University of Alberta

Methods and Issues in Path Provisioning on Ring-based Networks

by

Ernest Siu ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science.

Department of Electrical and Computer Engineering

Edmonton, Alberta

Spring, 2000

0-612-60177-3

University of Alberta

Library Release Form

Name of Author: Ernest Siu

Title of Thesis: Methods and Issues in Path Provisioning on Ring-based Networks

Degree: Master of Science

Year this Degree Granted: 2000

Ernest Siu
11560 – 14 Avenue
Edmonton, Alberta
T6J 7B3

Date: Dec 23, 1999

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Methods and Issues in Path Provisioning on Ring-based Networks** by **Ernest Siu** in partial fulfillment of the requirements for the degree of **Master of Science**.

W.D. Grover

B.F. Cockburn

J.J. Harms

Date: Dec 17/99

Abstract

This research effort is driven by the growing interest of network operators in the ability to analyze the availability and cost of an end-to-end path construction in their networks, and to enable the operators to provide various levels of assured service availability to customers while keeping the cost to a minimum. The specific area of research is survivable ring networks (SONET bi-directional line switched ring) and, in particular, the use of matched-nodes (MN) and dual feeding (DF) configurations at inter-ring connections. The major focus of this thesis is to find a way to take advantage of the combined use of MN and DF, which have different costs and availability properties under different circumstances. Path search strategies are proposed and implemented in a software prototype. Test cases results based on various requirements of cost and/or availability (with constraints) are presented and analyzed.

To my loving parents Emily and William, and my beloved Wendy

Acknowledgments

First I would like to thank the Department of Computing Science in the University of Alberta, which gave me exposure to the field of telecommunications. I thank Matthieu Clouqueur and Dave Morley for their innovative and valuable input to the research problem. I thank Robert Novak, Sing Cheng, Heng Foo, Kay Wee Ang, Chee Yoon Lee, John Doucette, SC and AOE for technical input and the endless number of nights working late in the lab together. I thank TR*Labs* for giving financial support and providing me a great environment to work on my research. I also thank all the staff at TR*Labs* and the ECE Department at the University of Alberta for providing great support in very many ways. I thank Luke Chong for his exceptional technical support.

I especially thank Dr. Wayne D. Grover for bringing me into the research area of network planning, and I also thank him for his guidance and innovations which made this research effort a success.

Last but not least, I would like to thank my parents and my girlfriend Wendy for their continuous support and understanding throughout my graduate studies.

Table of Contents

List of Tables

List of Figures

List of Symbols

Td_1	Formula for calculating single fed path unavailability
Td_2	Formula for calculating dual fed path unavailability
U_{sl}	Unavailability value for 1km of span
U_{nl}	Unavailability value for a network element
U_{na}	Unavailability value for a add/drop port card
U_p	Unavailability value for a path segment
U_{mn}	Unavailability value for matched-nodes only path
U_{df}	Unavailability value for dual feeding only path
$U_{mn/df}$	Unavailability value for MN/DF combined path

List of Abbreviations

SF	Single feeding inter-ring connection configuration
MN	Matched-nodes inter-ring connection configuration.
DF	Dual feeding inter-ring connection configuration.
BLSR	Bi-directional line switched ring.
SONET	Synchronous Optical Network
OC	Optical Carrier
STS	Synchronous Transport Signal
VT	Virtual Tributary
DS	Digital Signal Level
U	Unavailability
A	Availability
RS	Ring Sequence
PC	Path Construction
RS-R	Ring Sequence Reduction
PC-R	Path Construction Reduction
RS/PC-R	Ring Sequence and Path Construction Reduction combined
O-D	Origin and destination
SLA	Service Level Agreement
OPTICA	Optimal Path Provisioning Tool Integrating Cost and Availability
RBI	RingBuilder Interactive
MTBF	Mean Time Between Failures
MTTR	Mean Time To Repair

1. Introduction

1.1. Background

The availability of telecommunication systems has become more and more important as different sectors of society increasingly utilize communication services such as voice, data, video and Internet traffic. As businesses and the general public become more dependent on these services, the availability requirement grows to an extent that is similar or in excess of that of the other utility services. Moreover, different customers have different requirements in terms of the cost and availability of their telecommunication services.

Nowadays carrying high volume traffic requires fiber optics technology and, specifically in the current generation of transport, a SONET transmission network, which has a capacity as high as 10 Gbps (OC-192) per fiber. Such high capacity links carry a large volume of traffic; for example, one OC-192 link carries 156,000+ voice circuits or their equivalent if rated at 64 kbps each. The unavailability of such a link may have a drastic impact on business and the community [1,2]. In the emerging field of deploying and providing high-speed transport network services, network operators are most interested in the ability to analyze the cost and availability of a path, that is, an end-to-end connection. In realizing this ability, service providers will be able to increase their confidence in assuring various availability levels of services to customers, while maintaining cost/effectiveness by utilizing the existing network.

1.2. Research Goals

The following are the major objectives of this thesis research:

1. Analyze the cost-effectiveness and availability of various end-to-end path constructions realized through ring-based networks.
2. Design search strategies for end-to-end path construction so as to satisfy desired levels of cost and/or availability.
3. Develop and implement a software prototype for path provisioning and support of the path search strategies designed in objective 2.

1.3. Thesis Outline

This thesis is in four main parts: the introduction and background information (Chapters 1 to 3); problem definition and characterization (Chapters 4 and 5); research results and analysis (Chapters 6 to 8); and the conclusions (Chapter 9).

In the first part, specifically in Chapter 1, the research topic is introduced and some related developments in this area are presented. The key ideas of network integrity are reviewed in Chapter 2. In Chapter 3, the SONET network, its ring architecture, the nature of an end-to-end path on a ring network, and their cost and availability properties are presented.

In the second part, the path design or routing problem is presented using graph notation in Chapter 4, and a preliminary analysis on the problem is performed. In Chapter 5, the problem is further characterized in terms of graph theory and practical approaches to solve it are proposed. The complexity of the problem is also analyzed, and that assists in designing strategies and algorithms that are efficient in terms of time and space.

In the third part, the proposed end-to-end path search strategies (heuristics) for single objective optimization (with or without constraints) are presented in terms of cost or availability in Chapter 6. In Chapter 7, the tools and environment during the development of the software prototype are described, which implements the search strategies presented in Chapter 6. Then the software architecture is briefly described, and information given as to how to utilize the software prototype as an end user. In Chapter 8, some sample simulation results and analysis are presented, to validate the optimality of certain cases and to illustrate the efficiency of the software prototype and the search strategies.

In the fourth part, the thesis is concluded by summarizing the main results and findings of this research effort. Future research topics in the area of path provisioning on ring-based networks are also presented.

2. Network Integrity

2.1. Background

Public communication facilities play a critical role in society, and disruptions in service have a severe impact on the community and businesses [1,2]. Public network integrity can be defined as the ability of a network provider to deliver high quality, continuous service with minimal customer impact from network element failure [1]. Network failure can be due to various problems including architectural defects, design defects, inadequate maintenance procedures, or procedural error. In addition, acts of God (lightning, hurricanes, earthquakes), accidents (auto crashes, derailments) or sabotage (hackers) may also result in network failure. Techniques such as duplication, protection switching, standby power and alternate routing are traditionally used throughout a network to enhance its integrity [4]. There are also several ways to measure network integrity, namely, restorability, reliability, and availability [3]. The following three subsections will discuss each in detail, and their relevance will be compared briefly in the last subsection.

2.2. Restorability

Restorability describes a system's ability to restore itself given that a certain postulated failure has occurred. More precisely, it is the fraction of the transmitted signal units which are severed in a failure (e.g. fiber/wire cut or equipment failure) and that are subsequently restored within a target time, with some network restoration / re-routing mechanism.

For a network architectures like a bi-directional line switched rings (with 100% protection bandwidth) or 1-for-1 dedicated protection (1 working channel and 1 protection channel), they have 100% restorability upon single element failure. Restorability may not be too informative for such dedicated protection architecture. On the other hand, restorability is a measurement highly meaningful when applied to mesh restorable networks (see section 3.2.1). It quantifies the fraction of all end-to-end paths that are essentially uninterrupted under all possible single element failures. Since restorability is associated with a maximum restoration time limit, the impact on customers is at least bounded or limited, unlike the use of availability alone. On the other hand, it does not reflect the amount of traffic being affected (for example, the affected traffic may actually be lightly loaded in the middle of the night [4]) or the probability of the failure event itself.

2.3. Reliability

Reliability is the probability that a system or device will perform its function without failure for a given period of time [3]. Because there is a posted time limit (as opposed to availability, which is intended to measure long-term system integrity) associated with this measurement, it is usually used in mission-oriented engineering activities like the launching of the Space Shuttle or a 747 flight. Since there is no opportunity to do repairs during the mission, if a critical element or subsystem fails, the whole mission might fail. Reliability is strictly always a probability that is decreasing function of time, describing the probability a system or component will work for a specific amount of time without a service-affecting failure.

2.4. Availability

Reliability is described above to distinguish it from availability, which seems similar but is actually different in the nature of the problem domain. Availability is the measurement that is used mainly in modeling the integrity of a service or system in telecommunication. Availability is used to model a *continuously maintained repairable system* [3], and reflects the probability that a system is operating satisfactorily at any randomly chosen time after the start of the operation assuming an ongoing equilibrium between failure events and subsequent repair events. Therefore, the expression for availability is:

$$A = \frac{MTBF}{MTBF + MTTR}$$

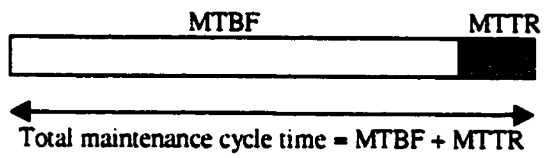


Figure 2.1: MTBF, MTTR and Availability

Where MTBF = mean time between failures and MTTR = mean time to repair. On the other hand, one may also see this in terms of :

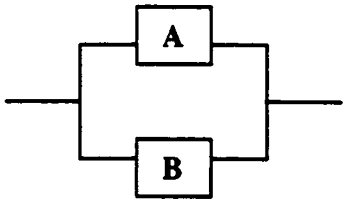
$$\lambda = failure\,rate \quad (MTBF^{-1})$$
$$\mu = repair\,rate \quad (MTTR^{-1})$$

In which $Availability = \frac{\mu}{\lambda + \mu}$

A key idea in calculating availability is that adding up all the unavailability scenarios and using (1-U), where U denotes unavailability, to calculate availability is easier than multiplying all availability elements. In many cases in telecom research, the following approach is taken:

$$Availability = (1 - Unavailability) \quad \text{In which} \quad Unavailability = \frac{\lambda}{\lambda + \mu}$$

The following example illustrates these ideas. Consider a 1-for-1 redundant system (a working unit and an identical hot standby system):



Parallel Availability Relationship

Figure 2.2: Parallel System

Assume that each unit A, B individually undergoes failures and repair with resulting unavailability U. The following is an exact expression for the overall availability:

$$A = P(A+B) = P(A) + P(B) - P(A) * P(B)$$

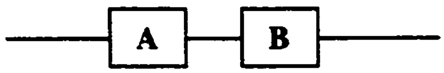
OR

$$\begin{aligned} A &= 2(1-U) - (1-U)^2 \\ &= 2(1-U) - (1-2U+U^2) \\ &= 1-U^2 \end{aligned}$$

This is equivalent to, but much simpler in form than:

$$A = \frac{2\lambda\mu + \mu^2}{(\lambda + \mu)^2}$$

In a series availability relationship between elements, both elements must work for the system to function:



Series Availability Relationship

Figure 2.3: Serial System

The exact expression for availability is:

$$A = P(A \cdot B) = P(A) \cdot P(B) = (1 - U)^2$$

OR $$A = 1 - 2U + U^2$$

Which is very often numerically approximated by:

$$A = 1 - 2U$$

This is quite a useful and valid approximation for a wide range of cases encountered in practice where U for individual network elements is 10^{-4} or less.

Although it is widely recognized as the quantitative assessment of service availability for telecommunication, it does not adequately reflect the real impact and perceived importance of failures to the users. For example, although two end-to-end paths may have the same 30 seconds outage per year, the impact of the path with 30 one-second outages is different from the path with one 30-seconds outage. Furthermore, the impact of an outage that occurs during business hours is different from an outage occurs in the middle of the night. Therefore, we should recognize the limitations of the availability measure while we are using it for network integrity measurement.

The availability figure calculated using failure mode analysis in our study should be understood as a theoretical figure of merit of its inherent design integrity. An availability number in this context is not meant to be a prediction of specific performance in the future for any one individual path realization. The availability figure is only the limiting average over an infinite observation time. In reality, the period of subscription of telecommunication service will be at most in year's term. Section 9.2.3 will mention further research on this issue.

2.5. Relevance to the Research

The various measurements for system integrity have been developed for different problem domains. Restorability is most relevant for measuring a system as a whole as it describes the system's ability to restore itself once a failure has occurred. Reliability is good for measuring mission-oriented engineering activities. Availability is the most relevant measurement for this study on cost-effectiveness and customer perspective of the quality of an end-to-end communication paths. This is because end-to-end paths are the actual services that customers receive, and the perception of good quality service is based on the readiness of that service being available. End users do not usually care about the network as a whole as long as the particular services they subscribe to are unaffected. In addition, some service providers discount payment or do not even charge the customer if the service fails to meet the predetermined availability requirement. Further explanation for using availability analysis instead of restorability or reliability is mentioned in [4].

3. Ring-Based Transport Networks and Path Provisioning

3.1. Synchronous Optical Network (SONET)

Synchronous Optical Network (SONET) is a standard for optical signal transmission formats and physical layer interconnect set by the Exchange Carriers Standards Association (ECSA) for the American National Standards Institute (ANSI). This standard is also incorporated into the Synchronous Digital Hierarchy recommendations of the International Telecommunications Union (ITU) [6]. SONET is the transport technology for metropolitan and long-haul networks. Regional Bell Operating Companies (RBOCs) and long distance carriers are major service providers which use the SONET transport network.

SONET defines the optical carrier (OC) level and the electrical equivalent in the form of synchronous transport signals (STS). The actual line rates for various levels of OC and STS signals are described in [6]. A simple point-to-point SONET transmission system is illustrated in Figure 3.1. It consists of two network elements, with fiber optic cables between them. A network element is usually referred to as a **node**, and the fiber link between two nodes including any line regenerators is referred to as a **span** regardless of the number of physical fibers. Any type of traffic, such as voice, data and video, can be accepted by various types of service adapters, which map signals into various payload envelopes such as STS-1, STS-3c or various virtual tributaries (VT) payload envelopes. The actual frame formats used by the OC and STS formats and the various VT envelopes are described in [6]. We have to distinguish between line level signal and tributary signal in a SONET environment. The line signal describes the bulk signal level between nodes, for example for OC-48, which is 2.4 Gbps. Each node works like a multiplexer that can multiplex certain numbers of lower rate signal, i.e. tributary level signals. For example, an OC-48 node can take up to 48 STS-1 tributaries at 51.84 Mbps, or take four OC-12 tributaries at 622.08 Mbps.

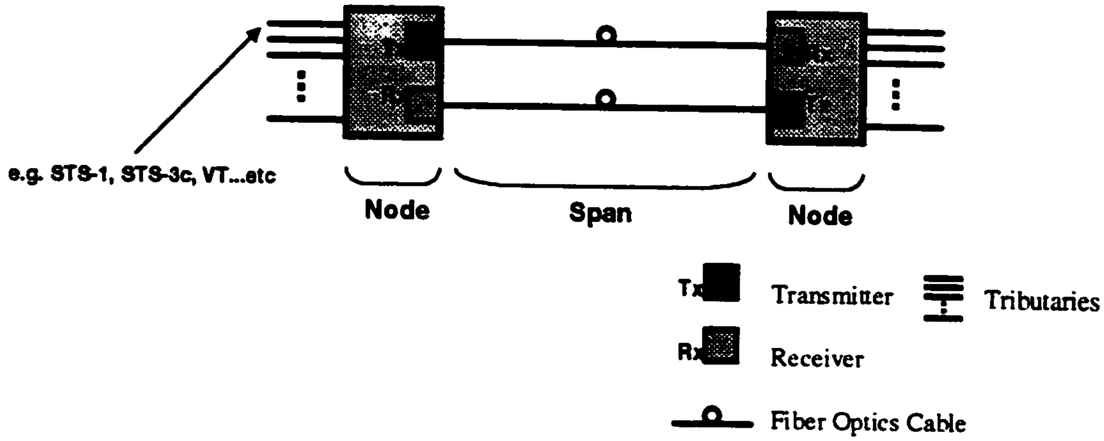
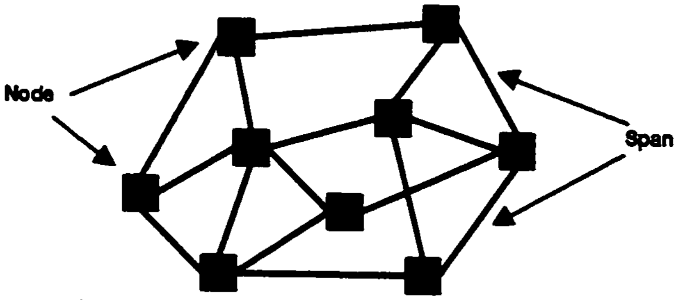


Figure 3.1: SONET Point-to-point System

3.2. SONET Protection Switching Architecture

3.2.1 Mesh Architecture

There are various architectures of SONET equipment that can be deployed so as to automatically protect the transmission from any single equipment failure or span cut. A basic system, like a 1+1 diversely routed or a 1 for N system [6,7], can manage a point-to-point service demand as in Figure 3.1. The 1:1 protection system is the simplest protection arrangement, to protect against single element failures. The system is set up as one fully working channel, and a hot stand-by channel that is available to use. In case the working channel fails, the system will be able to switch to the hot stand-by channel to replace the working channel, with minimal impact to the service.



For more complex topology, a mesh network (figure 3.2) can be constructed which would require cross-connects at each node to support automatic protection ability.

Figure 3.2: SONET Mesh Network

The mesh topology is essentially a set of point-to-point links between buildings (nodes). Network survivability is an inherent part of the mesh topology because there are often at least two paths between two nodes. Thus, a network that uses a mesh topology can often survive a single failure. Earlier techniques use a digital cross-connect (DCS) to reroute demands around a failure point using a centralized approach (i.e. a new route for rerouting is determined by a centralized system). As we approach an all optical networks, an optical cross-connects or optical switches will be in place using a distributed rerouting mechanism for faster switching time and lower cost (without optical-electrical-optical conversion). Mesh restoration does not require separate protection facilities dedicated to working systems for restoration. Instead, it uses distributed spare capacity which are being shared within working systems to restore affected demands.

For example in figure 3.3, assume there are 5 units of capacity flowing from node A to node B using the span as shown. Upon a fiber cut on this span, the capacity can be rerouted using 2 units of spare capacity on route A-X-B, and 3 units of spare capacity A-Y-B.

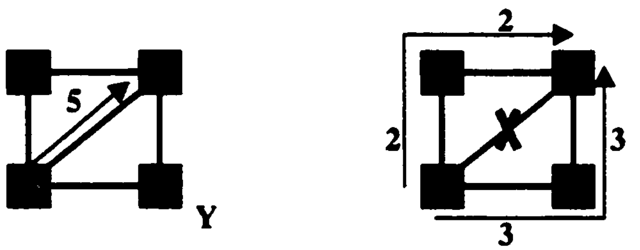


Figure 3.3: Mesh Restoration Example

3.2.2 Ring Architecture

In contrast, although ring systems may not be as capacity-efficiency as the mesh architecture, they have a relatively simple protection scheme for traffic restoration. Each ring reconfigures itself independently in a simple protection switch-like reaction upon a fiber cut [7]. The nodal equipment used (add-drop multiplexer: ADM) is also relatively less expensive than cross-connect systems. Therefore rings, particularly SONET bi-directional line switched rings (BLSR) which are the common standard of synchronous transport technology in North America, tend to predominate in metropolitan and some long-haul network applications today. Another configuration called the uni-directional path switched ring (UPSR), which deploys a slightly different protection mechanism, is not as popular. There is also a slight difference between 2-fiber and 4-fiber BLSRs in terms of their protection schemes. Both issues are addressed in [4,7].

Before describing the BLSR architecture, we have to take a look at the detail of a node in a ring. The following figure shows a node (i.e. ADM) in a BLSR. Each node is connected both west-bound and east-bound with two fibers of each direction. Transmitters and receivers are associated with the fibers. They are responsible for transmitting the line level signal. For example, for an OC-48 system, the transmission will be at 2.4 Gbps at line level for each direction. At each ADM site (i.e. node), traffic can be added or dropped upon user configuration. The add/drop traffic is at the tributary level, as described in the previous section. For example, for each OC-48 line signal, it can add/drop up to forty-eight DS-3 channels, or four OC-12 channels. The add/drop of tributary traffic instead of the total line level traffic gives flexibility to traffic management.

