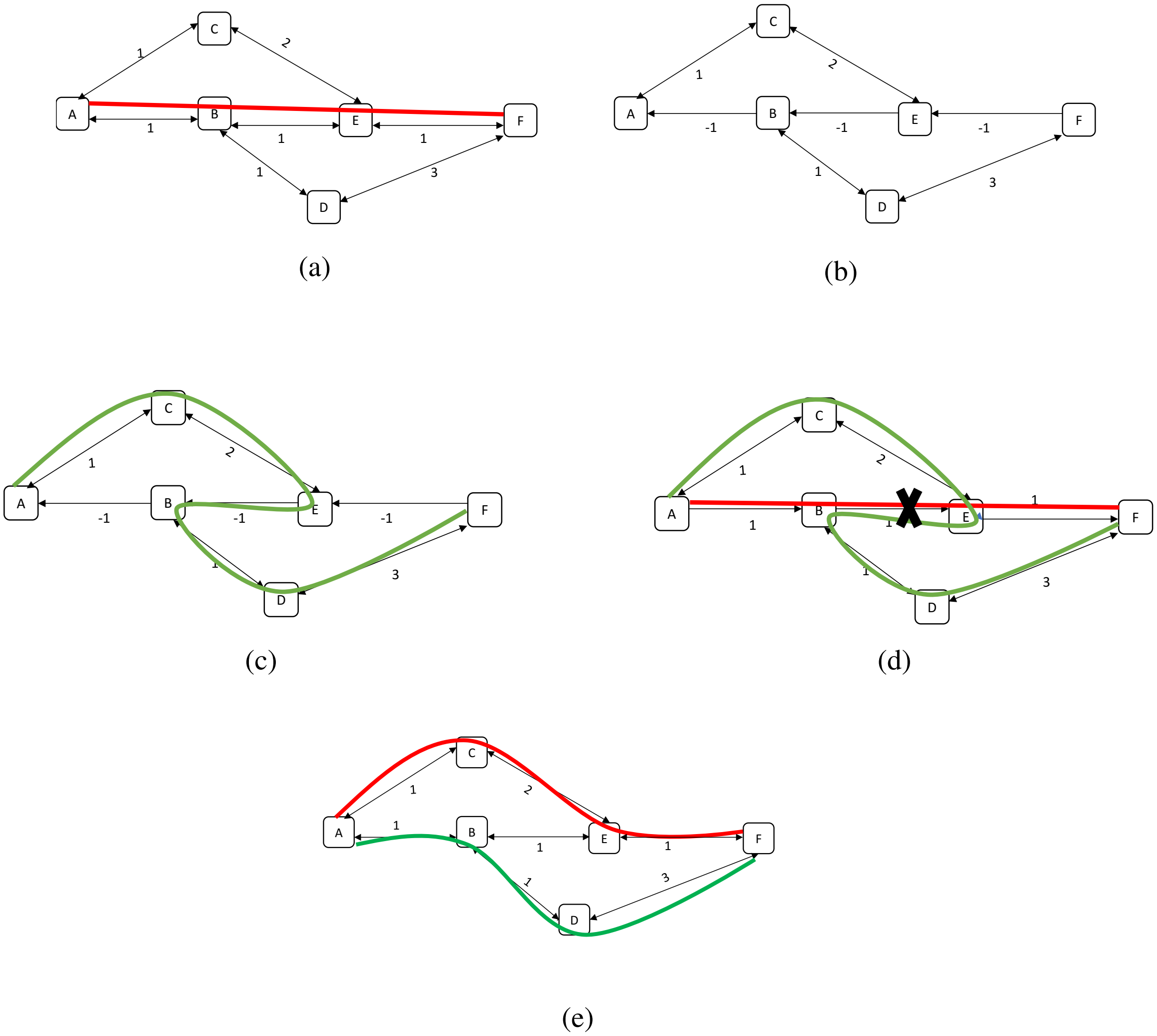
(a) Find the shortest path with Dijkstra.

(b) Reverse the edges of that shortest path with negative costs.

(c) Run the least cost path algorithm but this time Bellman Ford due to the introduction of negative edge costs.

(d) Remove the inverse edges obtained from the two paths. These inverse edges should be ignored in order to set up the disjoint paths

(e) Track all the edges from the two paths to get disjoint paths.



(a)

(b)

(c)

(d)

(e)

Figure 2.7. Bhandari's link disjoint algorithm

After obtaining the two disjoint paths, one would like to consider the least cost path as the primary or working path. The advantage of Bhandari is that it finds the total optimal cost and distribute amongst the paths. In other words, Bhamdari is not greedy as Djikstra and hence, it can find disjoint-paths everytime there is one. Hence, Bhandari is an excellent solution for protection scenario.

CHAPTER 3
NETWORK TOPOLOGY AND FAILURES

This chapter describes the topology, failure distribution, commonly applied failure and repair rates in the backbone networks.

3.1 Topology

The topology considered in this research is ESnet. ESnet is the Department of Energy's dedicated science network, helping scientists across the country meet their research goals. ESnet is optimized to help enormous data flows move quickly and efficiently around the world. ESnet is an all-optical network. This means that data transfer happens in the optical domain end to end without intermediate electro-optic conversion. ESnet is composed of heterogeneous optical nodes with differing routing and switching capabilities. The network may be used efficiently by considering these capabilities. Some nodes in the network posses Ethernet-level switching [14] while others are MPLS-enabled. Multi-protocol Label Switching (MPLS) is a switching method in which each hop-by-hop decision is based on a label. MPLS capable nodes allow traffic to route adaptively while Ethernet-only nodes does not.

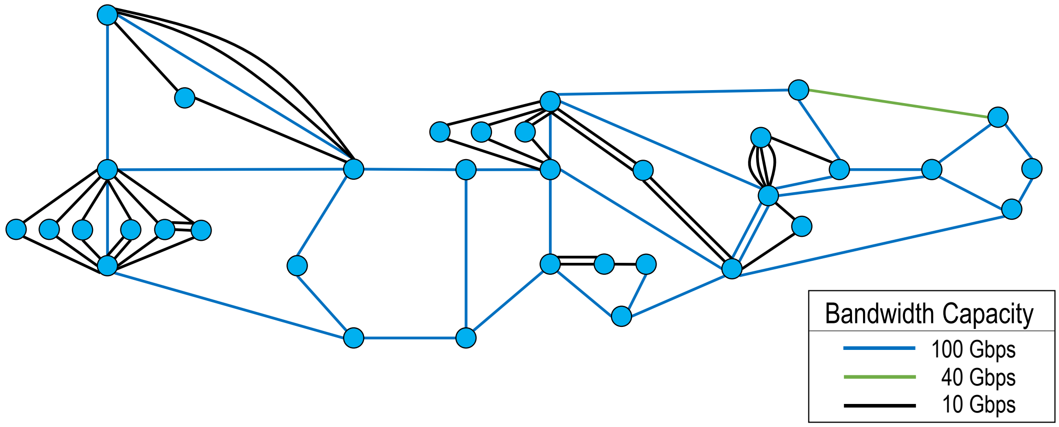


Figure 3.1. ESnet topology.

Figure 3.1 shows the ESnet topology. ESnet can transmit data at the rate of 100 Gbps. More than 40 percent of the links are 100 Gbps (precisely 34 in the above topology out of 73) and the remainder are 10 Gbps, while one of them supports 40 Gbps. It is assumed that almost all the blocking happens inside the network due to insufficient bandwidth availability during periods of heavy load. Links that can not support large flows have been omitted from the topology because most of them are used for management purposes rather than scientific data transmission. The bandwidth of access ports of switches (the entry point to the network) are all set to 100 Gbps to avoid blocking. This ensures that all the blocking only happens inside the core network domain.

3.2 Introduction to Failure Events

Analysis of survivability begins from introducing failures in the network topology and testing its performance against realistic failure scenarios. Optical networks are usually wellengineered and adequately provisioned, leading to very low packet losses and negligible queueing delays. This robust network design is one of the reasons why the occurrence and impact of failures in these networks have received little attention[15]. However, these reliability measures have little to guard links from accidental failures due to earthquake, terrorist attack, digging works. An in-depth understanding of reasons for such failures can only be known if vendors reveal the failure data of their respective network. Unfortunately, the lack of failure data from operational networks has further limited the investigation of failures in backbone networks. ESnet does not reveal any such failure data about the network. This work, therefore considers the theoretical distribution scenarios of accidental link failures. In most of the networks, 25% of the failures in the network are due to scheduled maintenance activities. This research considers only unplanned link failures due to fiber cuts and amplifier failures which may lead to unexpected connection interruption.