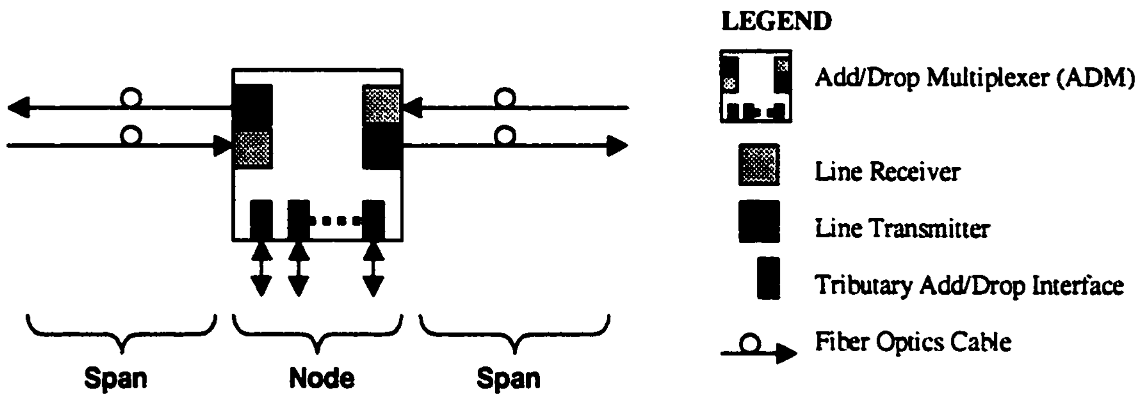
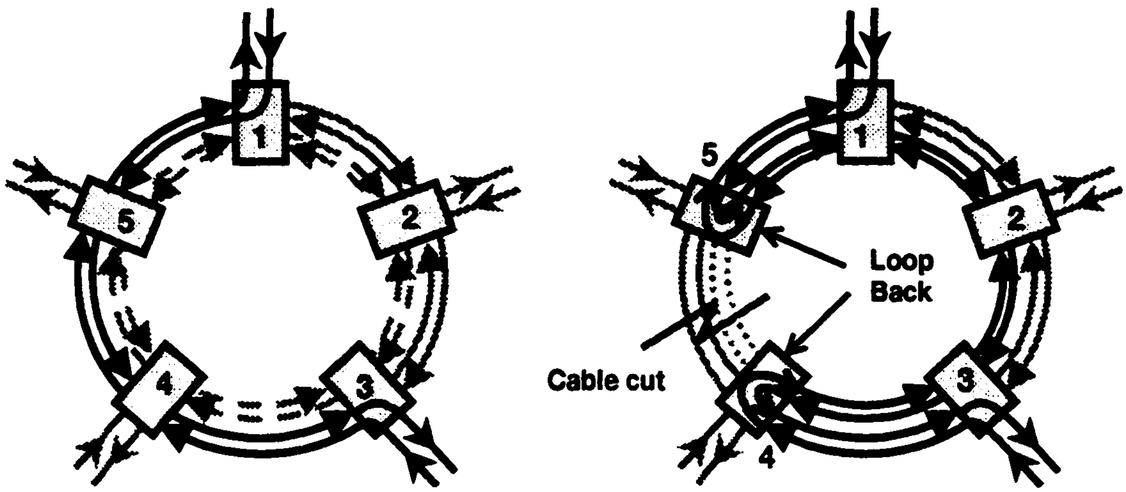


Figure 3.4: Add/drop Multiplexer and its Components

The following figure shows 4-fiber BLSR signal flows in case (a), and illustrates the automatic protection switching at work upon a fiber cut in case (b).



(a) Normal Operation (before failure) (b) Protection Operation (after failure)

Figure 3.5: BLSR Signal Flows and Protection Switching

For a SONET BLSR, bandwidth is divided into two parts: working bandwidth (2 solid lines around the ring) and protection bandwidth (2 dotted lines around the ring). During normal operation, a path is provisioned to flow from node 1 to node 3 through node 5 and then node 4. Upon a fiber cable cut between node 5 and node 4, the signal of the path at node 5 is looped back in the reverse direction on the protection bandwidth. The protected signal reaches node 4 and is looped back into the original working bandwidth. Finally, the signal arrives at node 3 on the working bandwidth.

Before going on with our discussion, we have to define a number of commonly used terms in the context of transport network:

- Point-of-presense (POP) – is a physical location (i.e. central office) in which a service provider (e.g. telco) can place and operate their network equipment.
- Graph – can be inter-changeably used with 'network topology graph', which is the underlying physical topology in which a *node* is a POP and a *span* between two nodes is an existing fiber cable.
- Add/drop multiplexer (ADM) – is the equipment that is used on a ring system in which traffic can be added or dropped. An ADM is placed in a POP. ADM can be referred as a *node* in the context of a ring.

- Ring – is the collection of ADMs and spans that together form a closed cycle. In other words, a ring is a series of ADMs which are placed in POPs, linked by spans forming a cycle.
- Ring set – is the set of rings that composes a ring network.
- Ring connectivity graph (RCG) – is a graph representation that describes the connectivity of the rings in the ring network.
- RCG node – is a representation of a ring in a RCG.

3.3. Related Research and Development in Transport Network

3.3.1 Ring Network Design Studies

Based on the network fiber topology and the usage demand pattern, rings are placed onto the network to cover all possible demands. The process of ring design involves demand routing, ring (or cycle) placement, and post-design refinement. The authors of [21] propose a heuristic procedure based on the trade-off between capacity efficiency and traffic capture efficiency of ring sets to find a minimum cost design. Depending on the relative cost of transmission capacity and inter-ring connection interface, test cases show that the designs generated using their procedure range from nearly optimal capacity efficiency to nearly optimal traffic capture.

Other studies focus on the ring placement portion of the process. [22] describes a lower bounding procedure and heuristic for determining a least cost cover of an undirected graph using cycles. It should be said that the ring placement problem has been proven to be NP-hard [23].

3.3.2 Mesh Network Design Studies

A mesh network has very good bandwidth utilization because the routing of each path can use a direct path (e.g. shortest path), and need not go through a sequence of rings as in a ring network. In addition, mesh network bandwidth usage efficiency is better than for dedicated protection or ring networks, because the one unit of protection bandwidth can be shared by more than one unit of working bandwidth. For example, in figure 3.6, the one unit of spare capacity of span AB can be used as protection bandwidth upon a fiber cut on path A-X-B or path A-Y-B, both with one unit of working capacity.

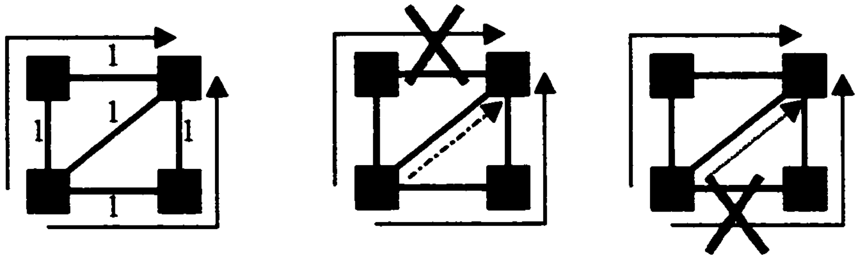


Figure 3.6: Spare Capacity Usage on Mesh Network

Also, if there are 5 units of working channels on A-B and the fiber got cut, then a mesh restorable network can reroute 2 of the working channels through A-X-B and 3 units of the working channels through A-Y-B as shown in figure 3.3.

However, optimally placing spare capacity is a complex problem. First, spare capacities on any span may contribute to the restorability of any of every other span. Second, the degree of protection any span *X* contributes to any other span *Y* depends on sparing of all other spans. A heuristic approach is proposed to obtain approximate solutions in [24] and [26], and a linear programming approach is proposed to gain an optimal solutions in a small network [25].

3.3.3 Ring and Mesh Availability Studies

Several references in the literature discussed transport network availability and survivability in various contexts. [31] is a general overview of availability and survivability for transport networks, in terms of service impact, software/hardware redundancy and defense, mechanisms and classifications for intra- and inter-ring protection, line and path protection for mesh network, and centralized vs. distributed restoration. It does not speak of any topic in specific terms but it provides a good general introduction to transport network integrity. [29] discussed general issues in evaluation of an end-to-end survivable network with proposed criteria and tools that are implemented in operational environment (i.e. telcos). It also discusses restoration mechanisms for mesh, ring and protection at the ATM level. [30] describe transport survivability on various layers (e.g. transport, access and switch) and the related issues in telephone circuits routing.

On the other hand, for end-to-end availability analysis of a path in a ring network, only a number of articles specifically addressed this issue. To and Nuesy [11] considered the availability of SONET and DS3 point-to-point systems as well as certain digital cross-connect restoration and end-to-end diverse routing over a hypothetical ladder-like network. They also derived results for 2-fiber and 4-fiber BLSRs, but did not consider the interconnection issues in a chain of rings. And so did chapter 6 in [27], which only proposed a method to calculate intra-ring availability of a path segment. [28] has considered a path through a chain of rings, however, it only considered rings with single node connectivity (two rings with only 1 common node available for inter-ring connection). We believe that there has been no other previous work specifically in cost/availability-based path provisioning in survivable ring networks. Grover in [4,5] specifically addressed this issue with all the mathematical models of availability for single-feeding, matched-nodes and dual feeding inter-ring connection configurations. This work forms the basis of this thesis research, and on top of that, we introduce the use of combined matched-nodes and dual feeding for dual redundancy at inter-ring connection, which are more cost and availability-optimized than using matched-nodes or dual feeding alone.

3.4. Inter-Ring Connection Configurations

In this section, we start to consider paths that pass through more than one ring to realize their end-to-end routing. The mechanisms required to interconnect two rings are single feeding, matched-nodes and dual feeding. In this section, each configuration will be described in detail, and then end-to-end path provisioning on a BLSR based network is characterized.

3.4.1 Single Feeding (SF)

In an SF configuration, each inter-ring connection point has at least one physical link between the two rings. Therefore, each ring must have at least one add-drop multiplexer (ADM) present for every inter-ring connection. These two ADMs must be present at a common physical site which we will call a point-of-presence (POP) (e.g. toll building, C.O.). These two ADMs are referred to as gateways. A "cross-office wiring" connection is used to connect the two gateways through add/drop cards. Cross-office "wiring" is the connection between two rings within the same Central Office. This inter-ring connection is usually made on tributary level (lower rate than on the ring level). For example, for two OC-48 rings (at 2.488 Gbps), a possible inter-ring connection would be an OC-3 (150 Mbps) channel dropped at a node of the first OC-48 ring. By using a fiber connecting to a node of the second OC-48 ring. And this is called cross-office wiring. Note that coax cable connection is also possible, for example, for DS-3 rate traffic (45 Mbps), then coax cable is being used.

This configuration will fail upon any single inter-ring element failure, that is, when one of the inter-ring gateways fails (ADM or add/drop card), or when the cross-office wire is cut, there will be outages on all service paths that go through this single feeding link. Note that it is also possible for the whole POP to be physically destroyed (e.g. toll building on fire).

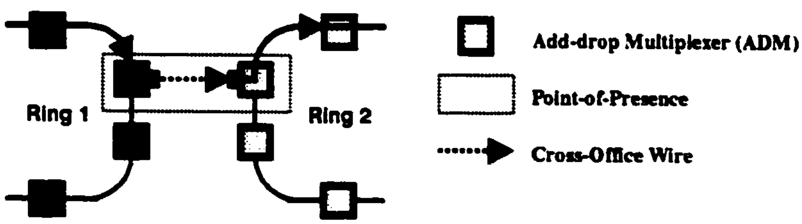


Figure 3.7: Inter-Ring Connection Configuration: Single Feeding

Although the probability of an ADM or cross-office wire failure is rare, having a single point of failure at an *inter*-ring transfer point along an end-to-end path is extremely undesirable. In contrast, all *intra*-ring signals (i.e. signals within a ring) are protected from any single element failure as mentioned in section 3.2. The single point of failure may dramatically reduce the end-to-end availability. Consider an SF path that passes through two rings: the signal passes through the first ring, then the inter-ring connection, and then the second ring. This situation can be conceptually abstracted as a parallel-serial-parallel system. It can be easily shown that the serial element dominates this type availability calculation because a single element failure on the serial subsystem will cause system outage. On the other hand, the SF cost is the lowest of the three inter-ring connection configurations to be considered.

3.4.2 Matched Nodes (MN)

In an MN configuration, each inter-ring connection has two physical links between the two rings. Therefore, there will be two pairs of inter-ring gateways. Figure 3.8 is an example of an MN configuration. When a single feeding incoming signal passes through node A to node W (primary gateway on ring 1), the signal is duplicated and passed on to the add-drop card. This duplicated signal then passes through the cross-office wire to node Y (primary gateway of ring 2). The original signal in node W continues on to node X (secondary gateway of ring 1), then passes through cross-office wire to node Z (secondary gateway of ring 2). This action is called "drop and continue" because the signal is dropped on node W and continues to node X.

After the duplicated signal arrives at node Z, it is sent back to node Y and terminated. During normal operations, the original signal is used to continue the path. If the cross-office wire between node W and node Y is cut, for instance, then the duplicated signal will be used. This configuration is protected from all single-element and most dual-element failures of inter-ring connection; however, the cost is considerably higher than that which uses SF.

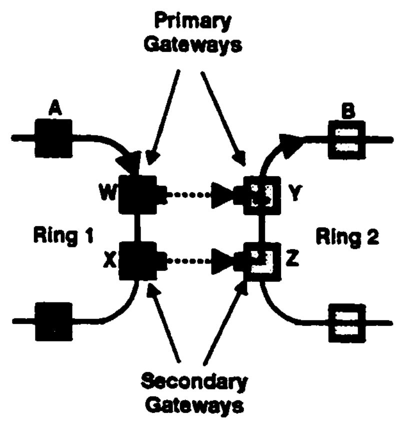


Figure 3.8: Inter-Ring Connection Configuration: Matched-Nodes

3.4.3 Dual Feeding (DF)

In a DF configuration, each inter-ring connection also has two physical links between the two rings. Therefore, there will again be two pairs of gateways, as in the MN configuration. For any path at the inter-ring connection point, there will be two copies of the same signal that travel through the physical links without rejoining. This configuration will be protected from all single-element and most dual-element (a different set of combinations from MN) failures for inter-ring connections. At first glance, because of the duplication of signals, it appears that DF consumes more bandwidth than MN; however, a more careful study in [4] shows that dual feeding can cost less in some specific system configurations particularly when rings are small or the separation between entry/egress nodes is large. One of the research goals of this thesis work is to attempt to exploit this feature by combining both matched-node and dual feeding configurations, in order to achieve the most cost-effective solution while maintaining high availability.

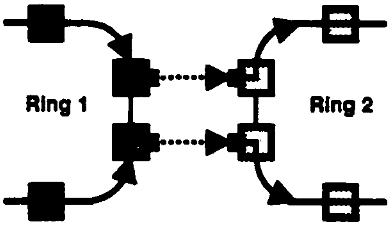


Figure 3.9: Inter-Ring Connection Configuration: Dual Feeding

3.4.4 End-to-end Paths

The following three examples are sample end-to-end paths using pure single feeding, dual feeding and matched nodes.

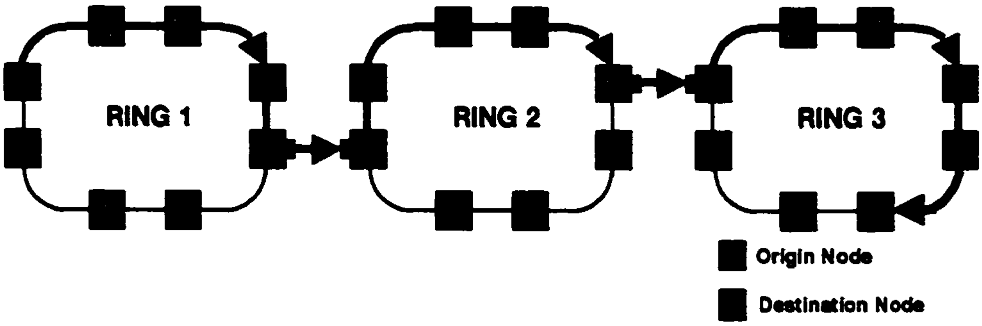


Figure 3.10: Sample Single Feeding Path

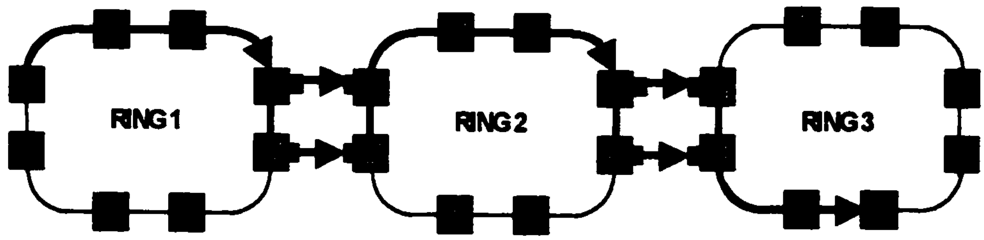


Figure 3.11: Sample Matched Nodes Path

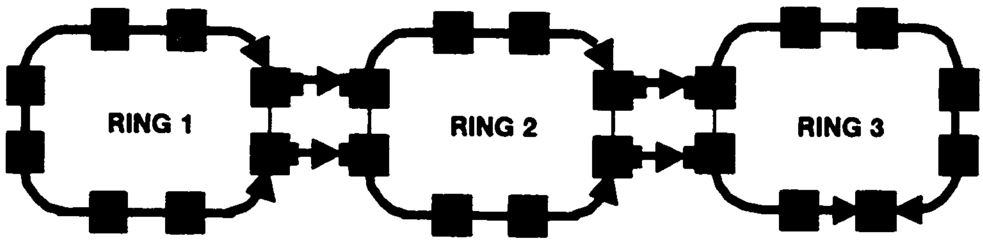


Figure 3.12: Sample Dual Feeding Path

3.5. Cost and Availability Models for Path Provisioning

3.5.1 Td_1 and Td_2 Formulations

The Td_1 and Td_2 functions are used to calculate the *intra*-ring path segment unavailability. The 'd' in the Td_1 and Td_2 formulations designates a dual-failure analysis. A dual failure analysis calculates the unavailability by summing up all the dual equipment failure combinations that cause outages. Single equipment failure is not accounted for because by design, single equipment failure will not cause outages. In addition, triple, quadruple...etc. equipment failure is not included because the resulting unavailability contribution is considered insignificant (e.g. 10^{-12} or less). The subscripts 1 and 2 denote whether a single or dual signal instance physically traverses a particular ring. Therefore, Td_1 is applicable to rings in SF and MN configuration, and Td_2 to rings in DF configuration. These were originally developed by Grover and the details are described in [4,5]. The following are the Td_1 and Td_2 formulations:

$$Td_1 = [\{s-w\}\cdot\{w\}]\cdot Usl^2 + [(w+1)\cdot\{s-w\} + (s-w-1)\cdot\{w\}]\cdot Unl\cdot Usl + [(w+1)\cdot(s-w-1)]\cdot Unl^2$$

Td_1 = total intra-ring unavailability
Usl = unavailability of 1km of span
Unl = unavailability of a node

s = number of spans in the ring
w = number of spans of this ring on the path
$\{s\text{-}w\}$ = total distance of the ring excluding the path
$\{w\}$ = total distance of the path inside the ring

Figure 3.13: Td_1 Formulation

$$Td_2 = [\{Wa\}\cdot\{Wb\}]\cdot Usl^2 + [(Wa+1)\cdot\{Wb\} + (Wb+1)\cdot\{Wa\}]\cdot Unl\cdot Usl + [(Wa+1)\cdot(Wb+1)]\cdot Unl^2$$

Td_2 = total intra-ring unavailability
Usl = unavailability of 1km of span
Unl = unavailability of a node

Wa = number of nodes on the first path in this ring
Wb = number of nodes on the second path in this ring
$\{Wa\}$ = total distance of the first path segment
$\{Wb\}$ = total distance of the second path segment

Figure 3.14: Td_2 Formulation

The Usl and Unl values could be obtained from statistics gathered by network operators or equipment vendors, and this study assumes that these values are given. To further elaborate on on Td_1 and Td_2 formulations, each term is briefly described here. The multiplier for Usl^2 is the total combination of dual span failures that will cause an outage. The multiplier for $Unl\cdot Usl$ is the total combination of node-span combined failures that causes an outage. Lastly, the multiplier for Unl^2 is the total combination of dual node failures that cause an outage. The three terms add up to Td_2, which is the intra-ring unavailability of the segment of an end-to-end path that traverses the given ring. Note that the add-drop interface is not taken into account because it is only an intra-ring path segment. However, it will be taken into account of the end-to-end path unavailability calculation later in the discussion.

3.5.2 Cost and Availability of Intra-ring Paths

On a ring-by-ring basis, it can easily be concluded that SF has the lowest cost and the lowest availability when compared to either MN or DF [5] based on the fact that MN and DF are dually redundant in terms of inter-ring connection, while single inter-ring equipment failure on SF will cause outage. In terms of comparing MN and DF on a ring-by-ring basis, the conclusions may be different depending on the configuration setup of the intra-ring path.

Figure 3.16 was generated using a cost criterion developed in [5]. It is applied at every valid point in the application space of rings from 4 to 16 spans in size, and for the quantity *Etot* (figure 3.15) from 2 to 14 spans.

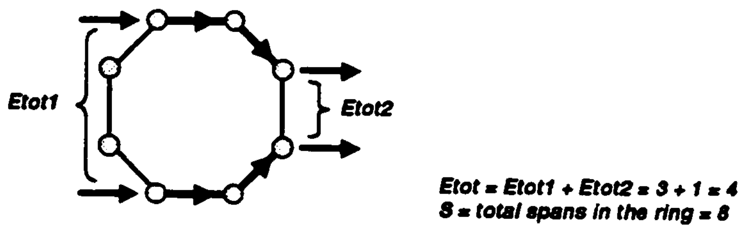


Figure 3.15: Explanation of *Etot* and *S*

Etot is the sum of the 2 "continue" signals on an MN ring. Figure 3.16 is also generated from a fairly conservative assumption about the length of the shortest working path through the ring in terms of the ring size: it is set at 1/4 of the ring length after excluding the entry and egress separations. This is for illustration purposes only.

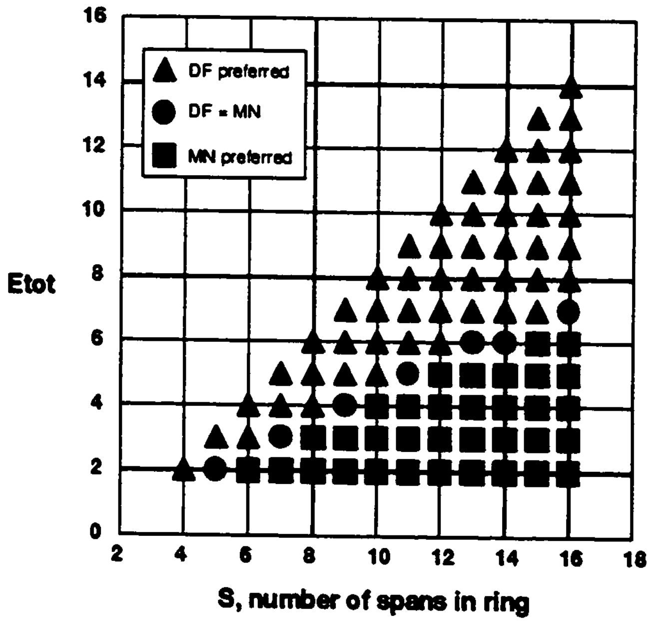


Figure 3.16: Sample of Cost-based Decision Criterion

It can be seen that the application area for MN or DF in various situations is quite balanced, meaning that MN and DF are both widely applicable. Note that figure 3.16 is taken from [5] and the details are described in the reference.

For availability, the situation is quite similar. By using the Td_1 and Td_2 formulations, the decision to use either MN or DF can be made on a ring-by-ring basis. We now analyze a sample case of a ring with 8 nodes, Usl = 1.0e-6 per km, Unl = 1.0e-5, and all spans equally assigned a distance of 1 km. The following graph shows various cases in which MN is lower in unavailability than DF, and other cases in which DF is lower than MN. Note that the X axis is set up for reference only and has no sequential meaning because MN is fundamentally different from DF, where MN uses Td_1 with one signal while DF uses Td_2 with two signals.

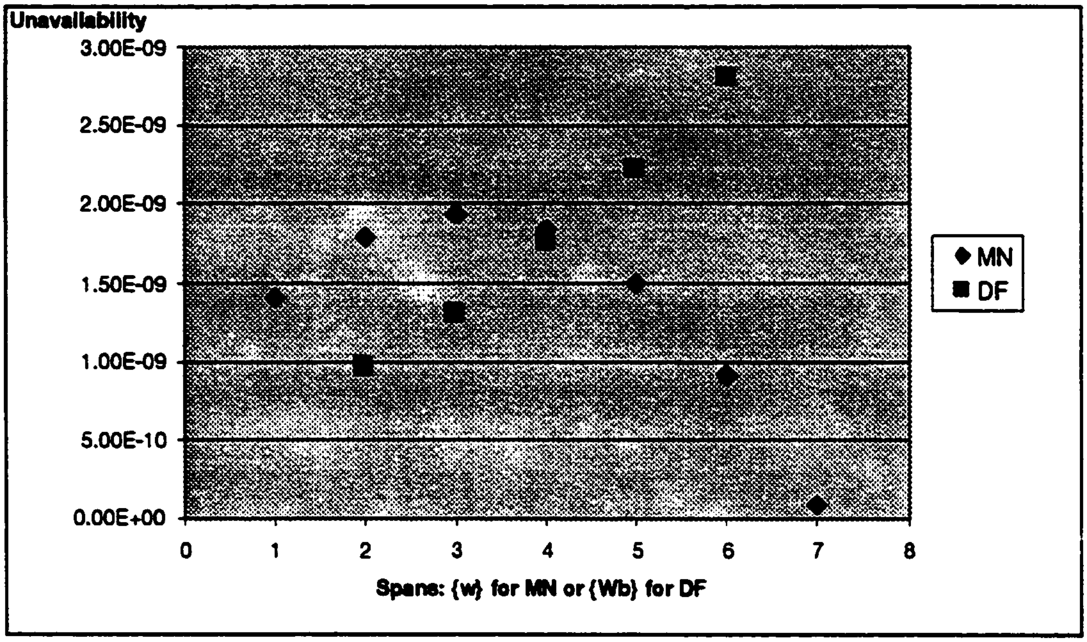


Figure 3.17: Sample of Unavailability-based Decision Criterion

3.5.3 Cost and Availability of End-to-end Paths

The cost of an end-to-end path is just the summation of all intra-ring path segments costs, plus the inter-ring connection equipment cost (i.e. add/drop cards), and the cost of the origin and destination add/drop card for terminating the signal.

The unavailability calculation is different for each of the following cases: SF-only path, MN-only path, DF-only path, and MN/DF combined path. First, the SF-only path is simply the series-element summation of the intra-ring (i.e. Td_1) and inter-ring unavailability in the path. Like Td_1 and Td_2, the inter-ring unavailability is also based on dual failure analysis. For details of all the failure combinations, please refer to [5]. For SF-only paths with K rings, we have the following formulation:

$$U_{sf}[K] = 2K(U_{na} + U_{nl}) + \sum_{i=1}^{K} Td_1[W_i, S_i]$$

U_{sf} = total intra-ring unavailability
U_{na} = unavailability of an add/drop port
U_{nl} = unavailability of a node

Figure 3.18: Single Feeding Path Unavailability Formulation

The following figure shows the unavailability calculation for an MN-only path. The first term is the entry/egress non-redundant failures, the second term is the inter-ring failures, and the last term is the summation of all intra-ring failures.

$$U_{mn}[K] = \underbrace{2(U_{na}^2 + 2U_{na}U_{nl})}_{\text{Entry/egress}} + \underbrace{2(K-1)(U_{nl}^2 + 2U_{na}^2 + 4U_{na}U_{nl})}_{\text{Inter-ring}} + \underbrace{\sum_{i=1}^{K} Td_1[W_i, S_i]}_{\text{Intra-ring}}$$

Figure 3.19: Matched Nodes Path Unavailability Formulation

The next figure shows the unavailability calculation for a DF-only path. The first term is the inter-ring connection failures, and the last term is the summation of all intra-ring failures. Note that the second term is the number of redundant failure modes that is accounted for in both inter- and intra-ring failures.

$$U_{df}[K] = 4K^2(U_{nl} + U_{na})^2 - 2KU_{nl}^2 + \sum_{i=1}^{K} Td_2[W_{Ai}, S_i, E_i]$$

Figure 3.20: Dual Feeding Path Unavailability Formulation

Lastly, figure 3.21 shows the formulation that calculates MN/DF combined path unavailability. The first term is the failure modes for all MN intra-ring path segments using U_{mn}. Next is the term applicable to DF segments. The first part of this is the calculation for all DF intra-ring path segments using U_{df}, and the next two parts involve in calculating a special class of failure modes which only happen to embedded DF segments. Details of all end-to-end path unavailability formulations are described in [5].

$$U_{dfmn}[N_{df}, N_{mn}] = \sum_{j=1}^{N_{MN}} U_{mn}[K_{mn}[j]] + \sum_{i=1}^{N_{DF}} \{U_{df}[K_{df}[i]] + 2\delta_{df}[i](U_{na}^2 + 2U_{na}U_{nl}) + 4(1+\delta_{df}[i]) \cdot (K_{df}[i]-1) \cdot (U_{na}+U_{nl})^2\}$$

Figure 3.21: MN/DF Combined Path Unavailability Formulation

3.5.4 End-to-end Path Comparisons

With the inter-ring connection configurations of SF, MN and DF, an end-to-end path with no single points of failure can now be provided using combinations of these approaches. Single feeding should only be used by itself and not combined with the other two methods (MN and DF) because the single point of failure in any SF inter-ring site already brings down the availability, and the use of MN or DF will not help. On the other hand, since SF does not employ any inter-ring redundancy, SF should be applied on all inter-ring connections to minimize the cost. Therefore, SF paths are the lowest in cost, but they are also relatively the lowest in availability.

MN-only paths and DF-only paths are both generally excellent in availability; however, the cost is high compared to that of SF because of the investment in dual redundancy at every inter-ring transition. As mentioned in previous sections, MN and DF may cost different amounts in various situations. Therefore, in order to take advantage of this, it is proposed to combine MN and DF end-to-end paths [4,5]. The following table is a summary of the discussion.

	Provisioning Cost	Availability	
Single-Feeding	LOW	GOOD	
Matched-Node	HIGH	EXCELLENT	
Dual-Feeding	HIGH	EXCELLENT	
Mixed MN/DF	REDUCED	EXCELLENT	

Table 3.1: Problem Review Summary

4. Problem Definition

4.1. Background

As mentioned in [1], the SONET self-healing ring is one of the key transport technologies that allows network operators to achieve high service availability. However, deciding whether or not to provide dual-redundancy on inter-ring connections is a more difficult task.

A single feeding path is the choice for inter-ring connection if availability is of the least concern to the customer or if the customer or application requires outside plant line protection but is willing to accept risk of a cross-office disruption. Considerable costs for extra add/drop interfaces and system bandwidth consumption are thereby saved. However, given the complexity of ring-based networks, the issue of manually determining how even an SF path should be provisioned through multiple rings is a tedious task for the network planner.

When the customer has a specific availability requirement which single feeding may not be able to provide, dual-redundant inter-ring connections should be applied. Existing bi-directional line switched ring (BLSR) networks utilize matched-nodes configurations to provide inter-ring dual redundancy. As introduced in [4], dual feeding is an alternative inter-ring connection strategy that has proved to be able to provide cost-effective solutions in certain network configurations. The combined use of MN and DF will be even more advantageous depending on different settings of the ever-changing network configurations.

If the combined use of MN/DF inter-ring connection is to be deployed, a generalized path provisioning tool using SF/MN/DF, based on specific cost/availability requirement would be much desired. However, automating the search for a path that specifically suits customers' requirement based on cost/availability is not a trivial problem, even in a computerized planning environment. Furthermore, customers may have various preferences on both cost and availability as their optimization objective. For example, a customer may desire a path with minimal unavailability within a specific budget. Therefore, the tool should deploy various strategies in absolute and constrained search of optimal (or near optimal) solutions as appropriate. Such is the ultimate goal of this research effort, which are the three objectives mentioned in section 1.2.

4.2. Specific Objectives

The following are specific objectives that we want to achieve in this research:

A. **Single Objective Optimization**: to develop path search strategies that can provision an optimal or near-optimal path in terms of absolute cost minimization or availability maximization.

B. **Single Objective Optimization with a Constraint**: to develop path search strategies that can provision an optimal or near-optimal path in terms of cost or availability, with the other requirement set as a maximum value constraint (i.e. maximum cost constraint, or maximum unavailability constraint).

C. To develop a software prototype that will enable the search strategies in A and B to be executed and yield optimal or near-optimal path constructions satisfying the objectives and constraints in a timely manner.

Objectives A and B will consider all of the following available inter-ring connection configurations: matched-node only (MN), dual feeding only (DF), or MN/DF combined. In order to be consistent, the optimization problem will be referred to as "minimizing cost", and "minimizing *unavailability*" (instead of "maximizing *availability*") through the rest of the thesis. The following table summarizes the problem domain:

	User Requirement			
Path Configuration	Minimizing Cost	Minimizing Unavailability	Minimizing Cost with Maximum Unavailability Constraint	Minimizing Unavailability with Maximum Cost Constraint
MN-only Paths	X	X	X	X
DF-only Paths	X	X	X	X
MN/DF combined Paths	X	X	X	X

Table 4.1: Problem Domain for the Software Prototype

In other words, the software prototype should be able to solve the above twelve variations of the path-provisioning problem that is manifest in ring based networks. In addition the capability to find min cost paths of an SF nature and to analyze their availability will be undertaken.

4.3. Graph Representation

In this section, ring-based networks and the problem of path provisioning through them is first appreciated in a graphical way, and then preliminary analysis is performed based on some observations from this exercise.

4.3.1 Network Topology

Figure 4.1 illustrates the topology of a sample network.

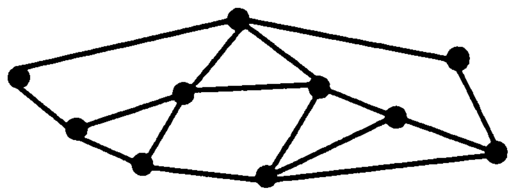


Figure 4.1: Network Topology

A network topology is a graph which consists of nodes (or vertices) connected by edges. Each node represents a point-of-presence (e.g. central office), a physical location where the network provider may be able to install network equipment. Note, however, that a node on the network topology does not necessary imply the existence of a network element on that physical site.

In our problem, each edge represents a fiber optic transmission span, that is, a physical connection that allows certain number of physical fiber optics to lie between the two POPs. The existence of a span simply means that a network connection using fiber optics between these two sites is possible. An inactive fiber connection is referred to as "dark fiber".

In figure 4.2, the bottom part is the same network topology as above. Now, however, each colored donut shape placed on the network topology reflects a ring-based transport system that has been deployed, while the top part illustrates the rings themselves as an overlay. Each ring represents a SONET BLSR being deployed with its network element physically located at the POP indicated below on the network topology graph.

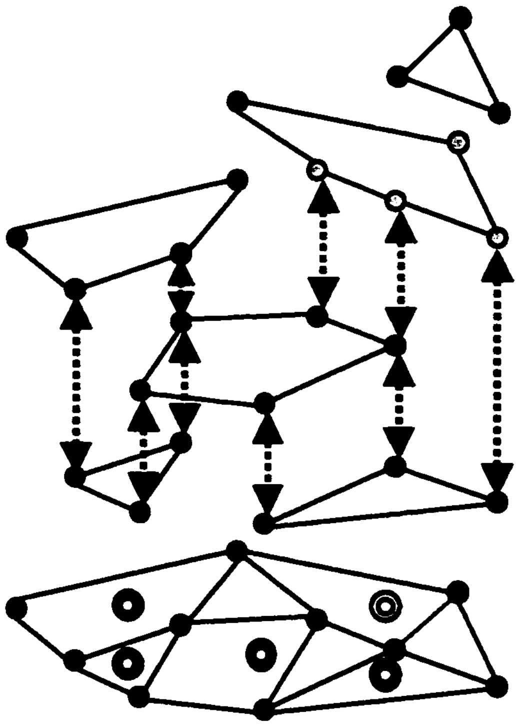


Figure 4.2: Ring Overlay Network on top of Network Topology

The vertical dotted bi-directional arrows represent actual inter-ring connections deployed (i.e. cross-office connections). For example, the orange ring and the green ring have two network elements (i.e. add/drop multiplexer) which are placed on the same physical site. Therefore, inter-ring connection is possible between these two rings at that site. Note that the grey node on yellow ring is a glass through node, which means that there is no active network equipment presented, the location only used a by-pass. It is also legitimate to provision a ring that is not inter-connected to other rings (i.e. red ring).

4.3.2 Ring Sequence

Since we are concerned with paths that traverse inter-ring connections, the origin / destination node pair (referred to as an O-D pair) may not on the same ring for problems of interest to us. If the O-D pair is not on the same ring, the path provisioned for this set up will have to go through a number of rings (at least two rings) referred to as a "ring sequence". Note that a valid ring sequence must consist of rings listed in sequential order, that is, ring i must have an ADM at the same POP as ring $i+1$ (which also has an ADM), with cross-office connections deployed between the two ADMs. In short, subsequent rings must be physically capable of being inter-connected in order to compose a valid ring sequence. For example, in figure 4.2, purple-orange-green is a valid ring sequence, as is yellow-blue. However, purple-blue is not a valid ring sequence. In addition, purple-yellow is invalid because the glass through node in the yellow ring does not have equipment to inter-connect with purple ring. Also, red-yellow is not valid either because there does not exist any cross-office connection between the two rings.

4.3.3 Ring Connectivity Graph

Before attempting to search for a desirable path given the user requirement, we can further convert the ring network information into a more manageable form by transforming the network information into a simpler graph in terms of its ring connectivity. By applying a ring overlay transformation, path search algorithms for ring sequence can be more directly applied.

Basically, the network topology information will be used to transform the ring overlay into a ***ring connectivity graph*** (RCG). In an RCG, each *node* represents a ring on the network and each edge, if it exists on the RCG, implies that the two rings are able to support an inter-ring connection. This requires that there exists at least one node from each ring (in the ring overlay network) that is located at the same POP as a node from another ring.

By using this rule, the network can be transformed into an RCG with two different criteria. The ***single connectivity criterion*** states that there is an edge between two nodes in the RCG if and only if there exist at least one node from each ring (in the ring overlay network) located at the same point-of-presence (resulting graph shown in Figure 4.3 as *RCG/1*). The ***dual connectivity criterion*** states that there is an edge between two nodes in the RCG if and only if there exists at least two nodes from each ring (in the ring overlay network) located at the same two points-of-presence (resulting graph shown in Figure 4.3 as *RCG/2*). This procedure will be called the **Ring Overlay Transformation** (ROT). Figure 4.3 illustrates RCGs using the two different criteria.

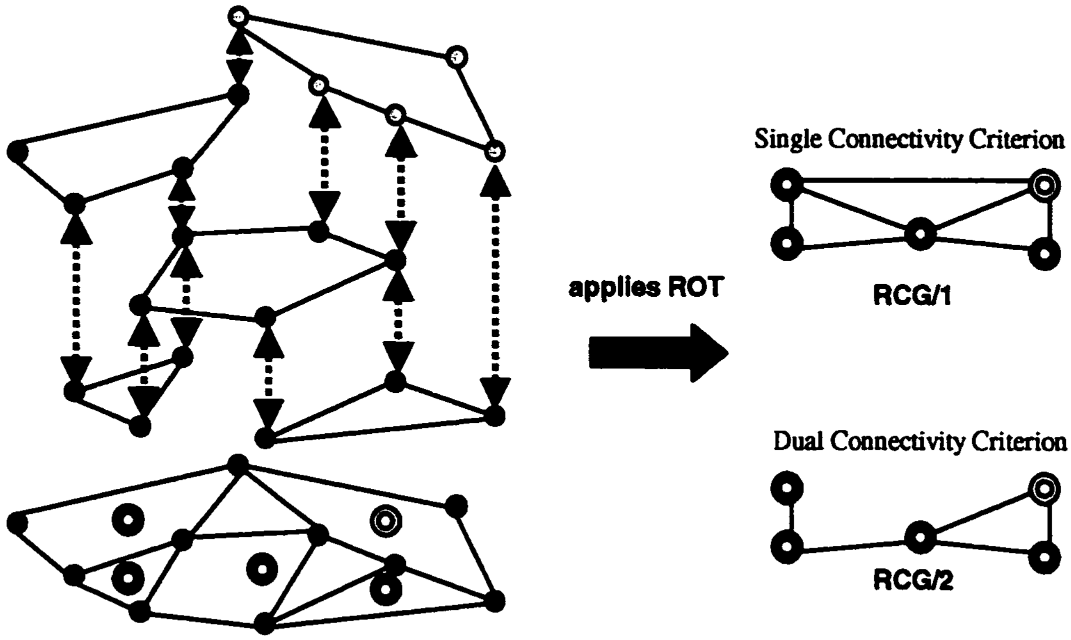


Figure 4.3: Ring Overlay Transformation using two criteria

The next figure shows a more complex example using the dual connectivity criterion.

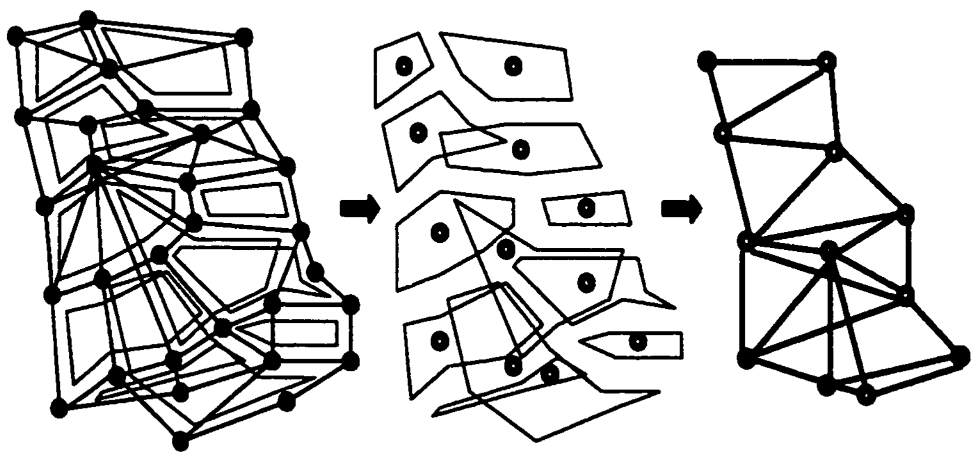


Figure 4.4: Another Example of using Ring Overlay Transformation

4.3.4 Path Construction

In this section, the end-to-end path construction is characterized. We will use a step-by-step procedure in building the path. This procedure reflects what a network planner has to analyse manually when provisioning an end-to-end path given a pair of specific origin and destination nodes on a ring network. After defining it, we will be able to use the methodology to help us understand the problem and to search for opportunities to optimize the solution, both in terms of run-time and optimality.