3.3 Failure Characteristics and Distribution

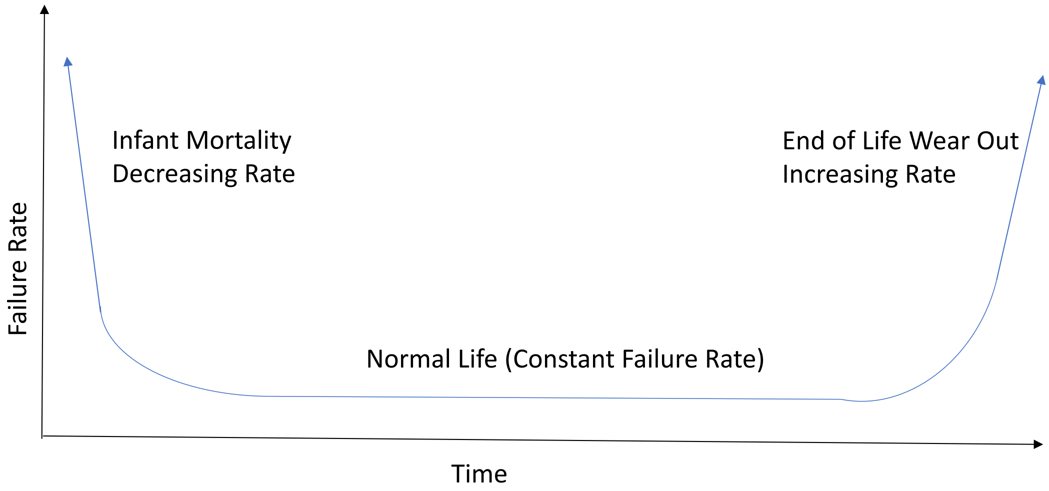


Figure 3.2. Bathtub curve

Figure 3.1 is called the bathtub curve and it does not depict the failure of a single item, but describes the relative failure rate of an entire population of products over time. It is common to assume Weibull distribution[16] for modeling failures for practical applications. This is because of the adaptability of this distribution to the bathtub curve. The curve is divided in to three regions and each of these regions behaves differently for failures. Any equipment in its early period has decreasing or zero failure rates because new items are less likely to fail often. This corresponds to the decreasing part of the bathtub curve. Later, failure rate remains constant throughout normal useful life. Here the failure rates are constant. This corresponds to the flat line of the bathtub curve. The third and the final region is ’wear-out’ region. Here, the failure rates increase to infinity and the components need to be replaced. This is the rightmost increasing part of the curve.

All three regions can be easily modeled with Weibull distribution[16]. In fact, the flat region of the curve exhibits exponential distribution. This is the region where the failures are random and can be used to model accidental failure scenarios. That said, it is not appropriate to consider other regions of bathtub curve unless more data on failures can be known. If the links are assumed to experience accidental failures due to external influence, it safe to consider the middle region of bathtub curve. This research does not consider deliberate failures or failure of links due to aging. Equipment vendors can model their own failure rate curve like bathtub curve and can set thresholds on replacements based on the demands and requirements of the customer.

The work in [17] explores the application of Weibull distribution and fits it to the network-wide time between failures. The work in [18] investigates UNINETT IP backbone on a per-link basis and fits the up time of these links to well-known distributions. It is important to consider that in [19], the links are characterized based on the length. For example, the short and medium distance links are fitted to Weibull distribution whereas the long-distance links are modeled with Gamma distribution [20]. To derive these failure data, the vendors should outsource the information and an accurate modeling can be possible by fitting each link to its corresponding distribution. However, the location and number of failures of each link should be observed for a reasonable period of time. The authors of [17], [21] also warn to avoid blindly apply these parameters to any other network as it may lead to a deception in the failure analysis. Another option would be assume Poisson distribution as the time between two failures in the same link is independent to each other. This is very appropriate to model the accidental failures caused in the link. This is because accidents causing failures in the same link at different times have no relation to each other. Hence, this research assumes arrival of failure follows Poisson distribution [22].

Many researchers from [21], [23], [24] have found that the failure of links is directly proportional to the distances and that failures of links are independent from each other. In fact, [23] has given the failure rates and repair rates of the links based on their installation types, which will be discussed in the following section.

3.4 Failure and Repair Rates

In reliability engineering, one should be very familiar with few terms.

(a) *Mean Time To Repair (MTTR) or Downtime*

The average time that it takes to repair a failed component.

(b) *Mean Time Between Failure (MTBF) or Uptime*

The average time that it takes for a link to fail.

(c) *Failure-In-Time (FIT)*

The term FIT is defined as a failure rate of 1 per billion hours.

This research fits MTTR and FIT values to link failures from extensive review of [21], [22], [23], [24] and derives approach based on majority consensus of these authors. Since not all ESnet links are buried (some of the them go under the sea), the FIT's for buried links were picked up from [21], overseas from[23] and amplifiers from [18], [19], [21]. The concluded final values are tabulated as follows.

Table 3.1. Summary of failure rate assumptions based on geographical location

Installation Type of Link	**FIT/km**	**Amplifier FIT**	**MTTR (in hrs)**
Buried	114	2850	20
Overseas	21.55	2850	450

Once the FIT of a link is known, the MTBF can be calculated by:

$$MTBF = (1000000000 \div FIT) \times Total\ link\ length$$

Using this formula, the MTBF can be calculated for all the links as shown in Figure 3.3.

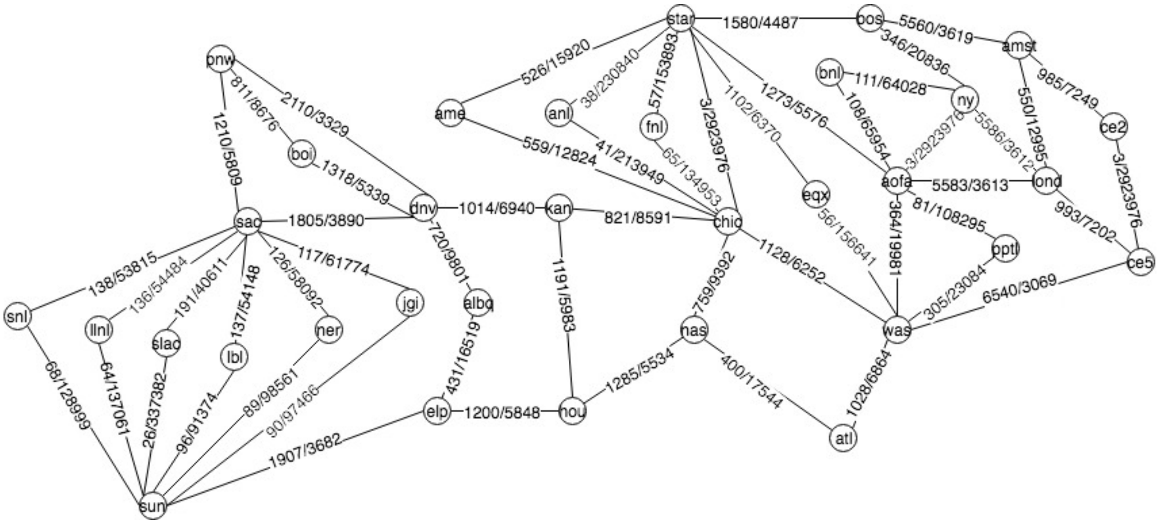


Figure 3.3. ESnet topology with distance/MTTF.

Type	Number of links
Buried	66
Overseas	6

Table 3.2. Number of links based on geographical location

3.5 Link Failure Overlaps

Overlaps is the number of link failures that overlap in time. By fitting MTTR and FIT values to the ESnet links, the overlaps in downtime of two or more links are observed. Figure 3.4, it is clear that overlaps are very rare yet some random overlaps tend to happen due to the presence of long distance links. Long distance links fail frequently and they have higher MTTR value. Such overlaps may lead to connection failures.

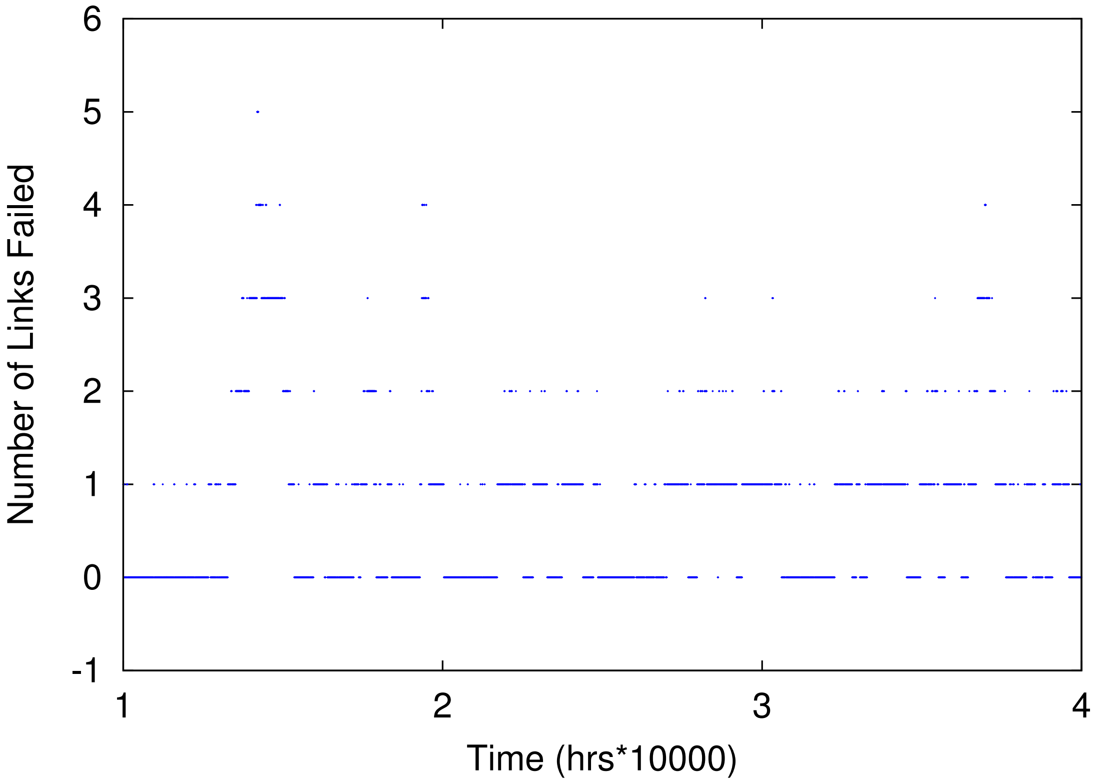


Figure 3.4. Link failure overlaps over time

It can be observed that majority of the time has zero or one failures. Rarely, the overlaps happen and the intensity of overlaps have decreased for three and four overlaps. This characteristic of overlaps in downtime of two or more links is good for survivability analysis as connection would still be able to survive in the event of single link failures in the system.