1. Identify the origin and destination nodes on the network topology.

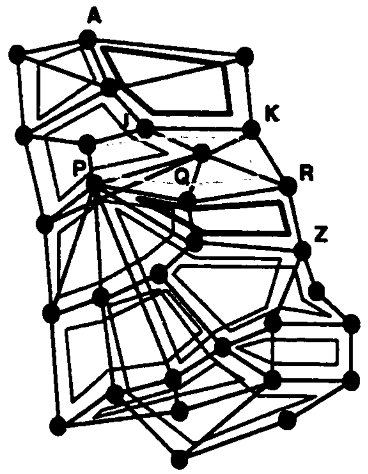


Figure 4.5: Identifying O-D pairs and O-D rings

In this case, node A and node Z on figure 4.5 are chosen as the O-D pair. Then, the rings the O-D pair resides on are identified, and they are called the "O-D rings". For instance, in figure 4.5, node A resides on two rings and node Z resides on three rings (the product of these two values is called the "O-D rings multiplying factor" which is 2 * 3 = 6 in this case; this value will be used to analyze the problem complexity in Chapter 5). One ring is identified as the O ring and one ring as the D ring. In this example, let the green ring be the O ring for node A, and the blue ring is D ring for node Z.

2. Specify a ring sequence for the path. In this example, let green-yellow-blue be chosen as the ring sequence, as illustrated in figure 4.5.

3. Specify the configuration at each inter-ring connection site.

Since the ring sequence is size 3 (i.e. 3 rings), either MN or DF must be applied at the two inter-ring connection sites. Before assigning the configuration, the two gateways for each site (both MN and DF requires two gateways) must be identified. For the interconnection at green-yellow rings, no decision needs to be made about which gateways to use because there are only two available nodes (i.e. nodes J and K) that are common to both the green and yellow rings. Note that the "node" in the gateway context is the network topology, not the node on the ring system. Strictly speaking, the point-of-presence is being chosen here.

For the site at the yellow-blue interconnection, three common nodes are available for gateways. Figure 4.6 shows the set up at the yellow-blue interconnection:

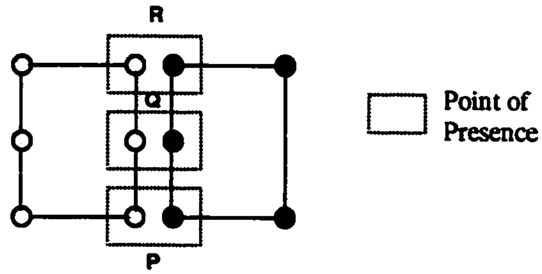


Figure 4.6: Identifying Gateways

The next figure shows the possible gateway combinations:

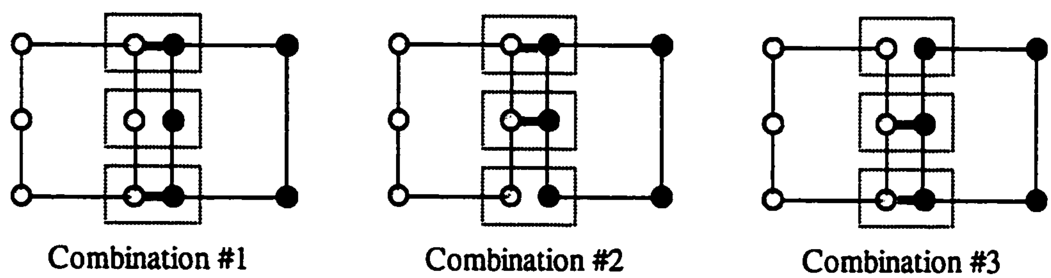


Figure 4.7: Gateways Combinations

In this case, let us say combination #1 is chosen, that is the P and R nodes. However we note, in general, that for k possible gateways, (k choose 2) combinations exist to consider. The following figure summarizes the logical view of the partially completed path:

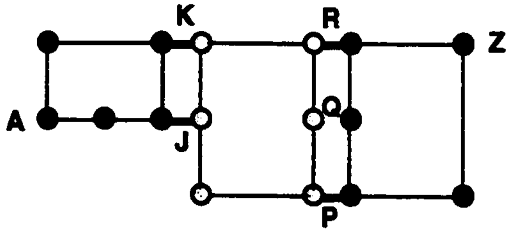


Figure 4.8: Identifying Gateways

4. The last design choice that has to be made is to choose to use MN or DF at each inter-ring interface, and in the case of MN, to further decide the signal flow directions in relevant rings. In this case, let us choose DF for the green ring.

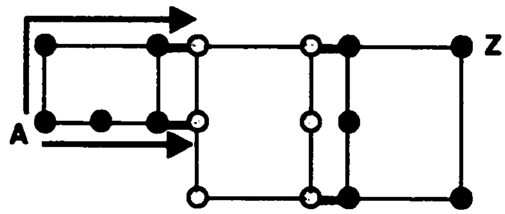


Figure 4.9: DF Signal Flows on Green Ring

The DF signal flow for the green ring is illustrated in figure 4.9. Combining for illustration, let MN be adopted for the yellow and blue rings. Although MN or DF configurations can be applied independently on a ring-by-ring basis, in order to demonstrate the naming convention of primary and secondary gateways, both rings will be considered at the same time to advance our working example. In figure 4.10 there are four combinations of signal flows, that is, two combinations for each ring.

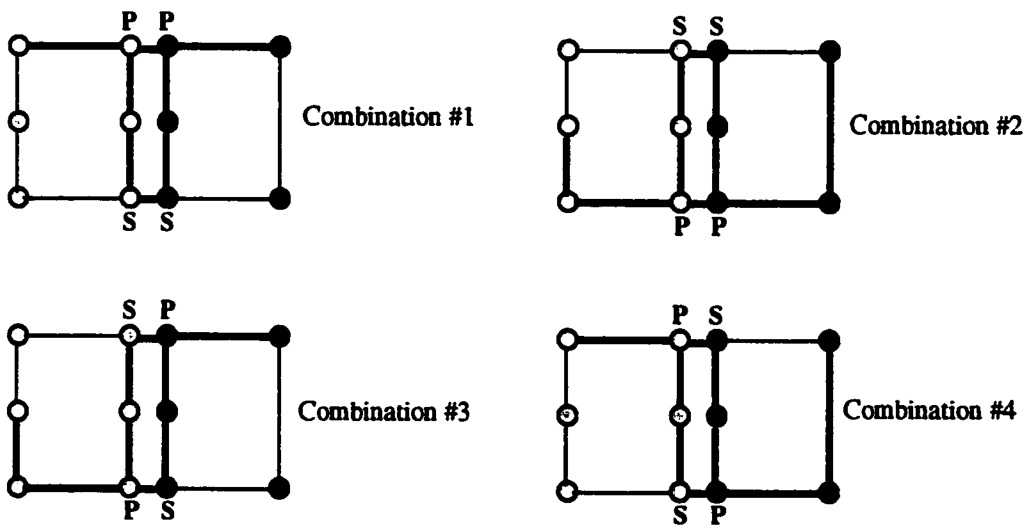


Figure 4.10: Possible MN Signal Flows Combinations on Yellow and Blue Ring

To refer to the primary and secondary gateways, the ring node in which "drop and continue" is performed is called the primary gateway. The other gateway on the same ring is referred to as the secondary gateway.

For the ongoing path construction example, let us choose combination #1. The completed path construction for the example is summarized in figure 4.11.

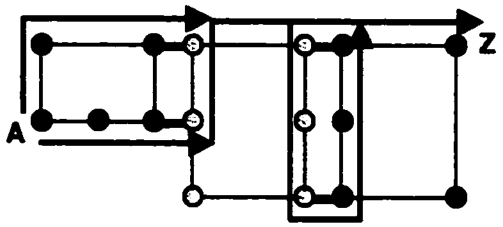


Figure 4.11: Complete Path Construction

4.4. Problem Review

In order to achieve objectives A, B and C mentioned in section 4.2, the path construction procedure described in 4.3 will have to be used. The following are the input and output factors of the research problem:

INPUT	OUTPUT
1. Network topology 2. Ring Overlay Network 3. O-D pair 4. Path Optimization Requirement: • Minimizing Cost • Minimizing Unavailability • Minimizing Cost with Maximum Unavailability Constraint • Minimizing Unavailability with Maximum Cost Constraint 5. Path Configuration: • MN-only • DF-only • MN/DF combined	1. An end-to-end path with the following characteristics specified: • O-D rings • Ring sequence • Gateways configuration for each ring • Path configuration for each ring (MN or DF) • Signal flow on each ring 2. Cost and unavailability information for the path

Table 4.2: Problem Review Summary

In summarizing how to build a path construction, first we have to gather the information for the network topology, the ring overlay set, and the origin/destination node pair. Then we have to apply the ROT to the network topology and ring overlay set to get RCG. Now we can select the appropriate ring sequence that connects the origin and destination nodes by using the RCG. For each inter-ring connection site, we have to choose two gateways to interconnect the rings in order to have dual redundancy. Finally, we have to apply either MN or DF on each of the ring and, specifically for MN applied rings, we have to specify the signal flow in each of those rings.

5. Problem Characterization and Complexity Analysis

5.1. Shortest Path Algorithms

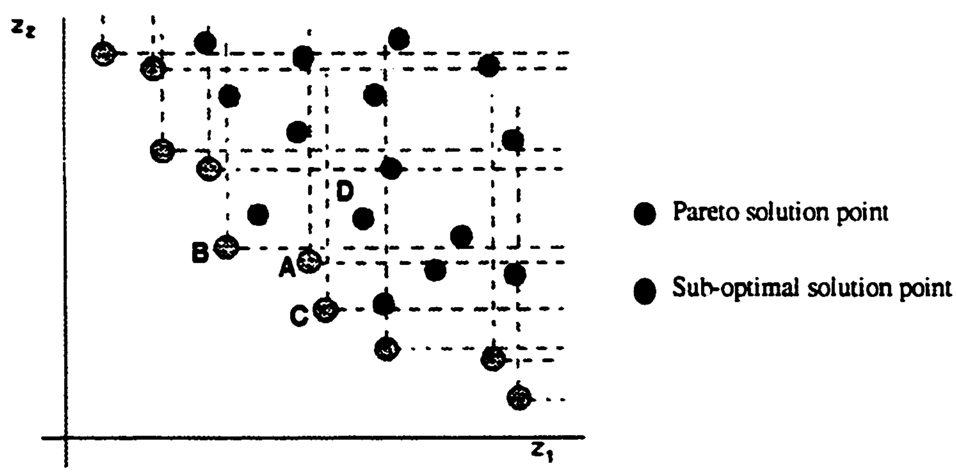
5.1.1 Single- and Multi-Objective Shortest Path Algorithms

In considering this research problem, the first intuition is to use some kind of shortest path (or least cost) algorithm on an RCG representation of the ring set (at least in the case of a single objective). However, with further analysis, it is apparent that aside from the pure min-cost SF problem, the greater generality of problem requires more than simply applying Dijkstra's shortest path algorithm [20]. Let us now consider how to optimize cost and availability simultaneously.

Consider the general problem of a decision maker who has to find the path in the graph which is optimal with respect to weight and length (i.e. two different attributes of an edge). If the utility function over weight and length is not linear, then no general efficient algorithm exists [8]. In a specific case of determining whether there is a path from s to t with weight at most W and length at most L, this problem is NP-complete [9]. The cost and availability of the problem can be characterized as the two attributes (weight and length) in the multi-objective shortest path problem. In this section, an overview on the basis of multi-objective (specifically bi-objective) shortest path algorithms is presented and their applicability to the problem at hand is evaluated.

The bi-objective path problem can be classified based on a number of single objective path problems [10]. For example, MINSUM (cost), MINPRODUCT (reliability) and MAXMIN (capacity) are some of the single objective functions. In particular, the MINSUM-MINSUM (minimizing the cost and unavailability) problem will be our focuse. Note that the MINPRODUCT or MAXPRODUCT problem can be transformed into the MINSUM problem [10]. The MINSUM-MINSUM problem is also known to be *intractable* in the worst case, i.e. it requires that the number of operations grow exponentially with the problem's characteristics.

To understand the basis of multiple objective optimization, the composition of the solution space must first be comprehended. For bi-objective optimization, there are two objective functions, $Z1$ and $Z2$. A feasible solution x that is not dominated by any other feasible solution y is said to be an "efficient" solution [5], that is, further improving one objective will necessarily result in a degradation of the other objective. For example in Figure 5.1, say, if we are minimizing both functions $Z1$ and $Z2$., then each grey point describes one Pareto solution, and all the grey points together comprises the Pareto solution set. Notice that the solution points are all discrete. This means that each solution point represents a discrete value. For example, say we are finding the shortest path in a network with two objectives (e.g. cost and unavailability). For the Pareto solution point A, if we want to further minimize on function *Z1*, then we have to use solution point B, in which we will have a lower *Z1* value but a higher *Z2* value. As for further minimizing function *Z2* on solution point B, the result will be point C, in which we will have a lower *Z2* value but a higher *Z1* value. In summary, the Pareto solution set is the set of solutions in which for each solution point, it is not possible to improve on *Z1* without worsening *Z2* (and vice versa). And for solution point D, it is not a Pareto solution point (hence not efficient) because we can further minimize either *Z1* or *Z2* without any penalty, i.e., lowering a value of an objective without raising the value of another objective (as oppose to point A, B or C).



Therefore, for the Pareto solution points, we say that they are efficient solutions.

Figure 5.1: Solution Space and Pareto Solution Set

Most techniques used in solving the bi-objective (or multiple objective) shortest path problem focus on generating the set of efficient paths, then allow the user to pick one of the set depending on his/her preferences. Algorithms are mostly multiple labeling-based (like Dijkstra's) [8,10,12,13]; some approximation algorithms have also proposed [10,13]. Other papers show that the problem can be formulated and solved as a linear programming problem [13,14].

The numbers of non-dominated solutions to two objective shortest path problems may grow exponentially with the problem size. As the author of [15] suggests, "generation of the entire non-dominated solutions set is not practical for most real-world problems". Moreover, in our context, network planners may have a strict preference for one objective (i.e. cost) whereas they may only be concerned that the other objective be below or above an acceptable limit. Hence we have a domain of constrained single objective optimization. However, the demand for the second objective is mostly driven by the customers (i.e. availability).

5.1.2 Resource-Constrained Shortest Path Algorithms

After the analysis on section 5.1.1, it can be realized that our particular problem may not necessarily be viewed as a bi-objective path problem. And for more practical applications, the problem can be characterized in a more restricted form with two variations in terms of cost and unavailability:

1. minimizing cost subject to maximum unavailability constraint; or
2. minimizing unavailability subject to maximum cost constraint.

These are single objective path problems with a side constraint. The shortest path problem subject to side constraints is also known to be "as least as hard as NP-complete problems" [16,17]. Various approaches, including linear and dynamic programming methods, have been proposed [15,18]. Also, a number of pseudo-polynomial time algorithms have been examined [9,16,18]. Another approach, which may be more suitable for this problem, is the use of *k*-shortest paths algorithms (i.e. the 2nd shortest path, the 3rd shortest path...etc) followed by testing for satisfying of the related constraint. The shortest path algorithm (e.g. Dijkstra's) can simply be used to evaluate if the shortest (in terms of cost or unavailability) path satisfies the additional constraint (max unavailability or max cost constant). If it does not, it will then evaluate the 2nd shortest path, so on.

The basis and applicability of *k*-shortest paths algorithms will now be evaluated. One approach to find the *k*-shortest paths is to generate all possible paths from source to sink, and then to sort the paths according to their length. The result will be the list of all paths from shortest to longest [19]. Depending on the application, if the set of all s-t paths (a path that connects source s and destination t) is used extensively, or the *k* is very large, this approach might be considered. However, it is obvious that the storage requirement is enormous as the number of s-t paths may grow exponentially. Using the *k*-shortest paths algorithm for solving a resource-constrained shortest path problem guarantees solution optimality; however, this approach gets impractical (run-time) when the terminal value of *k* is very large. In addition, no matter how big or small the value of *k* is, the same number of operations is required because we have to generate the whole set of paths and sort them all.

Another approach is to first find all *k*-1 shortest paths, then set the distance of each edge in each of the 1^{st}, 2^{nd}, ... $(k-1)^{th}$ in turn to infinity. The shortest path problem is solved in each case. The shortest of all these cases is the resulting *k* shortest path [19]. For example, for 2^{nd} shortest path, we first have to find out the 1^{st} shortest path. Because the 2^{nd} shortest path must have at least 1 edge different from the 1^{st} shortest path, thus for each edge of the 1^{st} shortest path, we set its length to infinity and apply shortest path algorithm. Note that the number of shortest path algorithms to be performed grows exponentially with the value of *k*, therefore, this method is quite applicable to small values of *k*. It is apparent that this method is at least as fast as the previous method.

Another method for solving *k*-shortest path problem includes a branch-and-bound procedure that evaluates all paths that "branch" out from the shortest path. A method using a similar idea is to evaluate the "deviations" paths of the *k*-1 path to gain the *k*-shortest path [19]. Most of the evaluated *k*-shortest path algorithms result in an algorithm of at least cubic-polynomial time complexity.

5.2. Path Provisioning as a Combinatorics Problem

Simple shortest path algorithms (e.g. Dijkstra's, *k*-shortest path) discussed in the previous section are aimed at simple graphs. Applying these algorithms to the current problem (in its RCG form) is not thought to be a good approach, however. Although considerable effort has been spent investigating and characterizing the problem, solving the problem as a classic "shortest path problem" seems inappropriate for the following reasons:

- A path on a ring-based network is physically on top of the ring overlay network and logically on top of the network topology
- A dual redundant path is not a general "path" in the graph theory context (e.g. drop and continue for MN; dual signals for DF)

As simple shortest path algorithms are not applicable to the problem, the problem can be viewed as a combinatorics problem which can be solved by using an approach similar to that used to generate the Pareto solution set. That is, generate all possible paths for the original/destination pair and retrieve the answer with desired attributes (e.g. max cost or unavailability value) from the solution set. As mentioned in section 5.1.1, generating the full solution set (i.e. every possible paths for the origin/destination pair) is very likely to be intractable when the problem size is large, given limited time and space. Hence, pruning the problem space and generating paths that are "likely" to be desirable should be considered. Before attempting to design the heuristics for pruning the solution space, we have to analyze the complexity of the solution space. By segmenting the factors contributing to the path complexity, we would be able to identify the paths that should (or should not) be pruned. This will allows us to maintain the solution optimality while still getting the answer in a timely manner.

5.3. Problem Complexity

5.3.1 Solution Space Estimation

In this section, the total number of path constructions for a specific O-D pair in a network will be estimated. Obviously, the number of path constructions for an O-D pair in a specific network cannot be calculated exactly without examining each of them. This is because each network is unique in size, connectivity, ring overlay,...etc. So it is difficult to give a general equation to calculate such a value. On the other hand, attempting to break down the solution space for pruning opportunities will provide a rough figure for estimation purposes related to the design of heuristics to be used later.

First, the use of an MN/DF combined configuration must be estimated, in a way that facilitates similar analysis for the other two cases (i.e. MN-only and DF-only). The goal is to assess the complexity of an approach which would generate the entire solution space and retrieve the desirable solution. Solving for the single objective or the single objective with the constraint approach (or any multi-objective optimization problem) can use a similar approach to that developed for the single objective problem.

First, let us consider a single ring sequence should be considered (figure 5.2). Since MN or DF can be applied on each of the rings, there will be 2^n MN/DF combinations if there are n rings.

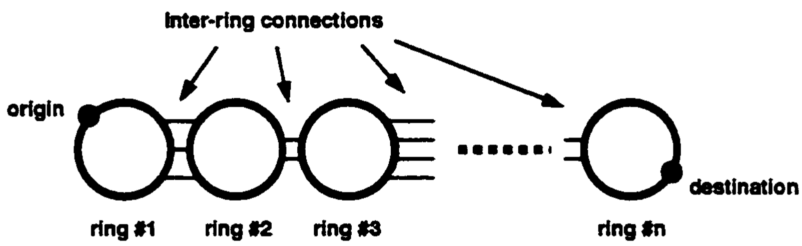


Figure 5.2: Sample Ring Sequence

Next, for the gateways arrangement, there will be "*k choose* 2" gateways arrangements for each inter-ring connection (because only two gateways are need for each site). Thus, there are $(k\ choose\ 2)^{n-1}$ for $n-1$ inter-ring connection sites. Next consider signal routing inside each ring: for DF rings, there is only one type of signal flow through a ring (figure 5.3).