CHAPTER 4
SIMULATION ASSUMPTIONS

This chapter explains all the simulation assumptions and it's effects in the network. Also, it also discusses some of the factors causing failure of connections in the network.

4.1 Dynamic Traffic Generation

Source traffic modeling and traffic generation is a very important aspect of research about network communication. An accurate estimation to the network traffic is the basis to handle/control the traffic. Extensive research has been done in this area [25] and the approaches cover every aspect of dynamic traffic.

4.1.1 Traffic Modeling

Traffic generation and traffic modeling are two sides of the same coin. A good traffic model should develop a detailed understanding of the traffic characteristics of the network. Analysis of the traffic provides information like the blocking probability for various load, the bandwidth requirements, traffic flow from a source to destination for a given type of traffic (unicast, anycast) and numerous other details. Also, traffic models enable network designers to make assumptions about the networks being designed based on experience and enable prediction of reservation performance for future requirements.

A good dynamic traffic model demands a close approximation to the input traffic parameters such as arrival and holding time, bandwidth, source and destination for a request and their corresponding distributions.

4.1.1.1 Input Traffic Assumptions

In this research, the requests are assumed to arrive according to a Poisson process with average arrival rate λ and hold for a exponential holding time with average service rate μ. Hence, the load in the network can be calculated by λ/μ [26]. From [25], it is true that commercial load has diversified request bandwidths. The appropriate distribution in this case, would be uniform distribution. However, this generalized assumption cannot be valid for ESnet. ESnet offers high speed, high volume data transfer. Today, it carries approximately 20 petabytes of data every month. Hence, it is highly unlikely that a request of bandwidths in the order of Kbps or few Mbps would arrive. ESnet requests may demand high bandwidths as they arrive because the topology is frequently being updated by ESnet to support high volume of traffic. Therefore, uniform distribution of bandwidth may not be realistic as most of the links in ESnet are 100 Gbps and the range of usage is not clearly known. For this reason, this research considers normal distribution with a mean of high bandwidth in the order of Gbps. This nearly eliminates requests with the lower values of Mbps. Unlike bandwidth, the further inferences on distribution of traffic flows in the network cannot be made without accurate data. Unfortunately, ESnet does not reveal any flow related data because they may form business critical information. For this reason, the source and destination nodes are assumed to be uniformly distributed as load may be equally distributed across all access nodes. The simulation period is set to 1 year and the number of link failures in this 1 year period is 63.

4.1.2 Input Parameters for Traffic Generation

To determine a realistic performance evaluation, the input load is carefully decided for reasonable blocking probability. Repeated trials and accurate examinations have been made to fix a proper load range.

Traffic Parameter	Value	Distribution
Bandwidth	1 Gbps (variance of 400 Mbps)	Normal.
Total Devices	34	Uniform.
Ports in the device	Varies based on devices	Uniform.
Start Time	(-ln U/λ),U is uniformly distributed in the range 0 to 1	Discrete Poisson.
End Time	Start time + (-ln U/μ), U is uniformly distributed in the range 0 to 1	Exponential.

Table 4.1. Traffic distribution summary

All results shown in this chapter represent the average of 10 unique sets, and we have included the 95% confidence interval of all data points.

4.2 Factors Causing Failure of a Connection

In this section, we discuss factors that may cause failure of connections in the network. These impacts may be the result of network design and have a varying impact from one network to another. Probing in to such details will make it easy to examine the results in subsequent chapters.

4.2.1 Holding Time

From Figure 4.1, increase in holding time of the connection increases the probability of connection failures. This implies that connections with longer holding time seem to fail more often than the shorter ones. It complements Poisson arrival of failure events. When the time increases, it is more likely that an failure event may occur in one of the links that can result in connection disruption. To achieve this, unicast traffic has been generated for the simulation parameters discussed in this chapter for different average holding time for the connections.

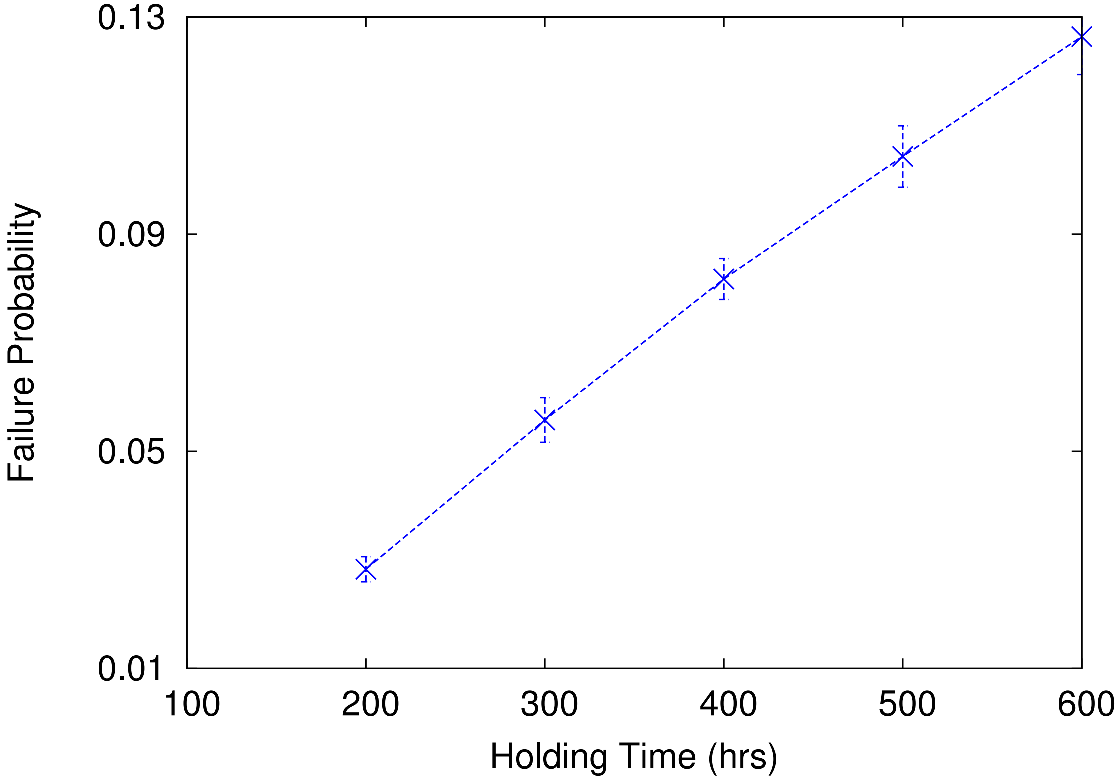


Figure 4.1. Failure rate vs holding time

4.2.2 Distance and Hop Counts

Figure 4.2 suggests that longer paths fail more often than shorter paths. Connections set up with longer paths are prone to link failures. This is because longer paths cover maximum number of long distance links to get to the destination. Thus, the probability that one of the links fail will also increase. To achieve this, anycast traffic has been generated to OSCARS for the variable load values. For each of these trend lines in Figure 4.2, the anycast degree has been increased by 1.

It is evident from [27] that anycast has spatial advantage over unicast but also indirectly affects failures. The anycast in OSCARS v1.0 prototype is built in such a way that a path

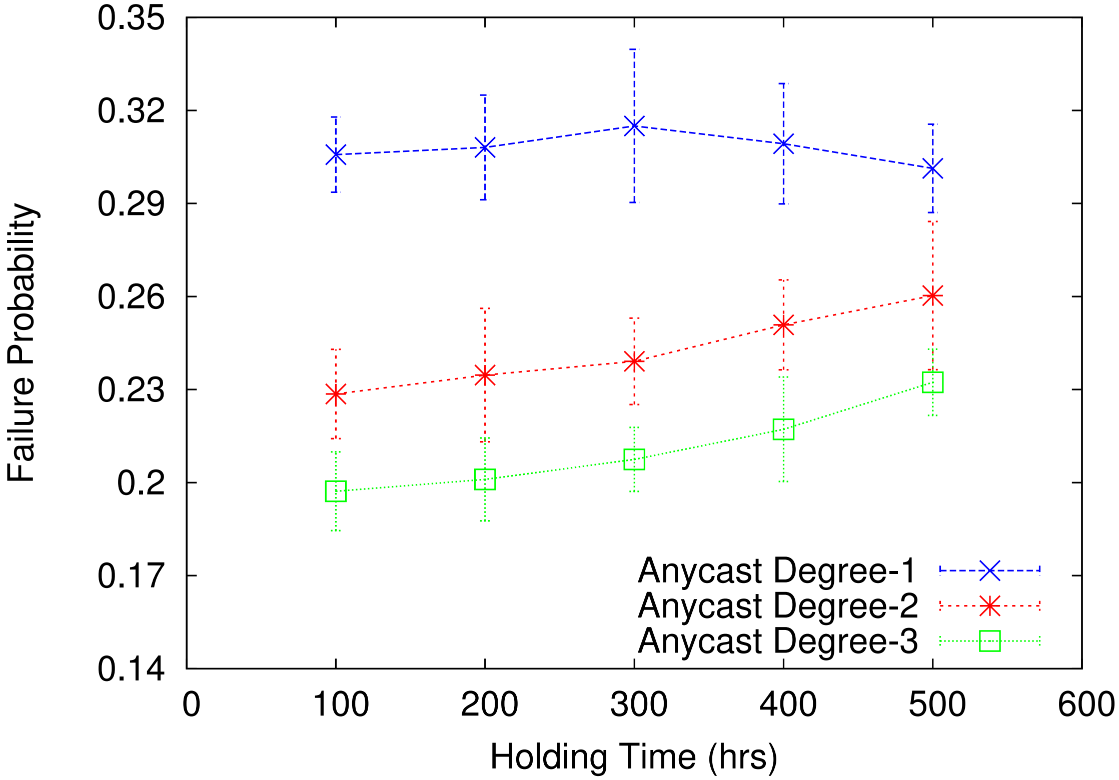


Figure 4.2. Failure rate vs distance

with minimum cost is preferred over others. The idea here is to decrease the resource allocation which may assist in decrease in failure rates as well.

CHAPTER 5
PERFORMANCE EVALUATION OF UNICAST SURVIVABILITY

This chapter analyzes the survivability algorithms and their sharp differences in their blocking probability, failure probability and resource allocation by introducing failures in the network as described in Chapter 3.

5.1 Performance Analysis

Before evaluating the survivability algorithms, the simulator is run for Palindromic requests to analyze if it has some advantage over non-palindrome in context of considered performance metrics. This aids in directing the future analysis of survivability to fix on one of these palindromic options. To achieve this, unicast traffic has been generated and failure has also been introduced in the network as discussed in the previous chapters.

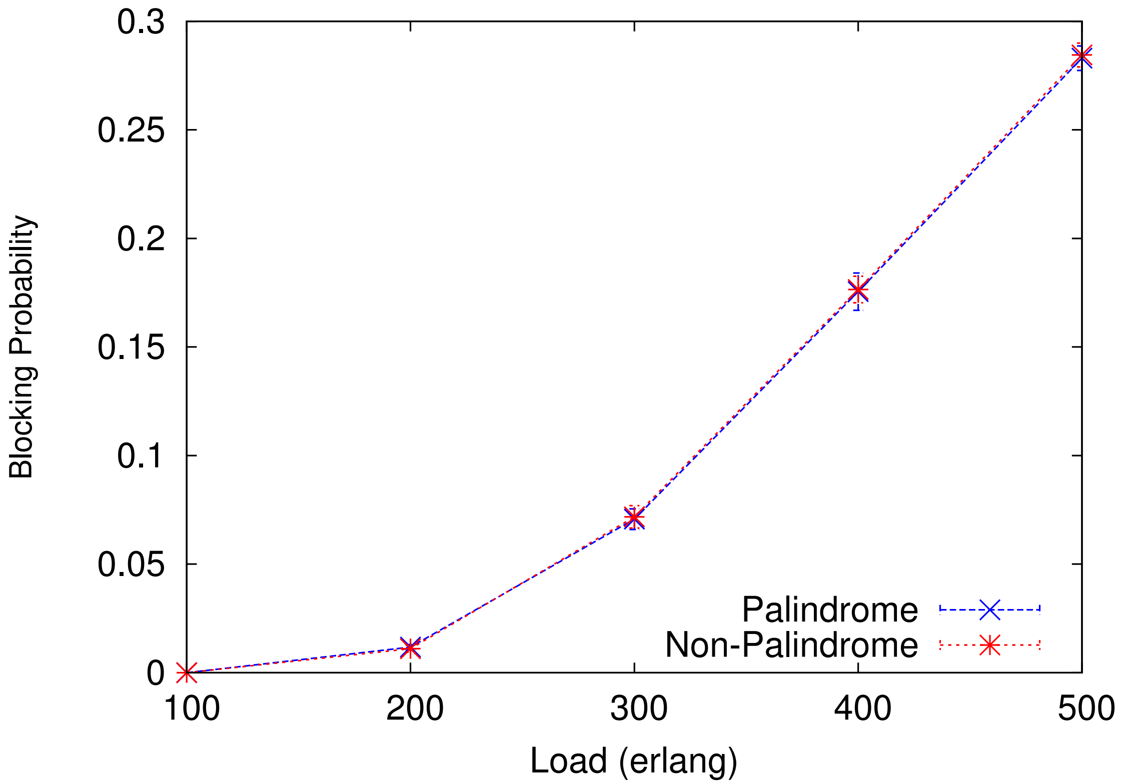


Figure 5.1. Palindrome vs non-palindrome - Blocking.

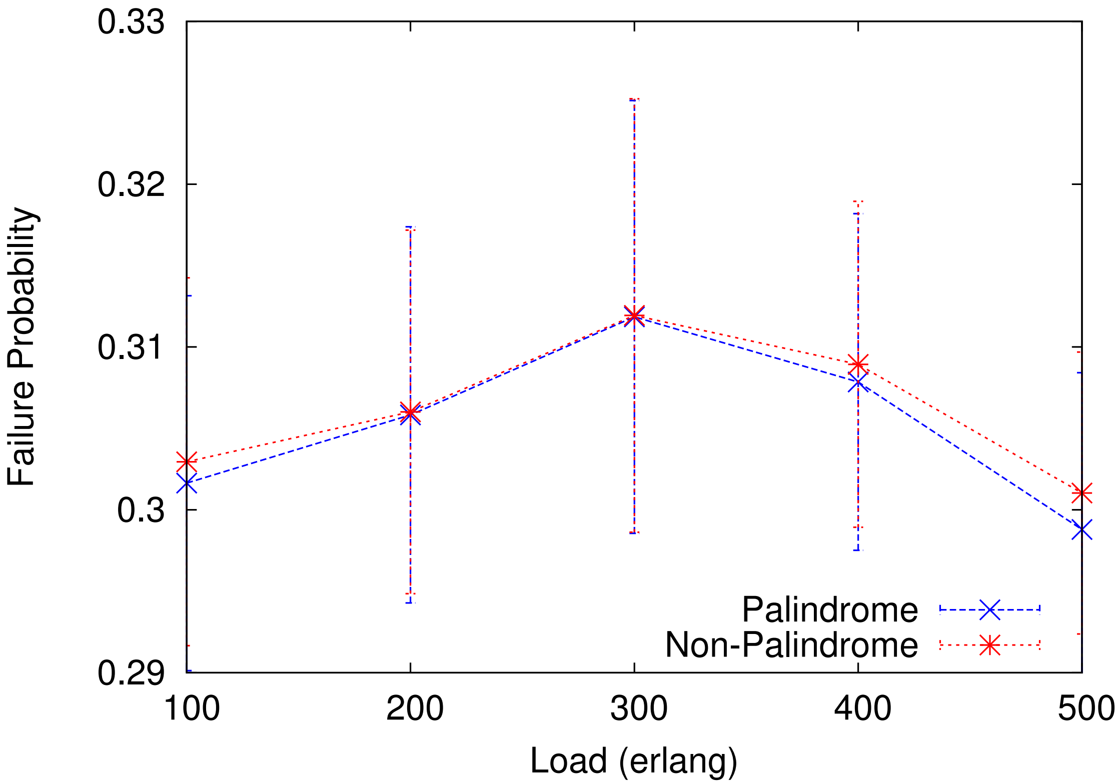


Figure 5.2. Palindrome vs non-palindrome - Failure.

Figures 5.1 and 5.2 suggests that non-palindrome does not have any advantage in terms of blocking probability if the forward and reverse bandwidths for connections are the same. The trend lines of palindrome and non-palindrome overlap during the entire period. In fact, non-palindrome performs worse for some of the load values. This is because the return paths of some of the non-palindrome connections tend to take a longer route and therefore, contribute to the blocking of future logical connection. Failure plots further bolster the consideration of palindrome because non-palindrome connections fail more often. non-palindrome introduces more number of links per connection because its return path may be different. As a result, non-palindrome increases the probability of failure of a connection.