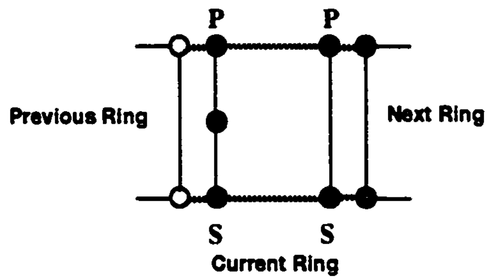


Figure 5.3: DF Signal Flow

But for MN rings, there are two types of signal flow: i.e. the single feed signal may flow clockwise or counter-clockwise from entry to the gateway nodes, as illustrated in figure 5.4.

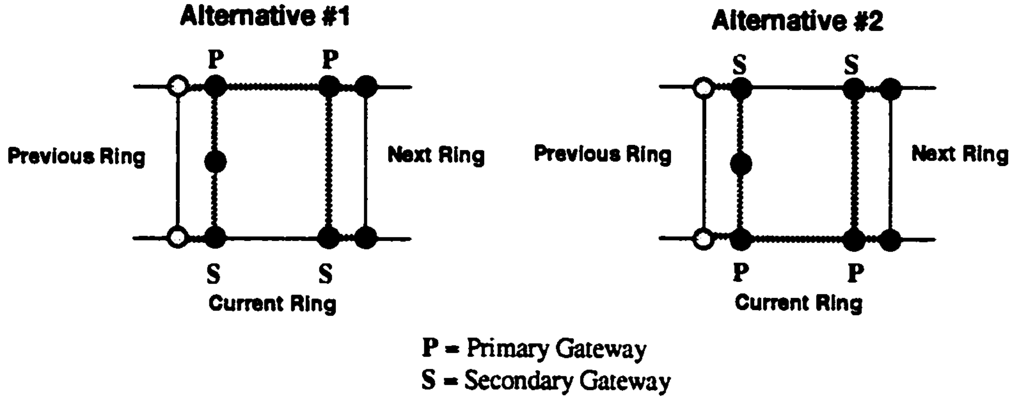


Figure 5.4: MN Signal Flow Alternatives

Therefore, if there are n MN rings, there may be 2^n different signal flow patterns within the collection of these rings. However, we have to assume that MN and DF each is used 50% of the time, this MN signal flow factor has reduces to $2^n / 2 = 2^{n-1}$ instead. We call this variable the "MN signal rating".

Having considered a single ring sequence, the next factor to be considered will be the total number of distinct ring sequences. Finally, because the origin or destination node may reside on more than one ring, the O-D ring-multiplying factor (ODMF) must be also taken into account. Figure 5.5 presents the final formulation for the path constructions estimate.

$$\text{Total number of Possible Path Construction} = 2^n \cdot \left[\binom{k}{2}\right]^{n-1} \cdot 2^{n-1} \cdot r \cdot f$$

n = number of rings in a ring sequence (average)
k = number of common nodes of two adjacent rings (average)

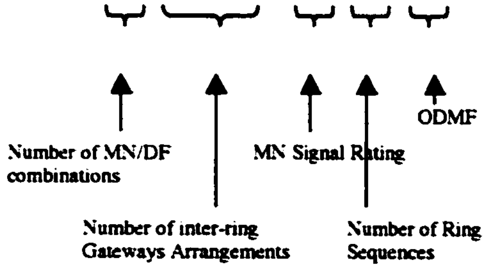


Figure 5.5: Path Construction Estimate Formulation

For a network of medium size (e.g. the continental MCI network), with an average ring sequence size of 8, average common nodes of 4, and total number of ring sequences of 40, and ODMF being 2, this results in an estimate of the following number of distinct path constructions

$$2^8 \cdot \left[\binom{4}{2}\right]^7 \cdot 2^7 \cdot 40 \cdot 2$$

$$= 256 \cdot 6^7 \cdot 128 \cdot 40 \cdot 2$$
$$= 256 \cdot 279936 \cdot 128 \cdot 40 \cdot 2$$
$$= 73383547278 = 7.34e11$$

← Total Number of Possible Path Constructions for an O-D pair

to be considered.

Figure 5.6: Path Constructions Estimate Example

It is easy to generalize the formulation in figure 5.5 to make it applicable to MN-only and DF-only paths. For MN-only, it is simply necessary to take out the number of MN/DF combinations because DF is not used, and also changing the MN signal rating from 2^{n-1} to 2^n because all rings use MN configuration:

$$\left[\binom{k}{2}\right]^{n-1} \cdot 2^n \cdot r \cdot f$$

Figure 5.7: Path Constructions Estimation for MN-only Paths

For DF-only, we take out the MN/DF combinations factor, and also the MN signal diversity factor:

$$\left[\binom{k}{2}\right]^{n-1} \cdot r \cdot f$$

Figure 5.8: Path Constructions Estimation for DF-only Paths

For pure MN and DF paths, we will simply use the reductions defined for the MN/DF combined path case. This is because pure MN or DF path provisioning is just a special case for MN/DF combined path cases.

5.3.2 Identifying Areas for Solution Space Reduction

In this section, we identify which contributing factor is dominant in our total path estimation. The value of MN/DF combinations (2^n) will not be too big because the base is 2 and n is expected not to be too high. Even when the number of rings (i.e. n) is 16, the number of MN/DF combinations is only 65,535. Since the combined use of MN/DF is the key to the research effort, in which cost and/or availability are improved over MN-only and DF-only, reducing the number of MN/DF combinations is inappropriate, therefore, reduction of MN/DF combinations is not considered.

As the network size grows, the total number of ring sequences grows exponentially. However, the growth rate is limited by the two-node connectivity requirement. Moreover, it can be as large as hundreds of thousands in a large network as we will see later on in section 8.2. Yet, a large number of them are 'excessively routed' ring sequences, that is, the ring sequence takes an unnecessarily long route. Based on observation, any ring sequence that is 10-15% longer (in terms of hops) than the shortest ring sequence could, for our purpose, be excluded. This is because ring sequence size has a high positive correlation with the amount of equipment deployed, which is in turn directly related to both the cost of the path and the path availability. Hence, the total number of ring sequences is a good candidate for reduction.

Next, as can be seen from the formulation, a dominant term of the calculation in figure 5.6 is the number of gateway arrangements, which increases dramatically as k increases. This is a good point for a practical limitation that helps to reduce the problem space. Specifically, if there are only two common nodes for any given two rings which are inter-connected, or we decide to consider at most two adjacent nodes, then the total PCs will be reduced to 262,144. Although high numbers of common nodes (>4) between two rings are rare, they may be a part of a long ring sequence. Thus the gateway arrangement value is also a good candidate for a practical constraint. Note that the above estimation is only used to demonstrate the dominant part of the path search, and the focus is on the path construction components where solution space pruning should be applied.

5.3.3 Run-time Analysis

Before provisioning a path for a customer's application, we propose that the network planner to analyze the cost and service availability in providing such service. To deliver the service to customers in a timely manner, a real-time path optimization design methodology is preferable. In the case of analyzing cost and service availability, our aim would be to complete the analysis within hours or even minutes, considering the day-to-day operation of a network service provider. It is preferred that the analysis be completed as soon as possible, while ensuring a reasonable level of optimality of the solutions (in terms of cost and/or availability).

For evaluating a path construction, a path can be constructed using the step-by-step procedure described in section 4.3. Then its cost and availability data can be calculated. However, even for just one path construction, the data manipulation and calculation are very intensive. Although today's computer power is enormous, calculations still takes time. If evaluating one path construction takes up 1 microsecond of CPU time, the sample calculation in figure 5.6, which has a total number of path constructions of *7.34e11* would take 7.34e5 seconds, which is 8.5 days.

Therefore, in order to obtain an optimal or near-optimal solution for a large network, some kind of heuristic must be deployed that can prune down the solution space (thereby reducing the run-time) while maintaining a certain level of optimality. On the other hand, the heuristic may also be interesting to smaller-sized networks because of its ability to analyze a path within a very short period time, allowing network planners to perform rapid evaluation. Therefore in the next chapter we will propose some heuristic using solution space pruning, based on the complexity analysis in this chapter.

6. Path Search Strategies

6.1. Background

This section presents the path search strategies proposed to cut down on run-time while still yielding solutions that can be optimal or near-optimal in terms of cost and/or availability. Note that since the solution space for a desirable answer is being generated, it does not make a difference whether a single objective, or a single objective with constraint or even multi-objectives is sought. These will require only different criterion for selecting the most desired path design result. Also note that the provisioning methods introduced are independent of the ring overlay set. However, the availability or cost of an individual path is dependent on the network design. For example, ring size, number of rings, rings interconnectivity...etc. On the other hand, it may not be desirable to alter network design to accommodate a path design because a single path serves only a small portion of the whole network demand.

6.2. Ring Sequences Reduction (RS-R)

6.2.1 Intuition and Approach

As mentioned in section 5.3.2, it is fairly evident that path cost and unavailability have a natural co-relation with increasing path length. Ring sequence length is a major factor because each extra ring will introduce an extra inter-ring connection (which lowers the availability), and will require at least four extra add/drop port cards and potentially more ADM nodes (which increases the cost). Therefore, ring sequence reduction is proposed as a very strong heuristic principle to exploit. Basically it builds on our belief about the problem domain that, for instance, a lower cost, higher availability path is unlikely to be found in say a 9-ring sequence if there is a significant population of 6-ring sequences available. For metro network, because the rings are often overlapped, the number of rings in a ring sequence is relatively small even with a large number of rings (e.g. 3 in Network A). For long haul network, it is more dependent on the physical distance that the origin and destination are apart. Even for a coast-to-coast US continental network like Network C, a value higher than 10 would rarely be expected.

Therefore, for small or medium sized networks (e.g. 5-12 rings), the number of ring sequences is not too large (<100) and may not need reduction at all. But for large networks, or networks with high ring connectivity, the number of ring sequences may be tens of thousands between the OD pair of interest. With the use of maximum ring limits, the total number of ring sequences can be reduced substantially. The limit can be set at around 5-10% of rings above the number of rings in the shortest ring sequence. That is, if the shortest ring sequence is 5 hops, then the largest ring sequence to be evaluated will be 7 hops. Any larger sized ring sequence will not be evaluated. Practically, we find later that setting the ring sequence limit at +2 of the shortest ring sequence, is often enough to encapsulate all the optimal solutions in the search bound (see section 8.4).

6.2.2 Run-time and Solution Optimality

Run-time is reduced substantially in large networks with a lot of rings, or networks with high ring connectivity, in which either case will generate a lot of excessively routed ring sequences. Section 8.4 presents test case results show that using a +2 for ring sequence limit is indeed sufficient to encapsulate the optimal solution most of the time.

As previously discussed, RS-R is unlikely to miss the optimal solution in practice because of high correlation of number of rings in a ring sequence. However, it is, strictly, still possible for RS-R to result in a sub-optimal solution. A worst-case conceptual scenario for RS-R would be the following. The set of ring sequences that is within the limit (e.g. minimum ring sequence length + 2) consists of rings that are physical wide apart and the path has to traverse more node and longer distances, whereas the ring sequences that are short in physical distance and traverse less nodes, are the ones with ring sequence length higher than the limit. For example in the diagram below, although ring sequence 1-6 has less hops than 1-2-3-4-5, the later is a better choice because it actually traverse less equipment. Note that this kind of situation occurs rarely.

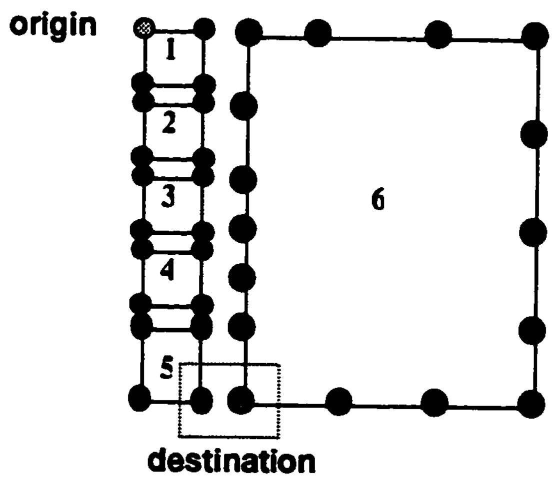


Figure 6.1: Worst Case Sample Scenario for RS-R

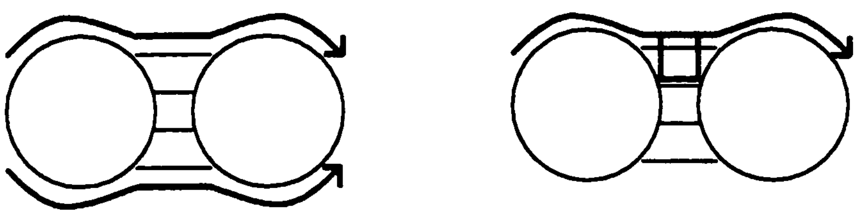
6.3. Path Constructions Reduction (PC-R)

6.3.1 Intuition and Approach

As demonstrated in section 5.3.1, the dominant contribution to the solution space estimation comes from the gateway arrangement. The effect is exponential in the number of common nodes between adjacent rings. Therefore, reducing the number of gateway combinations should be the key to drastic improvements in run-time. However, it is necessary to ensure that the gateway arrangement combinations retained still lead to good quality solutions.

In this method, the objective is optimized locally, hoping that it will result in a good global solution. In our case, the local optimization is performed at inter-ring connections where the choice of the two gateways is considered.

For each sub problem of determining the gateways arrangement, we will be considering a specific MN/DF sequence as given. Partial segments of a ring sequence must be considered. For the DF-DF sequence segment (figure 6.2), using the outer-most gateways will result in better availability and lower cost because less equipment is used. This is because if we choose to use any other gateway configuration, either of the path segments will have to traverse past the outer-most gateway into the inner nodes. And the equipment in that outer-most gateway is used as pass-through, which is redundant and therefore result in a higher cost.



DF-DF Assignment **MN-MN Assignment**

Figure 6.2: Path Constructions Reduction for DF-DF and MN-MN Path Segment

For the MN-MN sequence segment (figure 6.2), although the length of the continuous route does not have any impact on the availability for the inter-ring unavailability calculation, the shorter it is, the lower will be the cost. Even when only optimizing unavailability, using two gateways with a longer distance does not give us any benefit. On the other hand, for the SF signal, the shorter it is, the better for cost. Therefore, a cost-based criterion for MN-MN is to use the SF signal to the closest gateway since this is probably without impact on cost. Furthermore, we will stipulate that the secondary gateway, will always be the one closest to the already chosen primary gateway. This can only reduce cost and have no impact on availability.

For the DF-MN sequence (figure 6.3), to optimize availability, wider separation (called wide entry/egress) between gateways is actually desirable because it improves both DF ring availability, and has no impact on the adjacent MN ring because it is only the continue signal which is longer. If optimizing cost, there is not much difference between using adjacent or widely separated gateways because, either way, the amount of equipment deployed is the same. For better DF unavailability, widely separated gateways should also be chosen. MN-DF sequences the mirror image of the DF-MN, and the same criteria should be used. In summary, for an inter-ring connection with DF applied on one ring and MN on the other ring, then it is desirable to use a widely separated gateways, either in optimizing cost or availability.

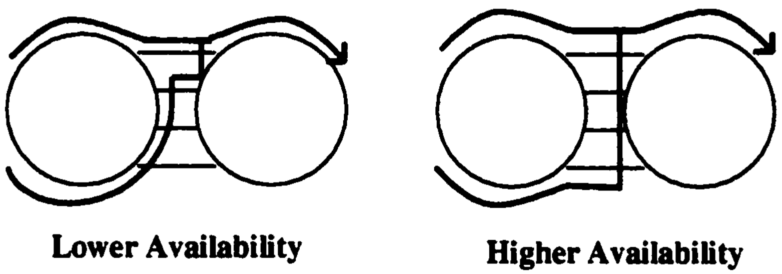


Figure 6.3: Path Construction Reduction for DF-MN or MN-DF Path Segments

6.3.2 Run-time and Solution Optimality

By using the two principles just covered for path construction reduction, the number of gateways arrangement is essentially reduced to a constant 1. However, the amount of overhead processing is increased substantially for "parsing" the ring sequence to abstract the problem information as just described. The run-time is substantially reduced, however, at the expense of a potentially sub-optimal solution because of the number of skipped gateway arrangements. The following is the updated path construction estimate formulation for PC-R.

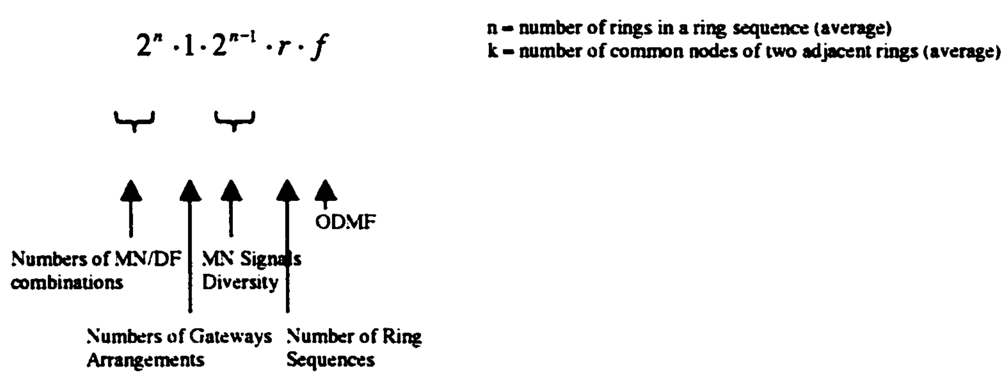


Figure 6.4: Path Constructions Estimation for Path Construction Reduction

And for PC-R, it is also possible for it to generate sub-optimal result. However, the 'worst-case' scenario is more difficult to identify. Moreover, we know that PC-R is a locally optimized method in which that particular inter-ring connection is optimized, but it may not be globally optimal for the end-to-end path. For instance, for a MN-MN assignment for a particular inter-ring connection, say if the single fed signal flows on clockwise direction, which is the shorter path, to the next possible gateway node X for inter-ring connection to the next ring. And the counter-clockwise path that is longer, leads to the next gateway node Y. However, the gateway node Y connects to the next ring where the destination node resides. On the other hand, the shorter clockwise path may need to traverse more than one ring before reaching the destination node. This is result in a sub-optimal solution because PC-R locally optimize the path segment on one ring, however, it affected how the path is routed overall.

6.4. RS-R and PC-R Combined

This method essentially applies both methods simultaneously during the path search. This method will reduce the run-time even further because of reductions in both RS and PC. On the other hand, the quality of the solution will remain the same as the PC-R's in most cases. This is because if the RS reduction does not eliminate any 'quality solution' in most cases, then using RS-R and PC-R will at worst result in a solution with PC-R quality, but with the run-time further improved.

n = number of rings in a ring sequence (average)
k = number of common nodes of two adjacent rings (average)

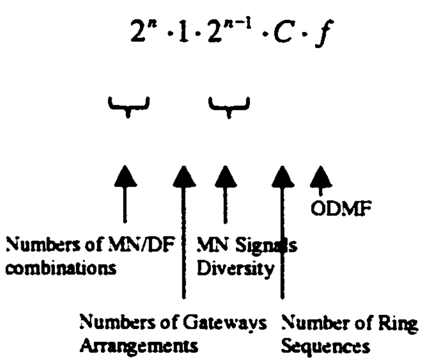


Figure 6.5: Path Constructions Estimation for RS-R / PC-R combined

7. Software Prototype: OPTICA

7.1. Background

In this section, we describe the software application developed for this research - OPTICA, in terms of the development environment, software architecture and how to use utilize it for path planning. The software prototype is intended to implement the heuristic search strategies just covered and proposed in this thesis. The software is called OPTICA: Optimizing Path provisioning Tool Integrating Cost and Availability. The following are some desirable properties of OPTICA:

- Graphical representation of the network topology, ring overlay, related network information and path search result
- Ease of use and intuitive user interface
- Software extensibility and reusability

In developing such a highly desirable, yet complex software package, we decided to use the RingBuilder Interactive (RBI) as the basis of OPTICA. RBI is a tool for automating ring network design, which was developed by the Network Systems Group in TRLabs. The language of choice is Java, an object-oriented language, and the existing architecture is extremely reusable in the development of OPTICA. Basic operations, like reading in a network data file, graphical manipulation of the network, underlying data structure...etc. are all implemented and fully applicable to this application. Thus RBI was used as a base application with OPTICA on top of it as an additional feature, fully utilizing all the existing data structures that are applicable.

7.2. Software Architecture

7.2.1 Development Environment

RBI is being developed under Windows NT 4.0 and using Java 1.2. Although Java is claimed to be platform independent, using Windows NT 4.0 in developing OPTICA is more consistent with RBI's implementation.

7.2.2 Architecture Overview

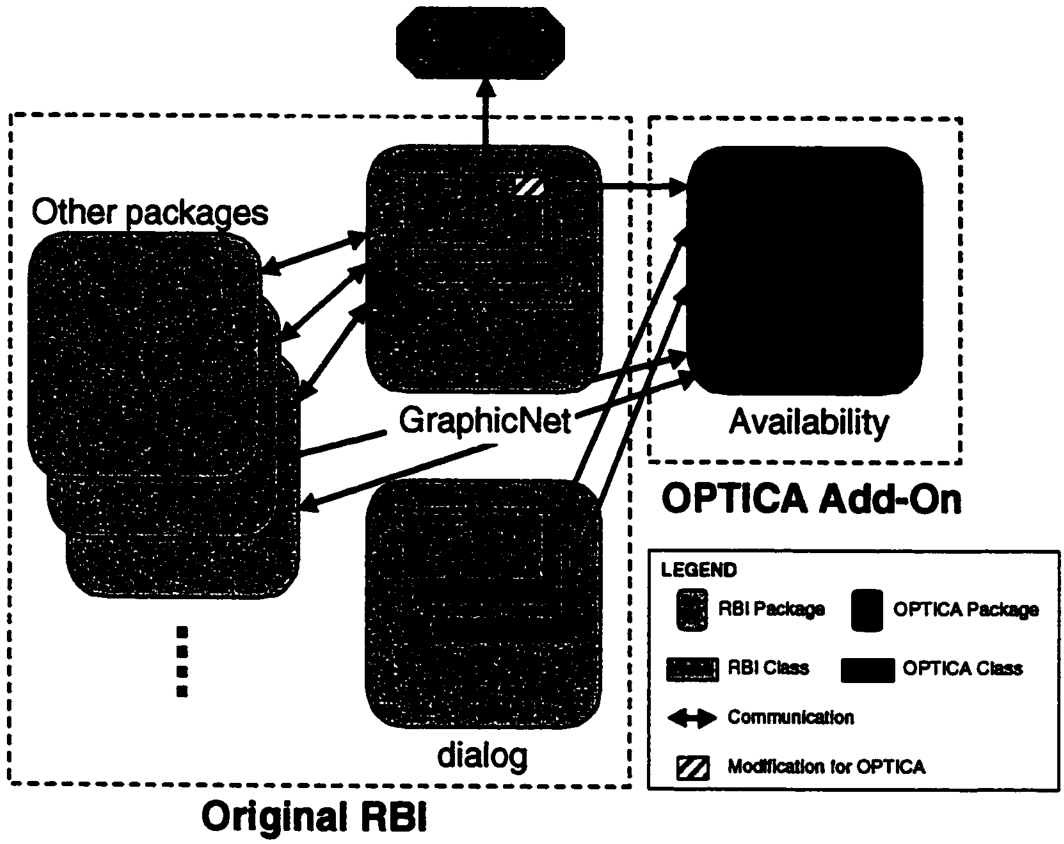


Figure 7.1: Architecture Overview

Figure 7.1 shows the overall architecture in terms of the interaction between RBI and OPTICA. A "class" is a source file that embeds an object's description (i.e. its data structures and related operations). A "package" is a group of "class" files that are logically related in the same domain. For example, the "GraphicNet" package in figure 7.1 contains all the classes that are used to display and control the main graphical network representation and the main window of the application. The "dialog" package contains all the pop-up sub-windows and dialog boxes in the application. The package for OPTICA is named as "Availability".

The OPTICA add-on is designed in such a way that minimal source code change is required to integrate RBI and OPTICA. Continuous development of RBI is independent of the development of OPTICA, yet it expected that integration is possible even after several cycles of software release. This is achieved by implementing OPTICA mainly in the Availability package. Only a small change is added to the main application class RBI in the GraphicNet package so that the path provisioning option will be available in the menu bar of the main RingBuilder window. All other operations are performed back in the Availability package. No change is made in any other class in RBI. However, in order to be consistent, all the classes for dialog pop-ups in OPTICA are placed in the dialog package, but this does not introduce any side-effects to either RBI or OPTICA.

7.2.3 Data Structure

The newly-defined data structures are all placed inside the Availability package. The major classes being introduced are listed in figure 7.2. The dotted lines group classes with similar functions together.

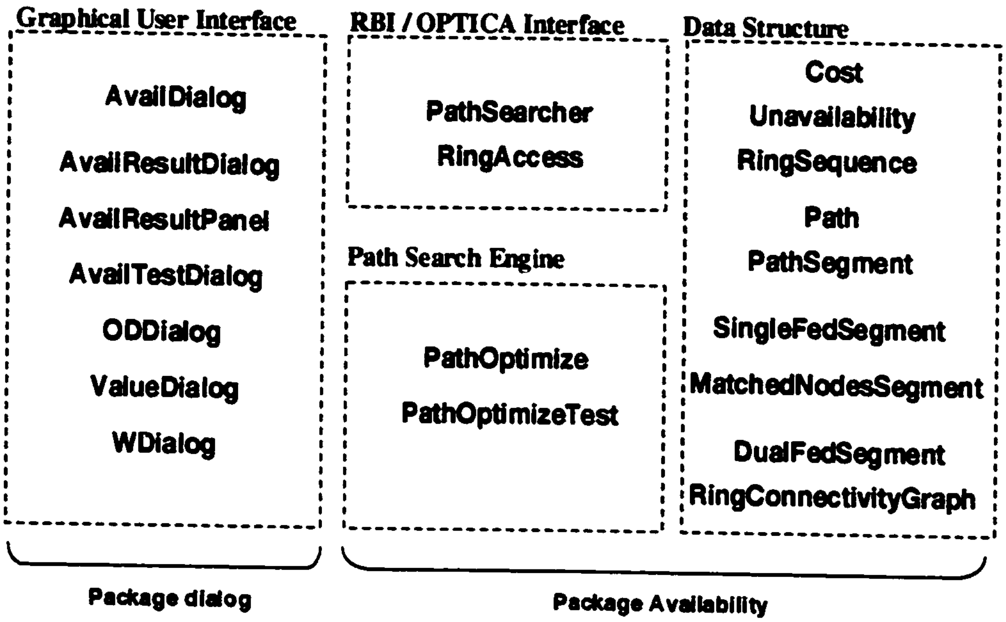


Figure 7.2: Classes for OPTICA

The *GUI* classes are responsible for asking for user input, issuing warnings and displaying results. The *RBI/OPTICA interface* is the link between RBI ("Modification for OPTICA" in figure 7.1) and OPTICA. The *path search engine* is the core of all the path search strategies / heuristics. The data structure group contains the main structures that enable path construction and provisioning.

7.2.4 Path Search Engine

The following is a simplified flow chart for the core of OPTICA: the path search engine. It begins with the user requirement input, and develops path constructs of each path type sought in turn until an answer is attained and displayed. Note that the alternative process flow indicates the use of RS-R and/or PC-R problem reduction heuristics depending on user preference; otherwise, it will perform an exhaustive search. Subsequent sections of this chapter describe each of these blocks in more detail.

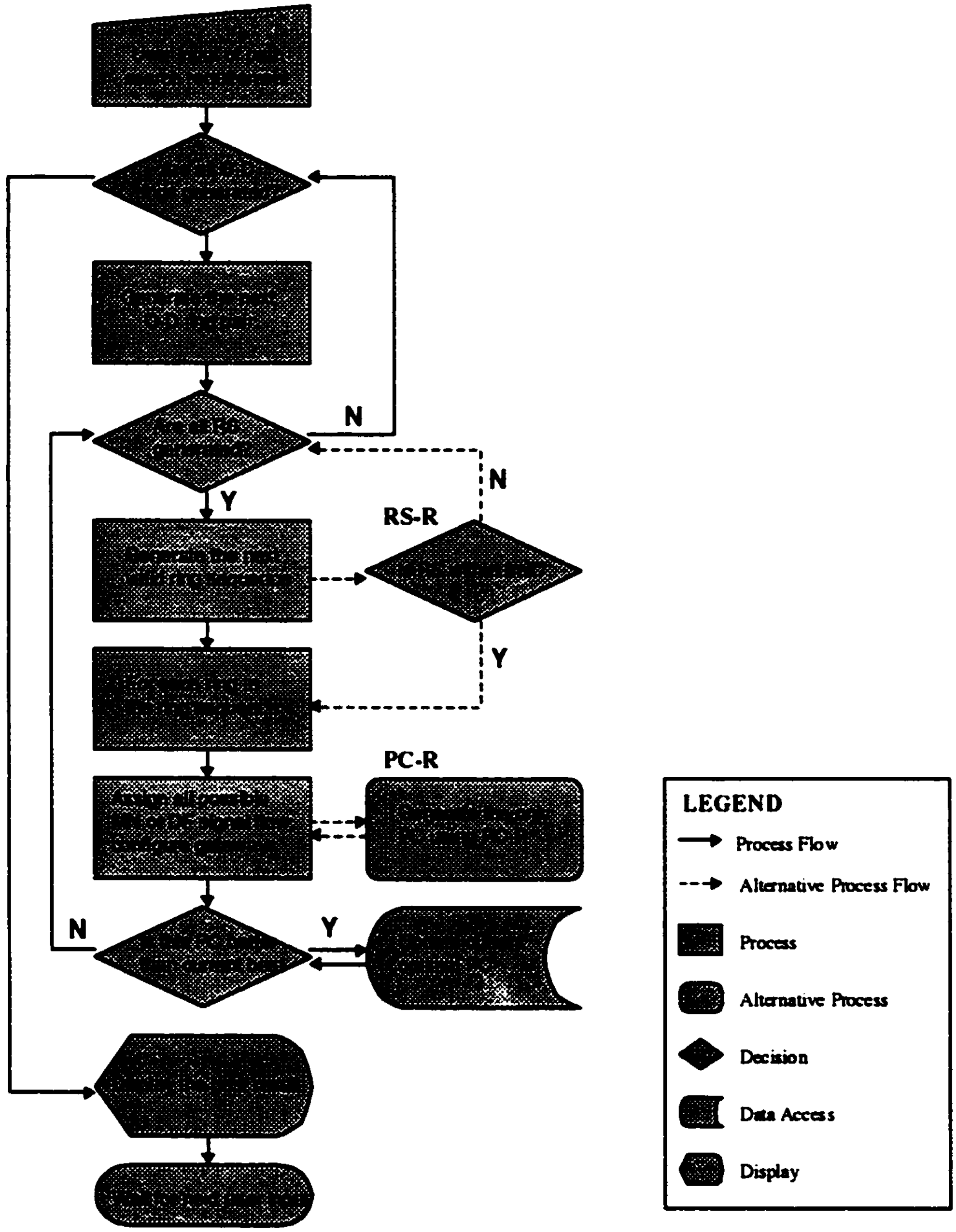


Figure 7.3: Flow Chart for the Path Search Engine

7.3. Graphical User Interface and Operation

In this section, the actual look and operation of the software prototype will be presented. The following is the main window view after opening an existing network file with ring set overlay.

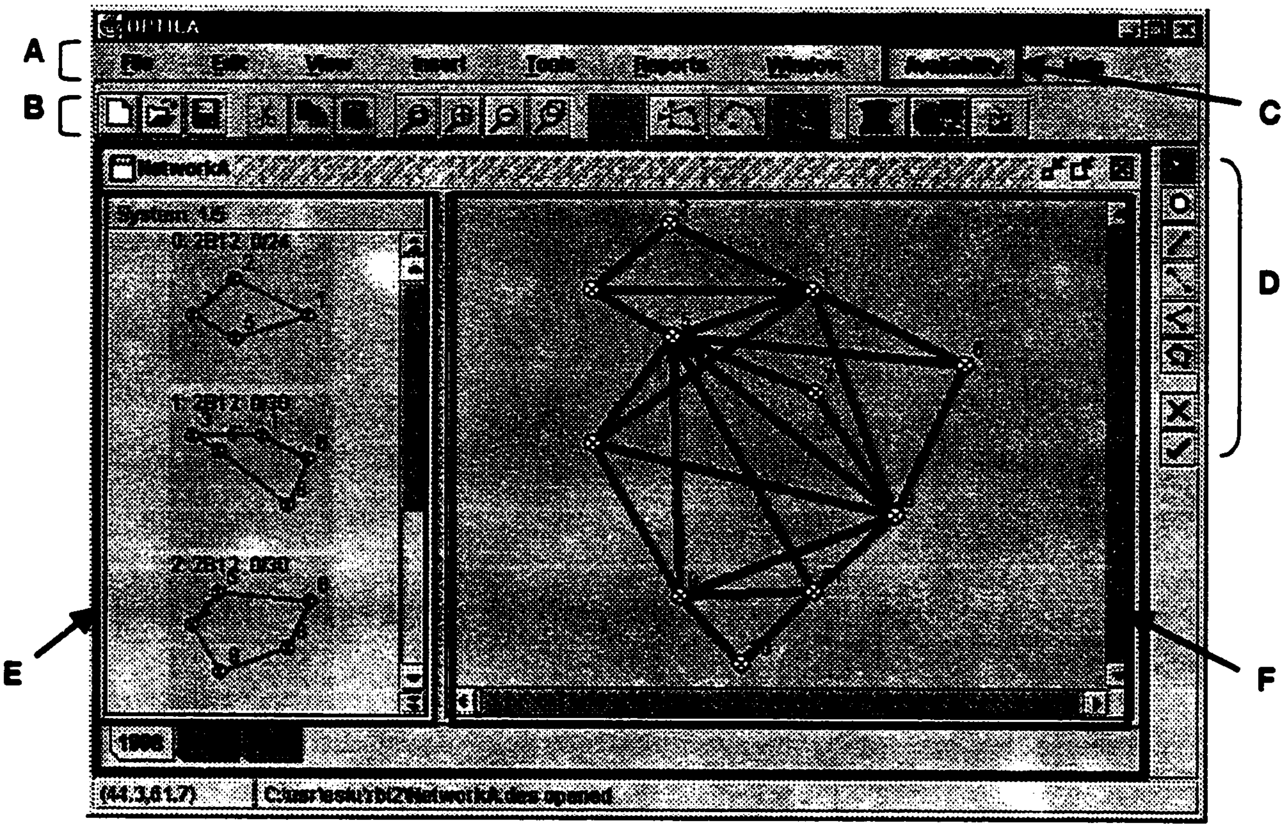


Figure 7.4: OPTICA Main Window

Area A indicates the menu bar, which includes all the function selections of RBI and OPTICA. The buttons in area B are basically a subset of the functions in area A which are more frequently used. Area C indicates the **Availability** option, which is the link from RBI to OPTICA functions. Area D is the list of basic functions that allow the user to graphically manipulate (i.e. create, modify, delete) the network topology or the ring overlay network. Area E displays the current ring-set, and area F displays the current network graph. Areas E and F together compose a design window which encompasses all the data for the current network design. The user can create a network using tools from area D, or import from a text file of network topology data in a TRLabs standard format called SNIF (standard network interface format). The user can load and save the design using the File menu in area A. With an existing ring network design, the network planner can apply path search functionality from OPTICA.

Figure 7.5 shows the functionality currently offered by OPTICA. "Path Search" is the major function that has been implemented. It can search for an end-to-end path with specific user requirement in terms of cost and availability. "Estimate Run Time" requires input similar to that of the "Path Search" (i.e. O-D pair), but this function is intended only for displaying the estimated run-time. Because of time constraints, it can currently only display information regarding the estimated total number of path constructions (figure 7.6) in a Java console window. "General Network Information" displays some useful information regarding the network (figure 7.7). These two functions are not yet integrated into the GUI, so they are currently displayed under the execution text box.

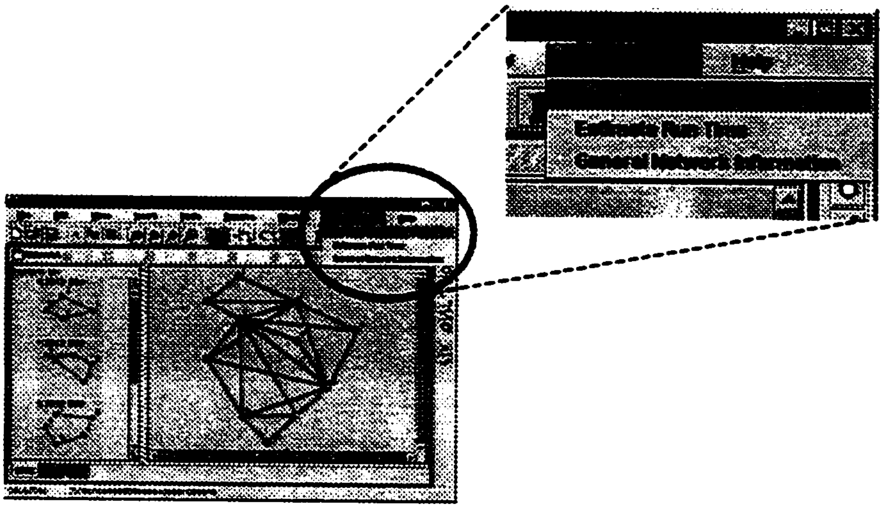


Figure 7.5: OPTICA Functionality Menu

Figure 7.6: OPTICA Run-time Estimation

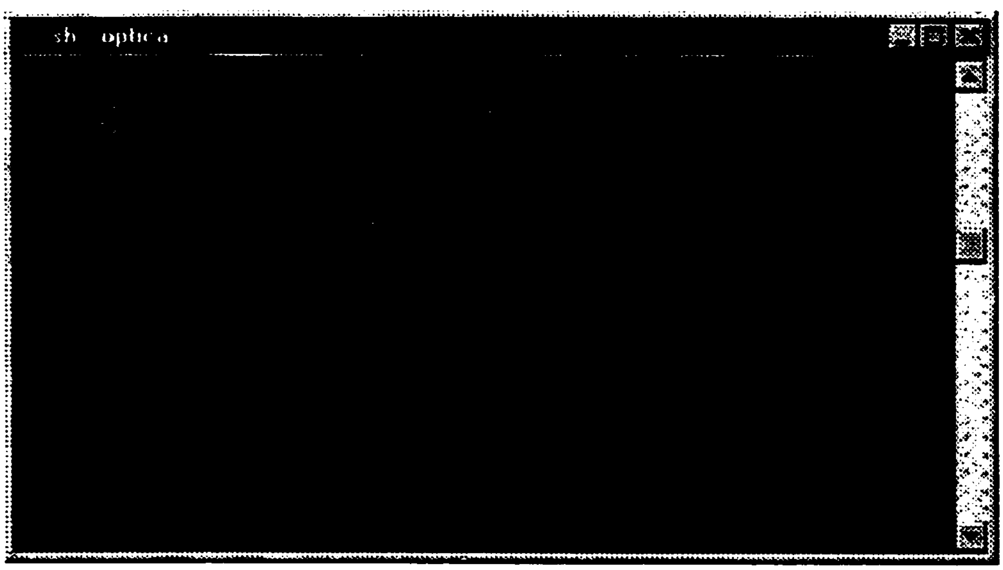


Figure 7.7: OPTICA Network Information

The "Path Search" operation will now be described in detail. First, "Availability" is selected from the menu bar, then "Path Search" as in figure 7.5. A dialog box will pop up asking the user to specify the path configuration to consider (e.g. MN/DF combined), the path optimization criterion (e.g. minimize cost), and the search reduction, if any strategy (e.g. RS-R). Figure 7.8 shows the actual pop up dialog box.

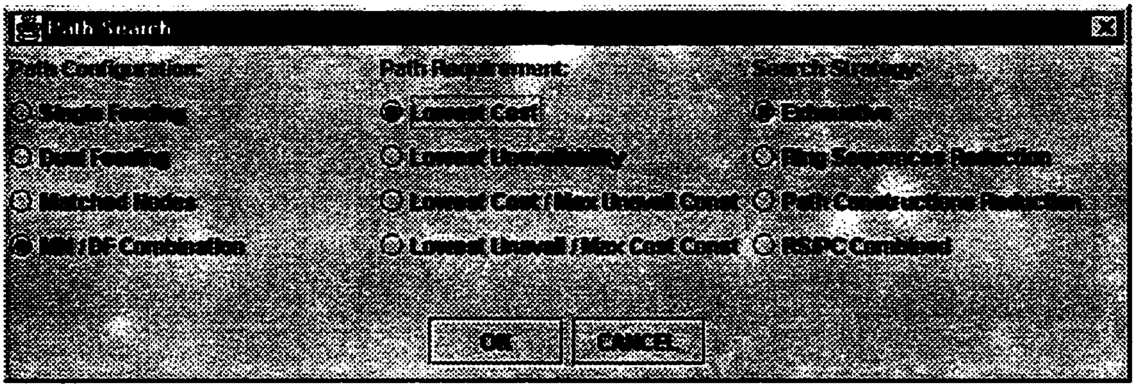


Figure 7.8: Path Search Requirement Dialog Box

After specifying the three options, the user will be prompted for a constraint value (i.e. maximum cost or unavailability constraint), if the 3rd or 4th Path Requirement is chosen in the dialog box (figure 7.8). The dialog will look like this:

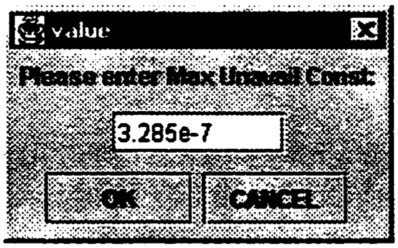


Figure 7.9: Constraint Value Dialog

Finally, it will prompt the user for the O-D pair between which the new path is to be provisioned. This will be displayed as two selection lists with their corresponding node names (Figure 7.10).