Figures 5.1, 5.2 help in directing the future analysis of survivability to fix on palindrome. Hence, this research considers the palindromic option only.

5.2 Comparative Analysis of Unicast Survivability

This section discusses how the survivability algorithms, Iterative Dijkstra and Bhandari perform against each other in terms of blocking, failure probability and path length measured in hops.

5.2.1 Blocking Probability

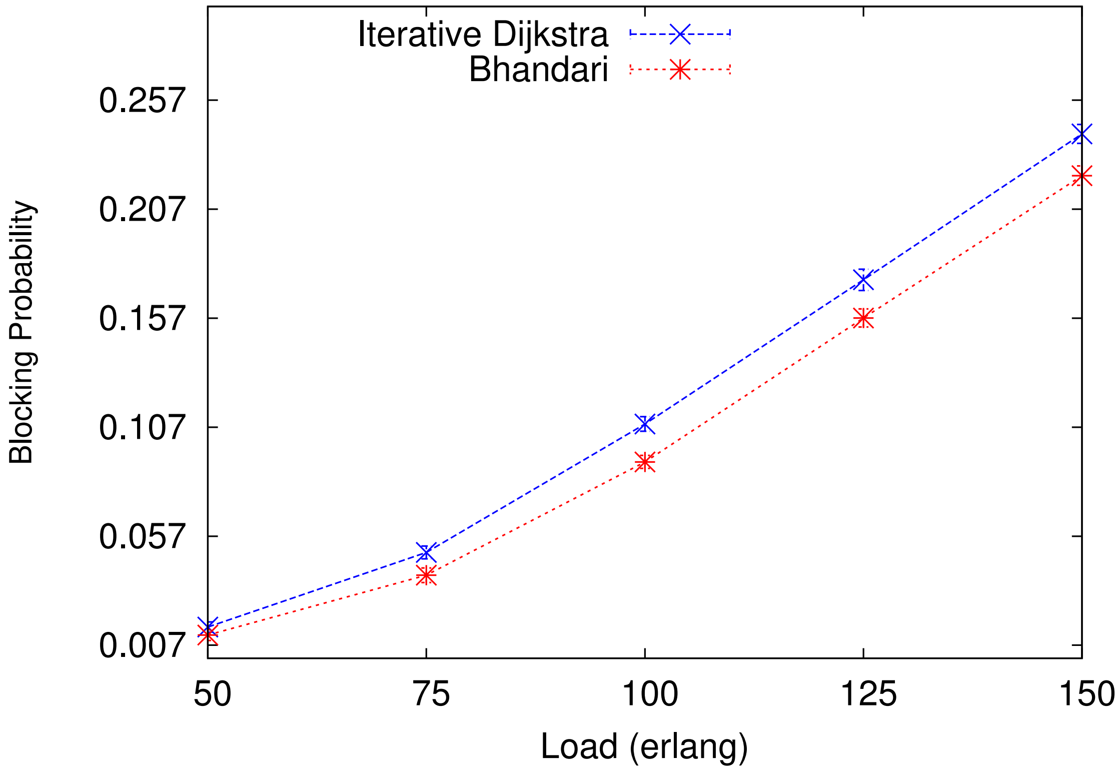


Figure 5.3. Iterative Dijkstra vs Bhandari - Blocking

As discussed in Section 2.2.1.1, iterative Dijkstra is greedy in allocating resources for the primary path and therefore, it may lead to more blocking than Bhandari. Figure 5.3 confirms that in this network, iterative Dijkstra has led to approximately up to 2% increase in blocking probability at higher loads over Bhandari. Even for the lower loads, the iterative Dijkstra performs considerably poors because it is unable to set up the backup paths. Note that the connection is successful only if both the paths (primary and backup) can be set up.

5.2.2 Resource Allocation

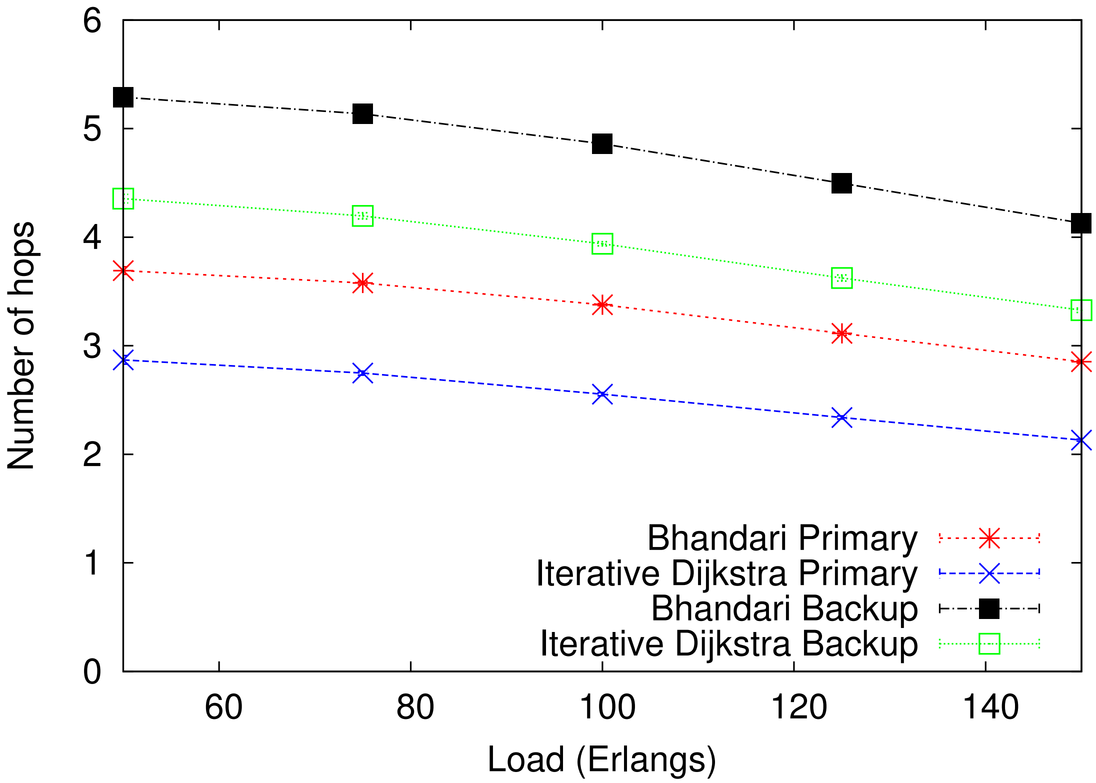


Figure 5.4. Iterative Dijkstra vs Bhandari - Hop counts

Figure 5.4 shows that average hop counts decreases by 1 for both primary and backup paths for iterative Dijkstra. This shows that the logical connections set up with Bhandari is using more resources than iterative Dijkstra even at lower loads with almost zero blocking. Also, decreased hop counts may positively approach node failures.

5.2.3 Failure Probability

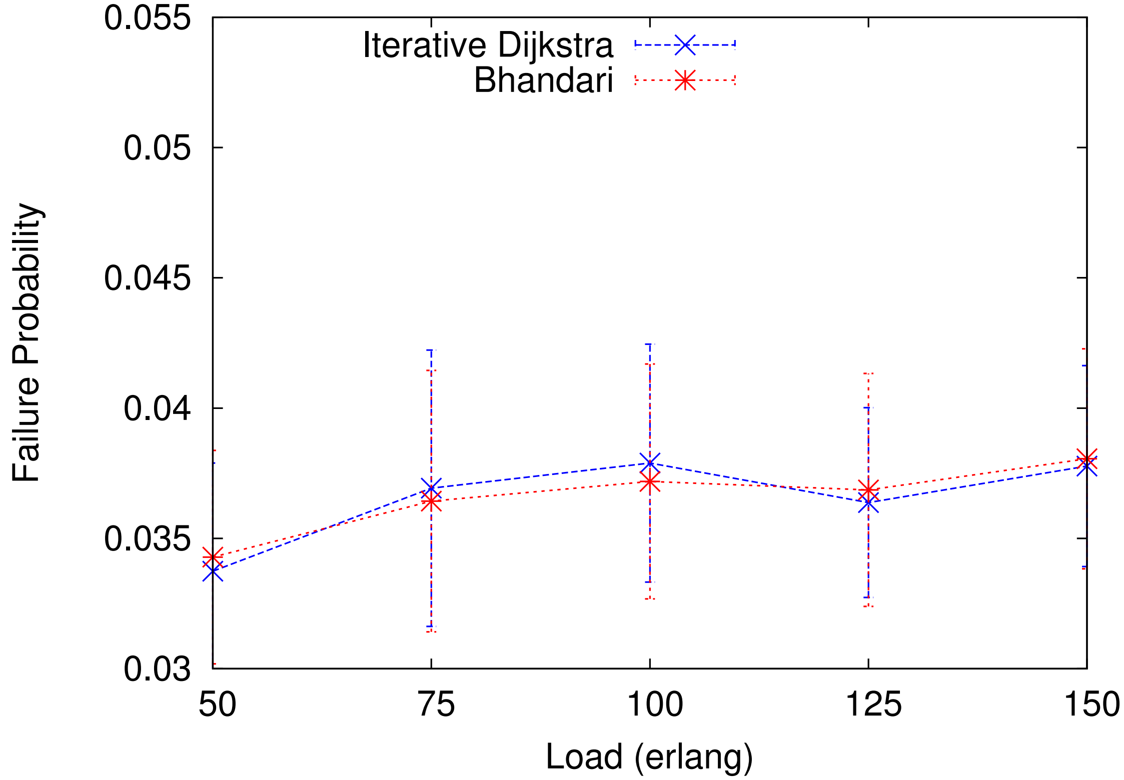


Figure 5.5. Iterative Dijkstra vs Bhandari - Failure Rate

Surprisingly in Figure 5.4, the fewer hops of iterative Dijkstra does not provide any advantage over Bhandari for link failures in Figure 5.5. Shorter paths are expected to experience fewer link failures. Iterative Dijkstra takes fewer hops to reach the destination node yet it might have taken a longer path. To explore more on this, the failure probability is calculated for individual paths.

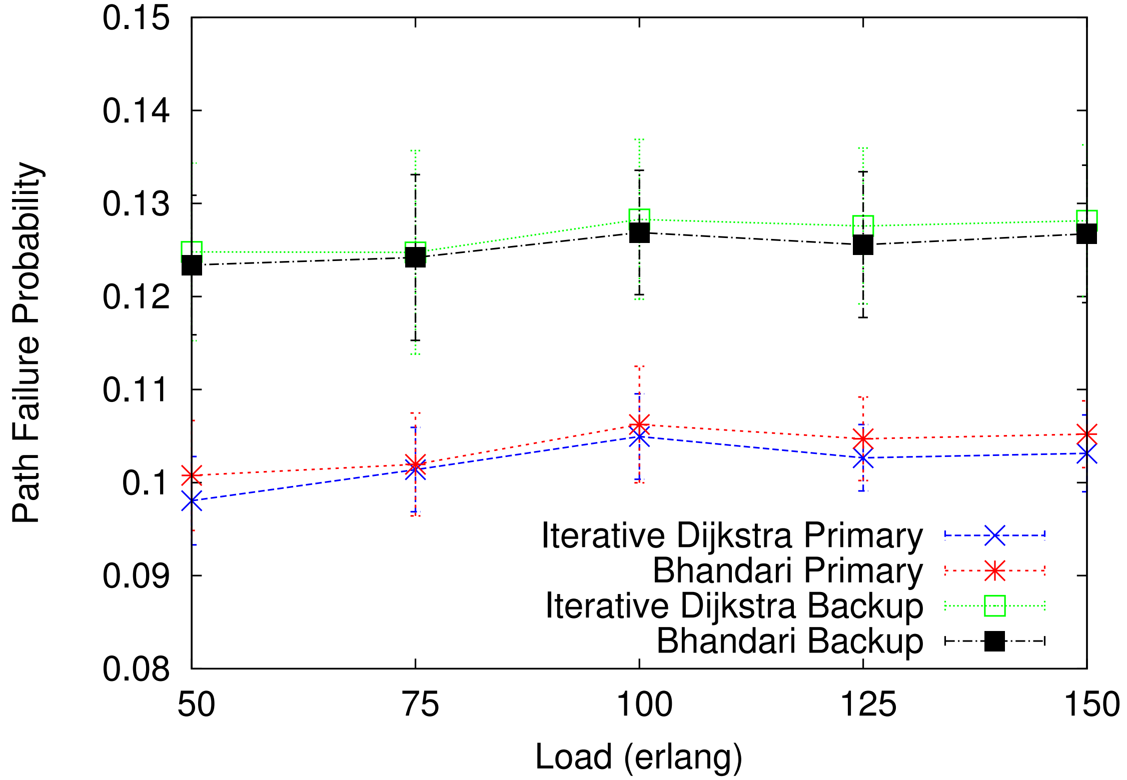


Figure 5.6. Individual path failures of Iterative Dijkstra and Bhandari

Figure 5.6 shows the primary and back up path failures for iterative Dijkstra and Bhandari. It appears in Figure 5.6 that backup paths have failed more than primary for both approaches. This implies that backup paths are actually taking the longer route in terms of cost for both iterative Dijkstra and Bhandari. Hop counts seem to acknowledge this because it is increased for back up paths of both iterative Dijkstra and Bhandari. This suggests that distance and hop counts may be directly proportional to each other. However, it is not actually true from what happens between primary or back up paths of both iterative Dijkstra and Bhandari. It is true from Figure 5.5 that the primary paths of iterative Dijkstra utilized fewer resources than the primary path of Bhandari. Accordingly, the primary paths of iterative Dijkstra experiences less failures than Bhandari. However, this does not stay

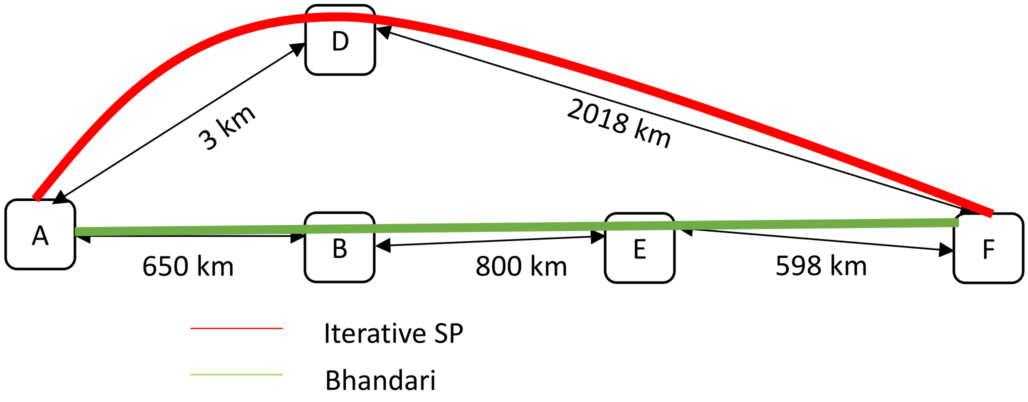


Figure 5.7. Backup paths of Iterative Dijkstra and Bhandari.

valid for backup path. The backup path of iterative Dijkstra fail more often than Bhandari even though it has lower hop counts. This suggests that the backup paths of iterative Dijkstra may have used more number of long distance links as a result of fewer hop counts. Fig. 5.7 illustrates where the iterative Dijkstra takes the path A-D-F with 2 hops and Bhandari takes the path A-B-E-F with 3 hop counts. Remember, the failure of the each link is independent and directly proportional to distance. Hence, the link of distance 2018 km fails with greater probability than the links of distance 650 km, 800 km, 598 km. That said, relative data on distance, costs and hops counts is required to realize the failure behaviors of both the approaches.

The reason for the increase in the failure of iterative Dijkstra's backup path is assumed to be because of its shortest primary path. This forces iterative Dijkstra to take longer backup path which are more prone to failures. On the other hand, Bhandari equally distributes its resources to both primary and backup path. Although backup path takes more resources then primary, its distance is not as long as that of iterative Djikstra. Thus, both the algorithms compromise on both primary and backup path failures. Thus, the failure of connections for both iterative Dijkstra and Bhandari does not differ much.

In summary, connection blocking with Iterative Dijkstra is higher compared to Bhandari . Also, failures also does not seem to favor iterative Dijkstra. Because of these reasons, Bhandari seems to be a convenient option for protection. Yet iterative Dijkstra's resource allocation is much better than Bhandari in terms of hops. Moreover, the blocking probability does not differ much at lower loads. This research proposes a modification to iterative Dijkstra in Chapter 6 to better compete with Bhandari.

CHAPTER 6
PROPOSED RISK-AWARE LINK AVOIDANCE

This chapter proposes a slight modification to iterative Dijkstra and argues that how such an approach can be periodically beneficial to the network.

6.1 Proposed Modification

The primary cause of blocking in Iterative Dijkstra is its incapability to provide resources for backup path. This can be overcome by increasing the availability of resources for backup paths. For a connection to survive in the event of link failures, this thesis provides the connection with a backup path that does not share any link with theprimary path. This is one of the primary constraints for achieving survivability. However in the proposed approach, this thesis is going to relax this constraint and allow backup paths to share some of the low-risk links with primary paths. The constraint is relaxed by streamlining the routing of both primary and backup path by only pruning the failure prone links in the primary path from the network using the link-distance threshold. Remember that the failure of each link is independent to each other and it is directly proportional to distance. The threshold is carefully chosen as specified in Section 6.1.1 such that it provides a good trade-off between blocking and failure probability.This approach will be called by the name risk-aware link avoidance. The proposed modification shows good performance for blocking at low and moderate loads by guaranteeing minimal resources. Also, the results shows a good trade-off between blocking and failure probability for the chosen threshold distance for pruning.