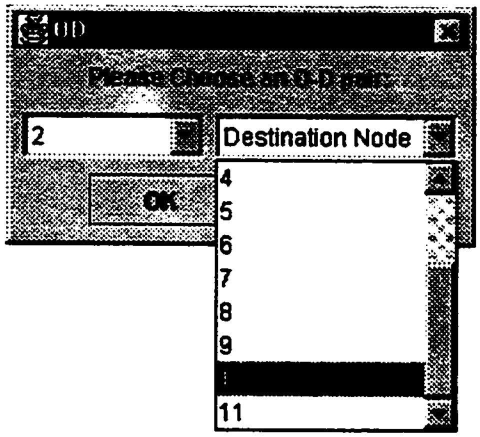


Figure 7.10: O-D Pair Dialog

Then the search engine is executed. It will search for the most desirable path based on the user requirements and the procedure described in section 4.3 and chapter 6. It reports the most desirable result found in the following format:

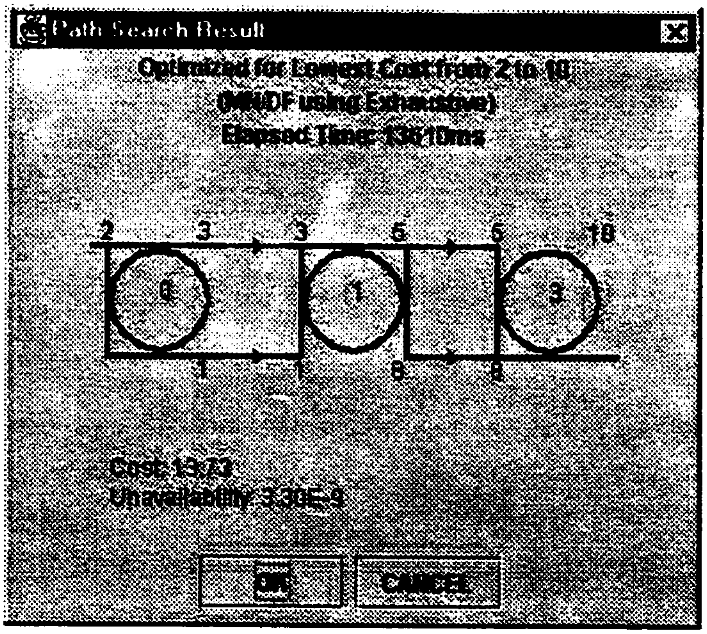


Figure 7.11: Path Search Result Display Window

The report displays an abstraction of the resulting path design in a simplified view showing the ring sequence with the gateways configuration and the signal flow. For example, the above result is an instance of a DF-MN-MN path construction in the network of Figure 7.4. Ring

0 contains the origin node 2. Ring 0 interfaces to Ring 1 at nodes 3 and 1. The MN signal takes the shortest route between node 3 to 5 in Ring 1. Then Ring 1 interfaces to Ring 3 at node 5 and node 8. The signal then takes the short route from node 8 to node 10, which is the destination node. This is a fully provisioned end-to-end path, originates from node 2 and terminates at destination node 10. All the inter-ring connection information are captured.

The path search result window also restates the user-specified requirements and information. In addition, it shows the path cost, unavailability, and elapsed time of the search.

8. Case Studies Using OPTICA

8.1. Background

8.1.1 Case Studies Objectives

This section will use OPTICA to perform test cases on path provisioning in various networks, to study various subjects in our research. The following are the objectives of the studies:

1. To verify the theory for the complexity of the problem.
2. To test and demonstrate the correctness of the software prototype.
3. To characterize the proposed heuristics and demonstrate that they can achieve optimal or near-optimal solutions while having significant improvement on run-time when compared to the exhaustive solution in the following two cases:
   - Single objective (i.e. minimizing cost or unavailability)
   - Single objective with constraint (i.e. minimizing cost with maximum unavailability constraint, or minimizing unavailability with maximum cost constraint)

By the term "exhaustive solution", we mean a solution that is extracted out of all possible solutions in a problem domain by complete exhaustive enumeration. For instance, if 'minimum cost' is the objective, then all possible path constructions are evaluated and the path with minimum cost is returned as the exhaustive solution. The term 'exhaustive solution' thus describes the most 'brute-force' method of optimization (i.e. exhaustive search) leading to assured optimality of the solution.

8.1.2 Study Design

8.1.2.1 Choice of Networks

Three networks were chosen for the test cases: the "Bellcore" network, BTRL network and the Sprint network. They will be referred to as Networks A, B and C, respectively. Data from the first two networks were retrieved from the Network Systems Group in TRLabs, and the data for the last network comes from Sprint's web page. The following figures show their physical topology.

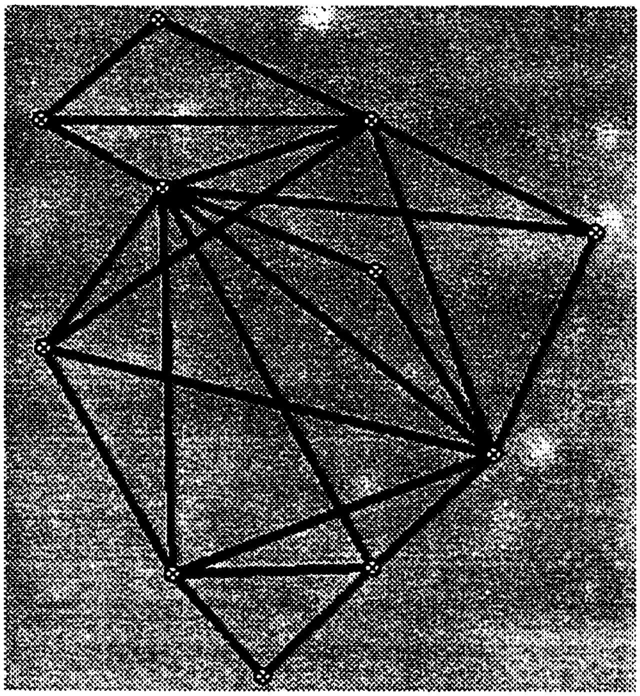


Figure 8.1: Network A (basis: a New Jersey LATA)

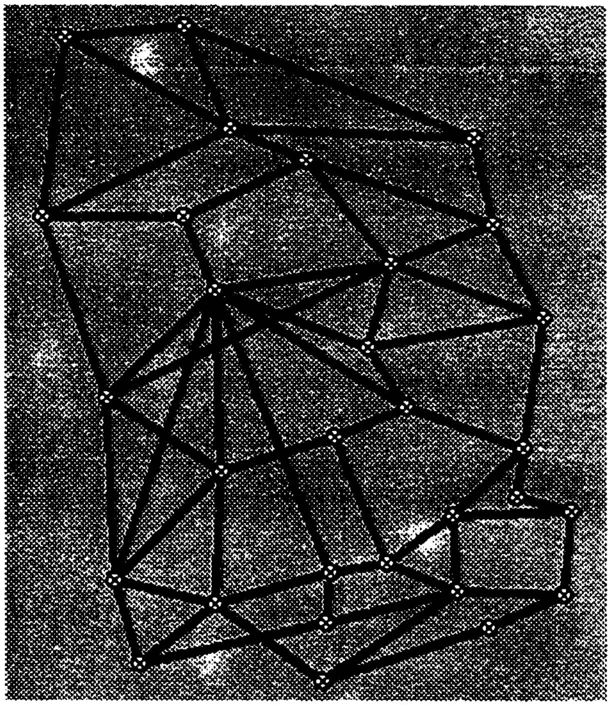


Figure 8.2: Network B (basis: BT backbone network)

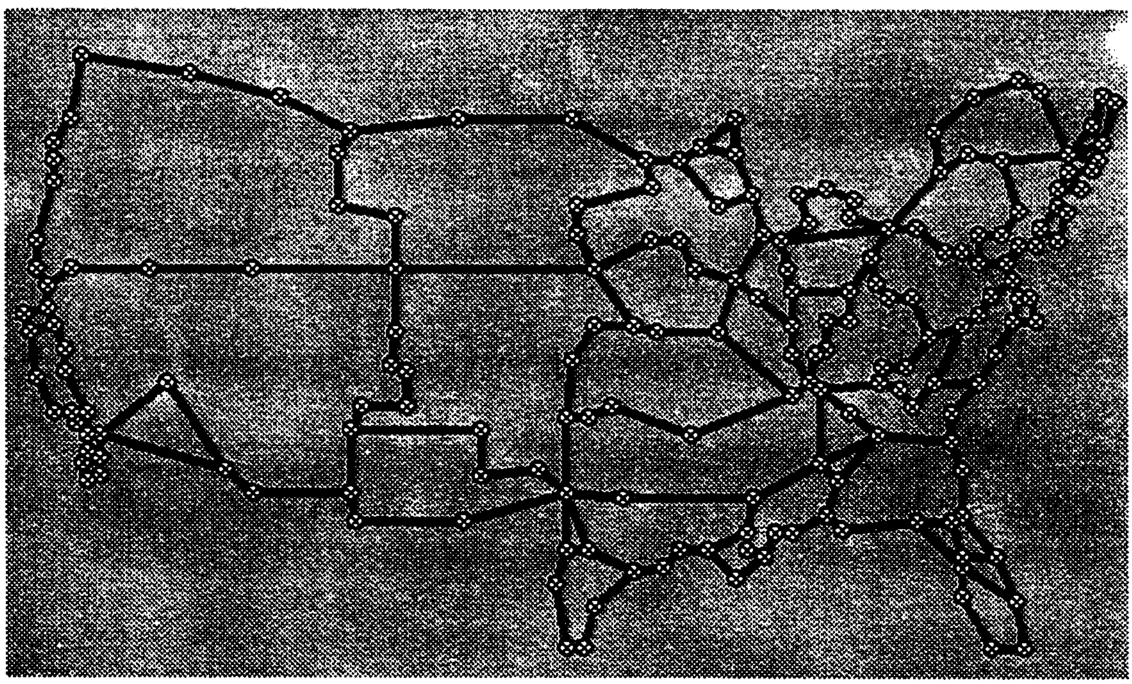


Figure 8.3: Network C (basis: US Sprint)

The three networks are different in their properties and the following table is a summary of those data:

	Nodes	Spans	Rings	A.N.D.	A.S.D.	A.R.C.
NetworkA	11	23	5	4.18	6.52	2.3
NetworkB	30	59	11	3.93	40	2.04
NetworkC	188	236	41	2.51	156.08	2.4

A.N.D.: Average Nodal Degree
A.S.D.: Average Span Distance (km)
A.R.C.: Average Ring Connectivity

Table 8.1: Simulation Networks' Properties

Network A is basically a metropolitan network with a small number of nodes, spans and rings, and also a small average span distance. Networks B and C are long-haul networks with longer span distances. These three networks were chosen for conducting a regimen of test cases to ensure that the study covered both metropolitan and long haul networks, and also small and large networks.

The ring overlay sets were designed manually, with the basic criterion being total coverage of the network topology. Therefore we placed rings on the network topology in a logical and planar fashion to simply cover the topology. That is why the average ring connectivity (A.R.C.) does not vary much (2.3, 2.04 and 2.4). Our study should be able to generate similar results for ring network designs using realistic demands, however, we designed the ring overlay sets in a planar fashion for ease of analysis and visualization.

8.1.2.2 Choice of Origin-Destination Node Pairs

Nine different pairs of origin-destination node pairs (referred to as O-D pair) were chosen for path provisioning design trials in each network. These are classified into three categories: *short O-D pairs*, *medium O-D pairs*, and *long O-D pairs*, with three O-D pairs in each category. The origin and destination nodes of an O-D pair are physically located at short, medium or long distances (in terms of hops) apart respectively. This was done to ensure that we have a variety of ring sequence lengths between the O-D pairs in the simulations. All O-D pairs in the table are specified in Appendix A.

8.2. Problem Complexity

8.2.1 Ring Sequences

The following table shows how many distinct ring sequences are options for paths between each O-D pair. Note that ODS1 to ODL3 in Network A are different from the ODS1 to ODL3 in Network B and also from Network C (Appendix A). The "Shortest Ring Sequence" column shows the hops of the shortest possible ring sequence for the O-D pair. For example, for the first short O-D pair (ODS1) in Network A, the origin is two rings (i.e. 1 hop) away from the destination.

	Network A			Network B			Network C		
	Total Ring Sequences	Shortest Ring Sequence	O-D Rings Multiplying Factor	Total Ring Sequences	Shortest Ring Sequence	O-D Rings Multiplying Factor	Total Ring Sequences	Shortest Ring Sequence	O-D Rings Multiplying Factor
ODS1	5	2	2	19	2	2	25900	3	1
ODS2	20	2	4	54	2	4	49154	3	2
ODS3	10	2	2	7	2	1	37037	3	2
ODM1	10	3	2	28	3	2	91612	4	2
ODM2	15	3	3	67	3	4	14082	5	1
ODM3	10	2	1	88	4	4	91388	4	1
ODL1	5	3	1	90	6	2	45393	6	1
ODL2	5	3	1	222	4	6	170756	7	1
ODL3	5	3	1	129	4	4	139386	7	2

Table 8.2: Ring Sequences Information on the Three Simulation Networks.

The data on the "Shortest Ring Sequence" column is obtained by applying a K shortest paths algorithm (similar to shortest path algorithm except it return all the shortest paths with the same length) on the corresponding RCG.

8.2.2 Path Constructions

Using the data from Table 8.2, calculations can be performed to estimate the number of all possible path constructions for each O-D pair using the formulations described in 5.3. Note that the estimate is not based on the unique details of each ring, rather it uses conservative values. This is because the number of rings in each ring sequence is a unique value, and the usable gateways for each inter-ring connection are also unique. It is hard to estimate these values without actually enumerating all of the possibilities. Therefore, averages are used in place of all unique values to come up with a rough estimation. The following exercise is only used to confirm the magnitude of the problem, and to identify the subset of cases where we may reasonably expect to be able to obtain globally optimum results by exhaustive search for reference purposes and comparison to the restricted search results.

	Network A	Network B	Network C
ODS1	2.700E+02	1.026E+03	6.294E+06
ODS2	2.160E+03	5.832E+03	2.389E+07
ODS3	5.400E+02	1.890E+02	1.800E+07
ODM1	4.860E+03	1.361E+04	4.007E+08
ODM2	1.094E+04	6.512E+04	2.772E+08
ODM3	2.700E+02	7.698E+05	1.999E+08
ODL1	1.215E+03	3.189E+07	8.041E+09
ODL2	1.215E+03	2.913E+06	2.722E+11
ODL3	1.215E+03	1.128E+06	4.445E+11

Table 8.3: Path Construction Estimation #1.

Table 8.3 uses conservative parameters to estimate the number of total path constructions. Ring sequence length is set to the shortest ring sequence length data from Table 8.2. The number of gateways between two rings is set to the minimum value 2. Almost all of the path constructions are manageable for Networks A and B. However, if 1 microsecond is required to evaluate each path construction, the worst possible case of ODL3 in Network C will require 4.445e8 seconds, or about 14 years to evaluate all path constructions.

	Network A	Network B	Network C
ODS1	1.944E+04	7.387E+04	1.813E+09
ODS2	1.555E+05	4.199E+05	6.880E+09
ODS3	3.888E+04	1.361E+04	5.184E+09
ODM1	1.400E+06	3.919E+06	4.616E+11
ODM2	3.149E+06	1.876E+07	1.277E+12
ODM3	1.944E+04	8.868E+08	2.302E+11
ODL1	3.499E+05	5.877E+11	1.482E+14
ODL2	3.499E+05	3.356E+09	2.007E+16
ODL3	3.499E+05	1.300E+09	3.277E+16

Table 8.4: Path Construction Estimation #2.

Table 8.4 uses less conservative parameters for the estimation. Ring sequence length is set to the shortest ring sequence length times 2. The number of gateway options between two rings is set to 4. It can be seen that even some of the Network B cases become unmanageable. As mentioned in the analysis in this section and 5.3, a heuristic approach is absolutely necessary in many cases because of the size of the solution space.

8.3. MN-only, DF-only Unconstrained Path Study and Verification

Before attempting to tackle the path search problem using MN/DF combined configurations, MN-only and DF-only test cases will be considered. In this section the optimization of MN-only and DF-only paths in terms of availability will be studied unconstrained by cost, and some of the availability calculation performed by the software prototype will be verified. In addition, a case study on the MN-only and DF-only effect on availability will be presented.

First, consider figures 8.4 and 8.5. The O-D pair is nodes 1 and node 28 in Network B (figure 8.2). OPTICA was used to perform a path search for each of the configurations: MN-only and DF-only, minimizing their unavailability without any cost constraint. For both cases, we have used RS-R instead of exhaustive method because exhaustive run-time is too long, and the optimality for RS-R is extremely good (will be discussed in later sections). In the result, the MN-only and DF-only paths use exactly the same ring sequence; however, there are two interconnections in which their gateway settings are different. Because there are two interconnections involved, there are three rings that are affected, that is, rings 4, 8 and 9.

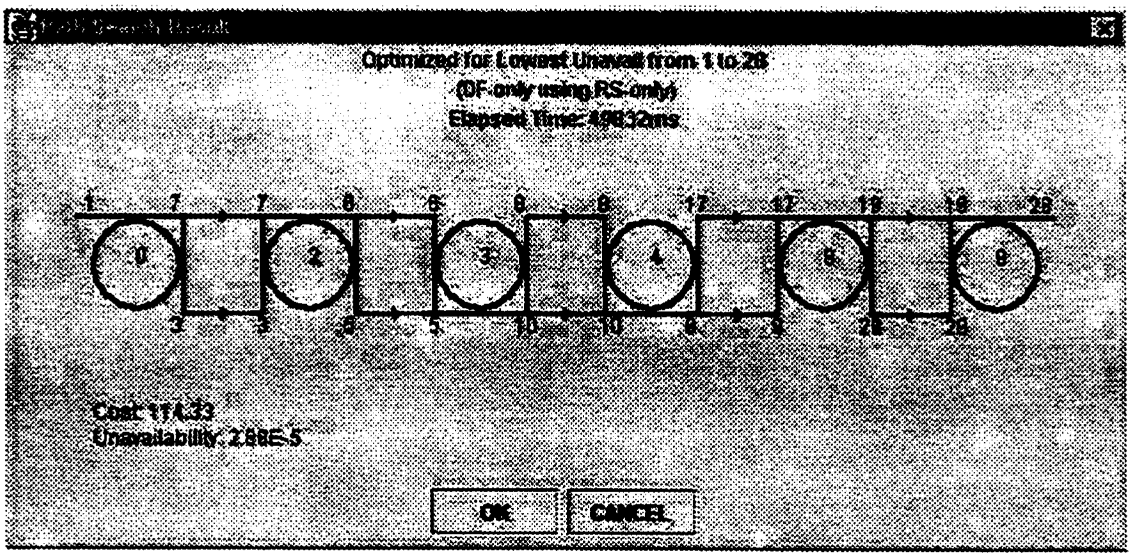


Figure 8.4: MN-only Path

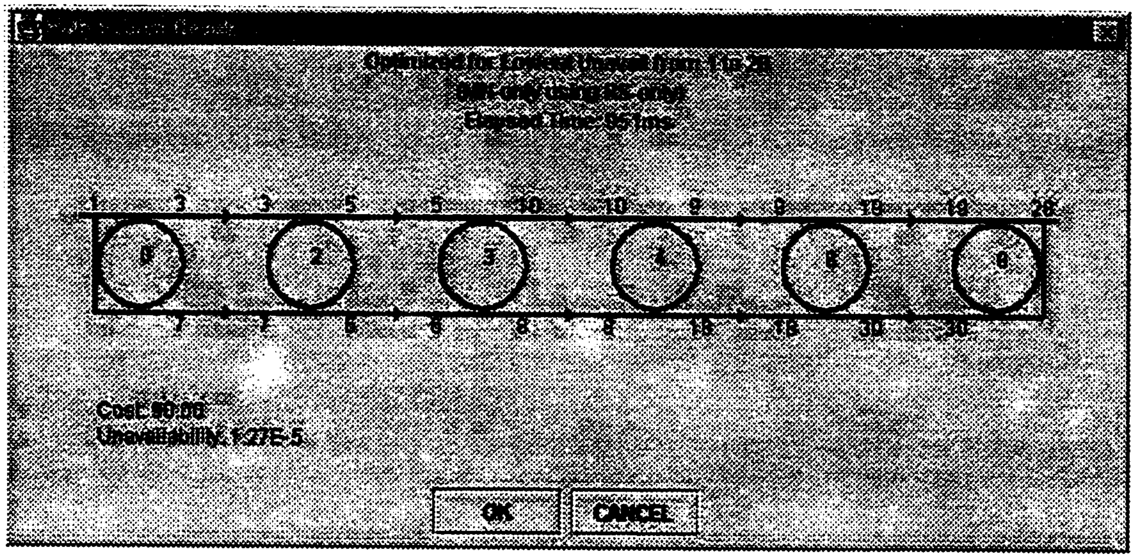


Figure 8.5: DF-only Path

Both test case results from figures 8.4 and 8.5 use RS-R to reduce the run-time. Otherwise, we would not be able to obtain any result. And for RS-R, it will generate the optimal answer in most cases (discussed in later sections in this chapter). Although there is a slight difference between the two path constructions, the cost and unavailability for pure DF is much lower than MN. This is a confirmation and validation that DF-only path construction can cost less and have lower unavailability than MN-only path construction, as suggested in [55].