6.1.1 Link Pruning in Risk-aware Link Avoidance

This section explains how this research reaches a threshold value to remove links found along the primary path from the topology. The pruning of links after finding the primary path is an important decision to make in risk-aware link avoidance as it plays a major role in not just blocking but also failure of connections. By pruning only a few links from the primary path, more links can be made available to the backup but that may contribute to decrease in survivability. While risk-aware link avoidance is not completely disjoint, it may contribute to failure of connections even when there are no overlaps in failure time of one or more links. Hence, it is important to make a crucial decision on pruning. However, this decision should be based on topology considered. Remember, ESnet connects sites and institutions across the country which results in a considerable number of long distance links. Also, there are links that go to different continent as well. According to the failure assumptions, it is evident that long distance links have smaller MTBF and thus failing more often than short distance links that have greater MTBF. Hence, it is important to decide on a threshold distance which helps in filtering out the failure-prone links from the primary path.

Threshold distance (km)	**Number of links**	**Number of failures in 1 year**
$distance > 1000$	19	44
$500 < distance < 1000$	9	19
$distance < 500$	44	9

Table 6.1. Link counts and failures based on distance

Table 6.1 shows that almost one-third of the links have distance greater than 1000 km. It is evident from their MTBF that these links fail often for the simulation period of 1 year. One approach is to take average of all the link distances for threshold distance. Another approach is to manually look at the number of failures that each set of links (as per Table 6.1) undergo, in a one year period.

From Table 6.1, more than half the failures are caused by the links of distance greater than 1000 km. Although the links of distance between 500 km and 1000 km contribute to the failure, their link count is small enough to ignore. Moreover, these links may play a crucial role in routing between two states in the country. So, pruning these links may increase blocking and thus reduce this approach to iterative Dijkstra. One more optionis be to remove half of the medium distance links. However, exploring these cases is out of the scope of this research yet very interesting. This research picks the threshold value of 1000 km. Figure 6.1 explains Risk-aware routing.

6.1.2 Risk-aware Link Avoidance Routing

The steps are as follows:

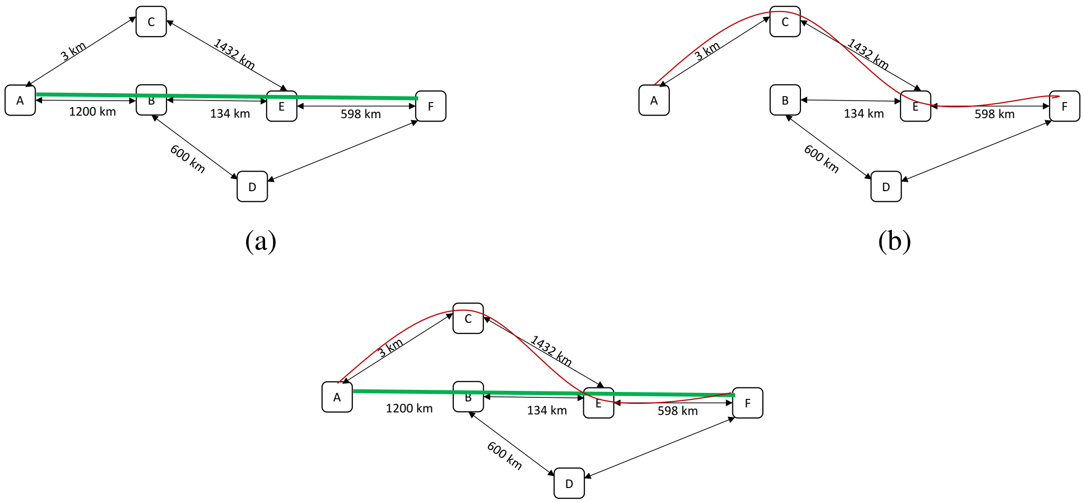
(a) Finds the shortest path with Dijkstra for primary path.

(b) Removes the links in first shortest path from the network as per Section 6.1.1 and run Dijkstra again

(c) The result is the two paths between a source and destination node.

6.1.3 Effects of Link Pruning in Risk-aware Link Avoidance

The expected increase in the failure of risk-aware link avoidance is not just due to the partially disjoint paths. The risk-aware link avoidance approach can only remove the vulnerable links in the primary path and there is no guarantee that such a link would not exist in the backup path. In Figure 6.1, the backup path still has one of the links greater than 1000 km. This scenario is common in the other algorithms as well.



(a)

(b)

(c)

Figure 6.1. Risk-aware Link Avoidance

6.2 Performance Analysis of Risk-aware Link Avoidance

This section compares the blocking and failure probabilities of proposed with the approached discussed in the Chapter 5.

6.2.1 Blocking Probability

Unsurprisingly in Figure 6.2, the blocking probability is decreased for risk-aware link avoidance compared to iterative Dijkstra. At lower loads, risk-aware link avoidance shows better performance than Bhandari. Although this decrease is very small in the beginning, it increases with the increase in the load. At moderate load, risk-aware link avoidance shows at most 3% decrease in blocking. This indicates that there are enough links for back up paths still available for risk-aware link avoidance to allocate resources greedily for the connection and continues to outperform Bhandari. However, this does not continue further as further increase in load contributes to closing this gap between Bhandari and risk-aware

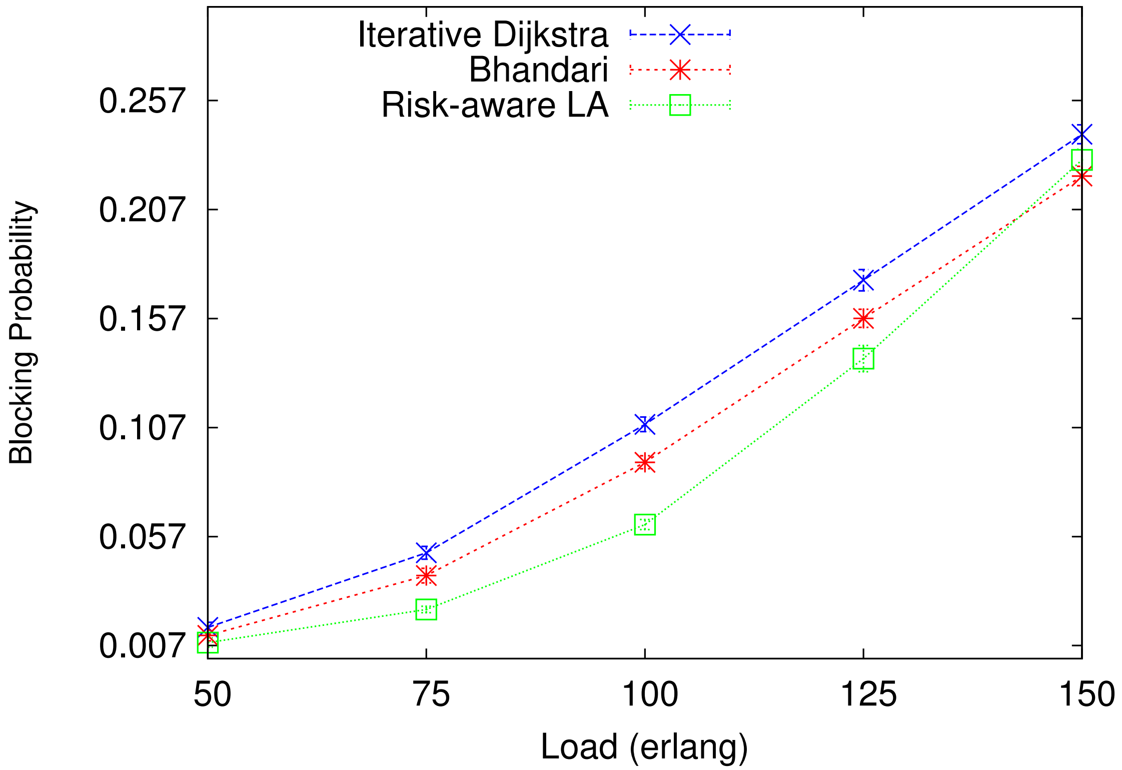


Figure 6.2. Comparative bocking probability analysis of Iterative Dijkstra, Bhandari and risk-aware link avoidance.

link avoidance. At higher load, Bhandari still provides better performance for blocking as risk-aware link avoidance pays the price for being greedy just like iterative Dijkstra.

6.2.2 Failure Probability

As expected, the failure probability for risk-aware link avoidance is greater than both iterative Dijkstra and Bhandari. The primary reason is that its not end to end disjoint. Moreover, removing the long distance links in the primary path may lead to replacement of those with other long distance links in backup path and hence makes it prone to failures. That said, the increase in failure is just 0.8 % while it can offer 3 % decrease in blocking.

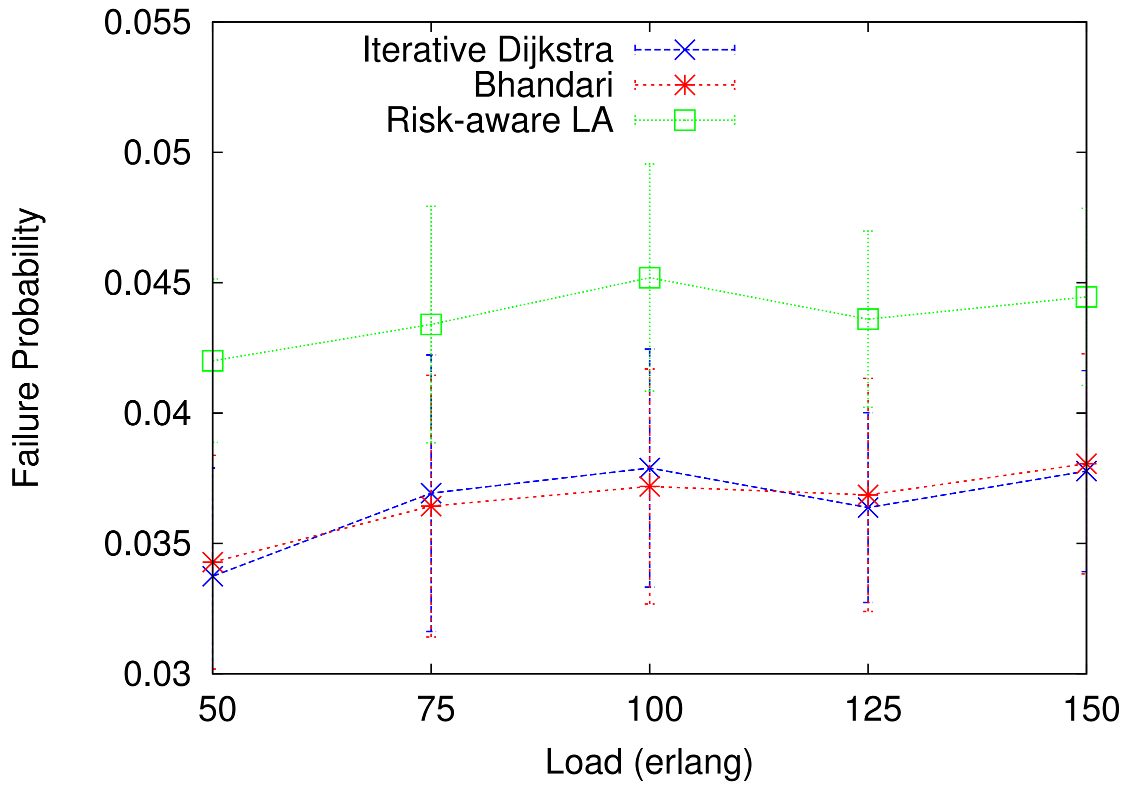


Figure 6.3. Comparative failure probability analysis of iterative Dijkstra, Bhandari and risk-aware link avoidance.

CHAPTER 7
CONCLUSION

In this thesis, we analyse three path protection algorithms, two existing, iterative Dijkstra, Bhandari and one proposed, risk-aware link avoidance in terms of its connection blocking probability, hop counts and connection failure probability. From the comparative analysis, it is found that Bhandari does better than the other two at higher loads. Bhandari uses more resources and does not differ much in terms of blocking probability than iterative Dijkstra at low loads. This thesis argues that a slightly modified iterative approach at low loads can offer less blocking with a negligible increase in circuit failure rates. Results shows that risk-aware link avoidance benefit network operations at 13% blocking for the traffic characteristics discussed in Chapter 3 and 4. Table 7.1 compares the performance of Bhandari and Iterative Dijkstra in terms of blocking probability with respect to load ranges.

From table 7.1 risk-aware link avoidance shows better performance for load values up to 140 for the considered traffic characteristics. Note that the increase in the failure for risk-aware link avoidance is constant at 0.8% in this range.

Using iterative Dijkstra and risk-aware link avoidance over Bhandari, the average resources in terms of hops is brought down by 1 for both primary and backup paths. While this decrease may not be beneficial for link failures from the analysis, it may be a crucial

Load Ranges	**Blocking Probability**
50-140	Risk-aware link avoidance < Bhandari
Above 140	Risk-aware link avoidance > Bhandari

Table 7.1. Load driven survivable approach

factor for node failures. Since Bhandari distributes the load better than others, it has high chances of connection disruption due to nodal failure[23].

This thesis considers that connections are set in advance. Hence, it is very complex to predict the future traffic behavior and apply various approaches. It may be true that the network traffic does not follow Poisson arrival and exponential holding time. That said, the approaches could still be applied to varying loads in the network depending on the holding time of the connection as it may vary from minutes to even years. For example, if the connection has short holding time requested at lower operational load, Bhandari may not be a good option. At the same time, if the holding time is longer and the operator is unsure of future traffic characteristics, Bhandari is the safest option. However, an intensive study should be conducted based on intended input traffic characteristics specific to the network in order to apply the approaches considered herein.

LITERATURE CITED

[1] A. Mokhtar and M. Azizog, "Adaptive Wavelength Routing in All-Optical Networks," *IEEE Transactions on Networking*, vol. 07, no. 02, April 1998.

[2] B. L. Xiaowen Chu, "A dynamic RWA algorithm in a wavelength-routed all-optical network with wavelength converters," *IEEE/ACM Transactions on Networking*, vol. 13, no. 03, April 2003.

[3] V. M. Vokkarane, J. Wang, X. Qi, R. Jothi, B. Raghavachari, and J. P. Jue, "Dynamic dual-homing protection in WDM mesh networks," *IEEE*, vol. 01, july 2004.

[4] B. Rajagopalan, "IP over optical networks: architectural aspects," *IEEE*, September 2000.

[5] IP routers and routing protocols. [Online]. Available: www.metaswitch.com/iprouting

[6] A. J. Gonzalez and B. E. Helvik, "Guaranteeing Service Availability in SLAs; a Study of the Risk Associated with Contract Period and Failure Process," *IEEE Latin America Transactions*, vol. 08, no. 04, August 2010.

[7] G. Andrs J and B. E. Helvik, "An Overview of Algorithms for Network Survivability," *ISRN*, no. 24, September 2012.

[8] L. Shen and R. B, "Centralized vs Distributed connection management schemes under different traffic patterns in wavelength-convertible optical networks," *IEEE International Conference on Communications*, vol. 05, September 2002.

[9] A. P. Vela, M. Ruiz, F. Fresi, N. Sambo, F. Cugini, G. Meloni, L. Pot, and L. Velasco, "BER Degradation Detection and Failure Identification in Elastic Optical Networks," *IEEE*, vol. 35, no. 21, November 2017.

[10] OSCARS v1.0 prototype architecture. [Online]. Available: https://github.com/NetLab/oscars-newtech

[11] Application Programming Interface API. [Online]. Available: www.mulesoft.com

[12] Disjoint-path algorithms. [Online]. Available: www.macfreek.nl

[13] Bellmann Ford negative edge shortest path algorithm. [Online]. Available: www.brilliant.org

[14] Ethernet switching. [Online]. Available: www.lightwaveonline.com

[15] A. Markopoulou, G. Iannaccone, C.-N. Chuah, Y. Ganjal, and S. Bhattacharyya, “Characterization of Failures in an Operational IP Backbone Network,” *IEEE/ACM*, vol. 36, no. 4, August 2008.

[16] Weibull distribution. [Online]. Available: www.lightwaveonline.com

[17] G. Iannaccone, C. nee Chuah, R. Mortier, S. Bhattacharyya, and C. Diot, “Analysis of link failures in an IP backbone,” *IEEE Workshops of International Conference on Advanced Information*, 2011.

[18] SofieVerbruggel, DidierColel, PietDemeester, RalfHuelsermann, and MonikaJaeger, “General Availability Model for Multilayer Transport Networks,” 2005.

[19] A. Jurdana, B. Mikac, and G. Kreso, “Heuristic Approach to Availability Calculation of Path Protected Optical Network Based on the Analysis of Cable Failures,” *IEEE*, 2011.

[20] Gamma distribution. [Online]. Available: www.itl.nist.gov

[21] M. Danko, M. Furdek, G. Zervas, and D. Simeonidou, “Evaluating Availability of Optical Networks Based on Self-Healing Network Function Programmable ROADMs,” *IEEE Optical and Communication Networks*, vol. 06, November 2014.

[22] H. Ma, D. Fayek, and P.-H. Ho, “Availability-Constrained Multipath Protection in Backbone Networks with Double-Link Failure,” *IEEE International Conference on Communications*, vol. 06, 2008.

[23] V. Miletic, D. Maniadakis, B. Mikac, and D. Varoutas, “On the Influence of the Underlying Network Topology on Optical Telecommunication Network Availability under Shared Risk Link Group Failures,” *IEEE International Conference on the Design of Reliable Communication*, 2014.

[24] R. B. R.Lourenco and D. A. A. Mello, “On the Exponential Assumption for the Time-to-Repair in Optical Network Availability Analysis,” *IEEE/ICTON Optical and Communication Networks*, July 20132.

[25] B. Xia, C. Yang, and T. Cao, “Modeling and Analysis for Cache-Enabled Networks with Dynamic Traffic,” *IEEE/ICTON Optical and Communication Networks*, vol. 20, July 2016.

[26] K. Gaizi, F. Abdi, and F. M. Abbou, “Realistic dynamic traffic generation for WDM Optical Networks,” *IEEE Optical and Communication Networks*, June 2016.

[27] J. M. Plante, D. A. P. Davis, and V. M. Vokkarane, “Parallel and survivable multipath circuit provisioning in ESnet’s OSCARS,” *Photon Network Communications*, June 2015.