Separate calculations in Matlab (Appendix B) have been duplicated to validate the unavailability calculation for this test case. The calculation has been double-checked that the they are correct for both cases (MN-only and DF-only). The results are listed in figure 8.6. The values for r1, r2...etc. are the Td_1 or Td_2 calculation for each individual ring. The U on the last line is the total end-to-end unavailability. As can be seen in CASE #1, the results are exactly the same as from the snap shot. Note that *Unl* and *Una* are set as constant 1e-5 for all cases.

```
CASE #1: SPAN LENGTH = 100   Usl = 1E-5
MN-only                   DF-only
-------                   -------
r1 = 4.1106e-06           r1 = 3.1415e-06
r2 = 3.0804e-06           r2 = 1.0609e-06
r3 = 4.1106e-06           r3 = 2.1012e-06
r4 = 7.2012e-06           r4 = 1.0609e-06
r5 = 6.1710e-06           r5 = 2.1012e-06
r6 = 4.1106e-06           r6 = 3.1415e-06
ee = 6.0000e-10           redun = 1.2000e-09
inter = 7.0000e-09        inter = 5.7600e-08
U = 2.8792e-05            U = 1.2664e-05
```

Figure 8.6: MN-only. DF-only Simulation Case #1 (MatLab verification)

The next two cases use exactly the same parameters as CASE #1 except the span distances are changed from 100 to 10 and 1, respectively.

```
CASE #2: SPAN LENGTH = 10    Usl = 1E-5
MN-only                   DF-only
r1 = 5.1600e-08           r1 = 4.5500e-08
r2 = 3.8400e-08           r2 = 1.6900e-08
r3 = 5.1600e-08           r3 = 3.1200e-08
r4 = 9.1200e-08           r4 = 1.6900e-08
r5 = 7.8000e-08           r5 = 3.1200e-08
r6 = 5.1600e-08           r6 = 4.5500e-08
ee = 6.0000e-10           redun = 1.2000e-09
inter = 7.0000e-09        inter = 5.7600e-08
U = 3.7000e-07            U = 2.4360e-07
```

Figure 8.7: MN-only. DF-only Simulation Case #2 (MatLab)

```
CASE #3: SPAN LENGTH = 1     Usl = 1E-5
MN-only                   DF-only
-------                   -------
r1 = 2.1000e-09           r1 = 3.2000e-09
r2 = 1.5000e-09           r2 = 1.6000e-09
r3 = 2.1000e-09           r3 = 2.4000e-09
r4 = 3.9000e-09           r4 = 1.6000e-09
r5 = 3.3000e-09           r5 = 2.4000e-09
r6 = 2.1000e-09           r6 = 3.2000e-09
ee = 6.0000e-10           redun = 1.2000e-09
inter = 7.0000e-09        inter = 5.7600e-08
U = 2.2600e-08            U = 7.0800e-08
```

Figure 8.8: MN-only. DF-only Simulation Case #3 (MatLab)

As can be seen, the $U_{df} < U_{mn}$ in CASEs #1 and #2, but in CASE #3 $U_{df} > U_{mn}$; therefore, whether $U_{df} < U_{mn}$ or $U_{df} > U_{mn}$ depends on different situations. The key explanation is that, for dual failure outage in $Td_1()$ and $Td_2()$, intra-ring terms begin to overtake the path design considerations, changing the preferred construct. Consider the simple case of a 'square' ring with four nodes.

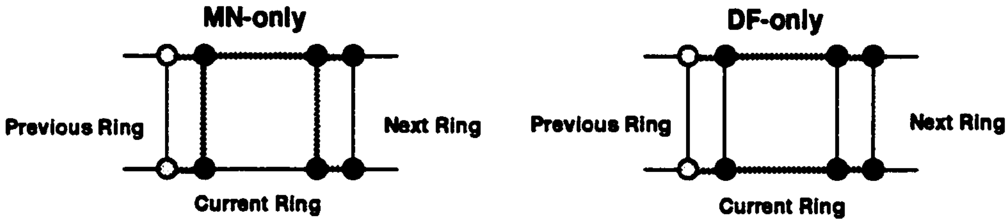


Figure 8.9: MN-only vs DF-only in Span Usage

Using the same ring and applying MN and DF, it can be seen that MN uses three spans and DF only two. When the span-span outage contribution is dominant (e.g. span distance is 100) in the U calculation, it makes a big difference (through $Td_{1/2}()$ intra-ring terms) compared to the case where span length = 1, in which span-span failure is only as 'important' as other failures (i.e. span-node or node-node), does MN show a better U value overall. Note that the inter-ring failure cases are not significant in the calculations of the three cases. The following three cases are duplicates of CASEs #1 to #3 except that the U_{sl} is changed from 1E-5 to 1E-6. Ring-by-ring details are omitted.

```
CASE #4: SPAN LENGTH = 100    Usl =1E-6
MN-only                     DF-only
U = 3.7E-7                  U = 2.4E-7

CASE #5: SPAN LENGTH = 10     Usl =1E-6
MN-only                     DF-only
U = 2.26E-8                 U = 7.08E-8

CASE #6: SPAN LENGTH = 1      Usl =1E-6
MN-only                     DF-only
U = 1.28E-8                 U = 6.42E-8
```

Figure 8.10: MN-only, DF-only Simulation Case #4, #5 and #6

As can be observed, by using U_{sl} = 1E-6, the span failure is not as dominant as before. Case #4 still shows $U_{df} < U_{mn}$, but in case #5 in which span length is 10, $U_{df} > U_{mn}$, as in Case #6 where span length is 1.

It can be concluded that the span length and span unavailability (U_{sl}) play a major role in deciding the path provisioning strategy to use in obtaining the best end-to-end unavailability in long-haul networks. Because of this, it is suggested that even using the exact gateway settings for both MN-only and DF-only, the same results would still be obtained for the outcome of span-span failures dominance. In conclusion, DF may, indeed, be better not only in terms of cost but also in terms of availability in certain situations.

8.4. MN/DF Combined Path Study and Optimality

In this section, we focus on the optimality of the MN/DF combined path solutions, generated using the proposed heuristics in Chapter 6. The results will be presented separately for minimizing cost and minimizing unavailability in each of the three test networks, with the four methods studied in this thesis, namely, exhaustive search, ring sequences reduction, path constructions reduction, and RS/PC combined reduction.

8.4.1 Cost Minimization Trials

First, only the trial runs completed within a four-hour limit will be considered. Some of the runs could not be finished within the time limit because of inherent problem complexity (see section 6.3.2). The other cases, however, provide an adequate sample of test cases to see the effect of reduction sequences on run-time and solution quality. A four-hour limit is set for all test trials as a practical time limitation. It also reflects that we are aiming that each path provisioning to be completed within a short period of time (e.g. a working day).

The most important question with regards to optimality is how the reduction heuristics affect the result compared with the optimal answer from exhaustive search. The following table summarizes how far the approximated answers are from the optimal answers, by showing the difference of the cost as a result of the particular heuristic method. A cell with value 0.00% means that the heuristic answer is exactly the same as the optimal answer from the exhaustive method, whereas a cell with 2.00% means that the heuristic answer costs 2.00% more than the optimal one.

	Network A			Network B			Network C			Overall
	ODS	ODM	ODL	ODS	ODM	ODL	ODS	ODM	ODL	
RS-R	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	-	-	0.00%
PC-R	0.00%	6.85%	0.00%	0.00%	2.83%	0.82%	0.89%	-	-	1.63%
RS/PC-R	0.00%	6.85%	0.00%	0.00%	2.83%	0.82%	0.44%	-	-	1.56%
Overall	0.00%	4.56%	0.00%	0.00%	1.89%	0.54%	0.44%	-	-	1.06%
Network Overall	1.52%			0.81%			0.44%			

Table 8.5: Effect of Reduction Heuristics on Solution Quality (Cost)

Note that the cells marked with a dash indicate that the optimal solution could not be obtained using the exhaustive method because the simulations were not completed within the time limit of four hours. Therefore, there is no optimal solution to compare with the approximated ones.

In all trial cases, RS-R is seen to be totally without effect on the overall difference in terms of cost. All the results using RS-R path designs are the same as the optimal. It is easy to explain in terms that RS-R eliminates the longer ring sequences that are extremely unlikely to contain a lower cost construction (i.e. ring sequences with excessive hops), otherwise it evaluates every possible path construction combinations for the ring sequence. This confirms the previously argued strong applicability of the ring sequence reduction principle. It can be seen that the RS limit set at +2 is sufficient for all cases in the simulations in order to obtain an optimal answer. In fact, almost all RS-R cases use the ring sequences of RS limit, and only a few (less than 5) cases uses RS limit + 1.

For PC-R and RS/PC-R, a 1.63% and 1.56% average cost increase respectively is observed in Table 8.5. However, if the average is calculated only for the solutions where the result *was* sub optimal, (excluding those where "heuristic = optimal" data points), the average cost increase is 5.388%. These approximate solutions are still very near-optimal and have practical use for network service provisioning. Although 100% solution optimality, like RS reduction, cannot be achieved, PC reduction still hits the optimal solution in 61.90% of the trial cases, and RS/PC-R hits the optimal solution 71.43% of the time. Theoretically (and in the results), the value for PC-R should be the same as for RS/PC-R given that RS-R had no deleterious effect. This is because if RS-R only eliminates the non-optimal solutions, applying PC-R is the same as RS/PC-R in terms of optimality. The reason why PC-R has a lower percentage in hitting optimal solutions is because some of the PC-R simulations were not completed within the four-hour limit. On some occasions, exhaustive search may not yield any result after the 4-hours time limit. This is because the search strategies are executed in a breath-first search manner, such that all possibilities are branched out on one level, before the next level is examined.

8.4.2 Minimizing Unavailability

As in 8.4.1, a table is now generated that summarizes how far the results from restricted search are from the optimal answers. Again, a cell with a value of 0.00% means that the heuristic answer is exactly the same as the optimal answer from the exhaustive method.

	Network A			Network B			Network C			Overall
	ODS	ODM	ODL	ODS	ODM	ODL	ODS	ODM	ODL	
RS-R	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	-	-	-	0.00%
PC-R	0.89%	7.32%	0.78%	0.00%	0.00%	6.62%	-	-	-	2.60%
RS/PC-R	0.89%	7.32%	0.78%	0.00%	0.00%	6.62%	-	-	-	2.60%
Overall	0.59%	4.88%	0.52%	0.00%	0.00%	4.41%	-	-	-	
Network Overall	2.00%			1.47%			-			1.73%

Table 8.6: Effect of Reduction Heuristics on Solution Quality (Unavailability)

The results are very similar to these in the previous section. Again, RS-R has no deleterious effect at all in terms of path design unavailability. The RS limit set at +2 was again sufficient for all cases in the simulation, in order to gain an optimal answer.

For PC-R and RS/PC-R reduction, there is a 2.60% increase in average unavailability as in Table 8.6. However, if only the increase for the approximated solutions is calculated, the average unavailability increase is 7.805%. Again, it could be concluded that the approximated solutions are near-optimal and have practical use for network service provisioning. PC-R and RS/PC-R hits the optimal solution in 71.43% out of all cases. In practice, an increase in unavailability of about 10% is insignificant. Notice that when comparing to single feeding path, the unavailability for dually protected path is at least 1000 times better because it can recover from all single element failure scenarios (and most dual elements failure).

8.5. Performance and Runtime

All trial cases were given an upper limit of four hours of run-time. Henceforth in reviewing our results, if a particular trial was not completed within the four-hour limit, then the best available results were gathered after four hours, and the search terminated. Note that the results must be compared separately in the context of complete results and incomplete results. The following table is a summary of the percentage of simulations completed within the time limit in all categories when minimizing cost.

	Network A			Network B			Network C			Overall
	ODS	ODM	ODL	ODS	ODM	ODL	ODS	ODM	ODL	
Ex	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	33.33%	0.00%	0.00%	70.37%
RS	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	0.00%	0.00%	77.78%
PC	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	33.33%	0.00%	0.00%	70.37%
RS/PC	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Overall	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	66.67%	25.00%	25.00%	79.63%
Network Overall	100.00%			100.00%			38.89%			

Table 8.7: Completeness of Trial Case Reults for Minimizing Cost.

As can be seen, the incomplete simulations only occur in Network C, which is the largest network with the most spans, nodes and rings. Furthermore, shorter OD pairs (ODS) have a better chance of getting completed. Provisioning a longer path is a more complex task than provisioning a shorter one. It is certain that the RS/PC-R can give completed simulation results (100%) even when the network is large (Network C) and the number of path constructions is enormous (see section 5.3). The following table is similar to Table 8.7 but it is for minimizing unavailability.

	Network A			Network B			Network C			Overall
	ODS	ODM	ODL	ODS	ODM	ODL	ODS	ODM	ODL	
Ex	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	0.00%	0.00%	0.00%	66.67%
RS	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	0.00%	0.00%	77.78%
PC	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	0.00%	0.00%	0.00%	66.67%
RS/PC	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Overall	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	50.00%	25.00%	25.00%	77.78%
Network Overall	100.00%			100.00%			33.33%			

Table 8.8: Simulation Completeness Table for Minimizing Unavailability.

The following graphs look at the run-time required by the test cases:

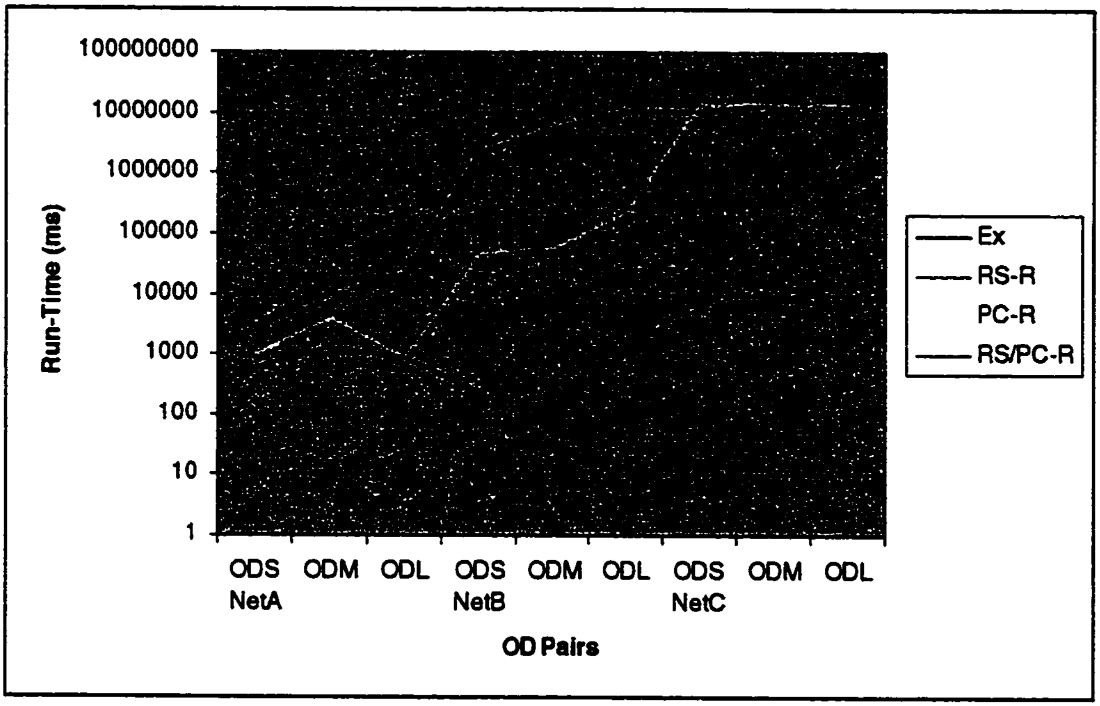


Figure 8.11: Run-time #1 (Minimizing Cost)

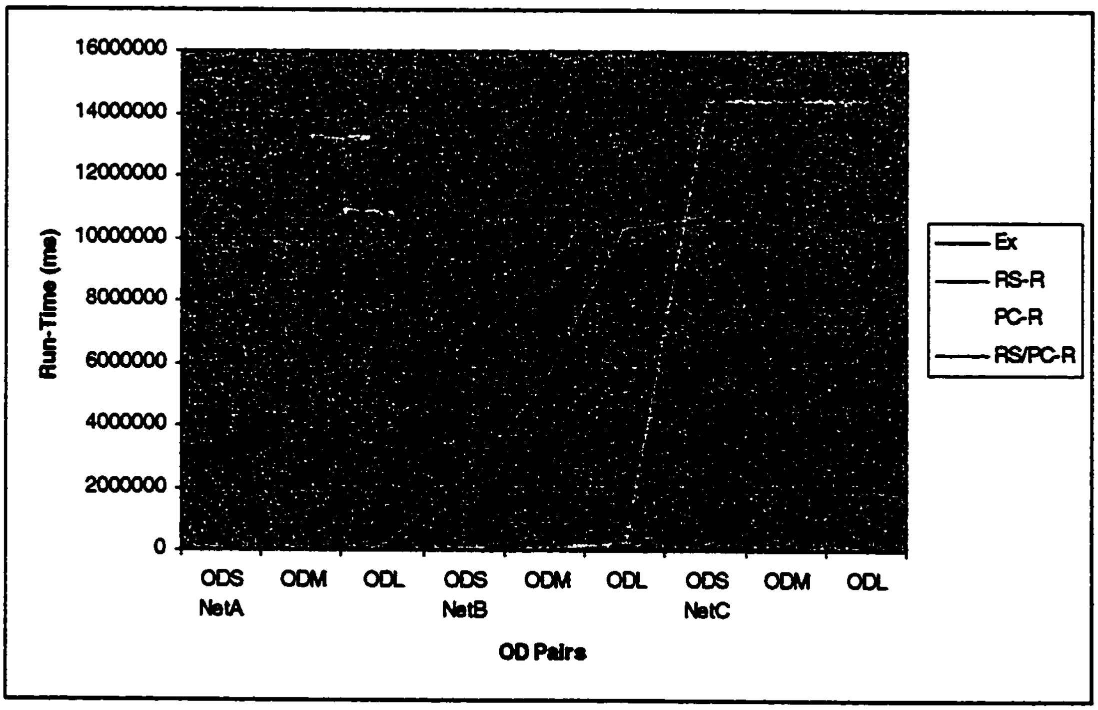


Figure 8.12: Run-time #2 (Minimizing Cost)

The two graphs capture the same data except that Figure 8.11 is in logarithmic scale and Figure 8.12 is in linear scale. Although the x-axis contains a set of non-sequential data, i.e. each

OD pair is independent (ODS, ODM and ODL from three networks), it was decided to order them this way and present them in a line graph instead of a bar graph. By putting them in an increasing order of the problem size (e.g. ODS before ODL; NetA before NetB), the data sets exhibit more or less exponential behaviour of run-time in response to network size and minimizing ring sequence length.

In Figure 8.11, it can be seen that all four methods exhibit exponential effects over certain regions. Note that the exhaustive method, RS-R and PC-R, does not plateau at the end of the line graph. As the simulation cannot be completed within four hours, all points plateau at 144,000 milliseconds (4 hours). However, in Figure 8.12 it can be seen that the RS/PC-R is the only search method that can be completed within the time limit in all cases, and it increases the run-time observably only when the problem size is very large (i.e. ODL in network C). The following two figures show the run-time of minimizing unavailability, and exhibit similar behavior.

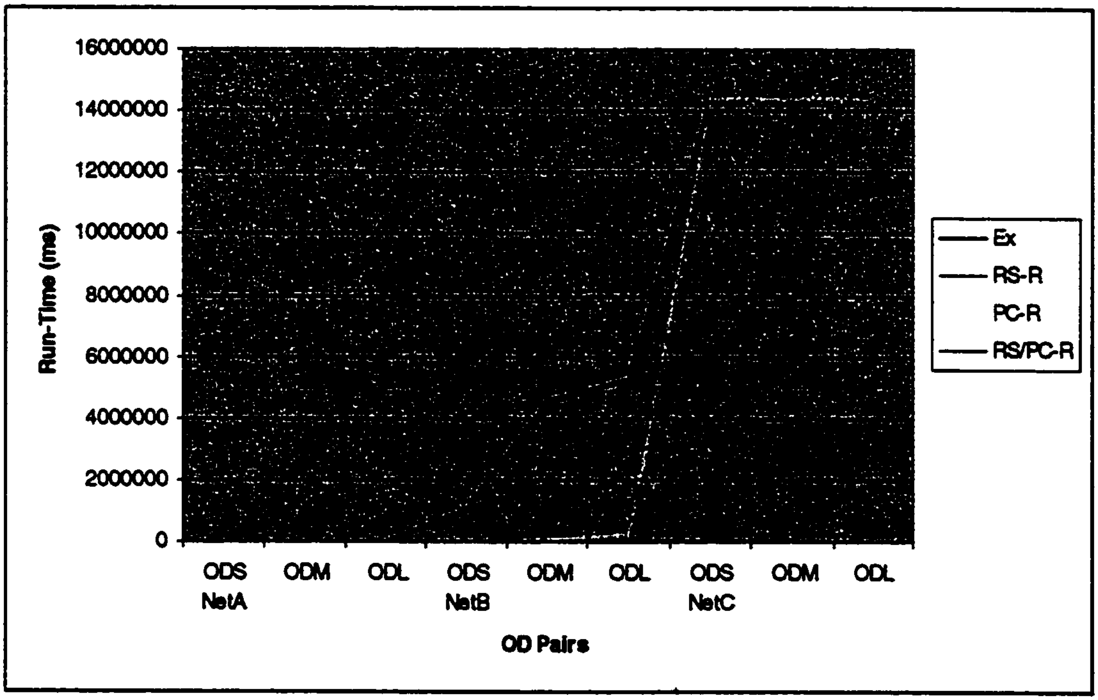


Figure 8.13: Run-time #3 (Minimizing Unavailability)

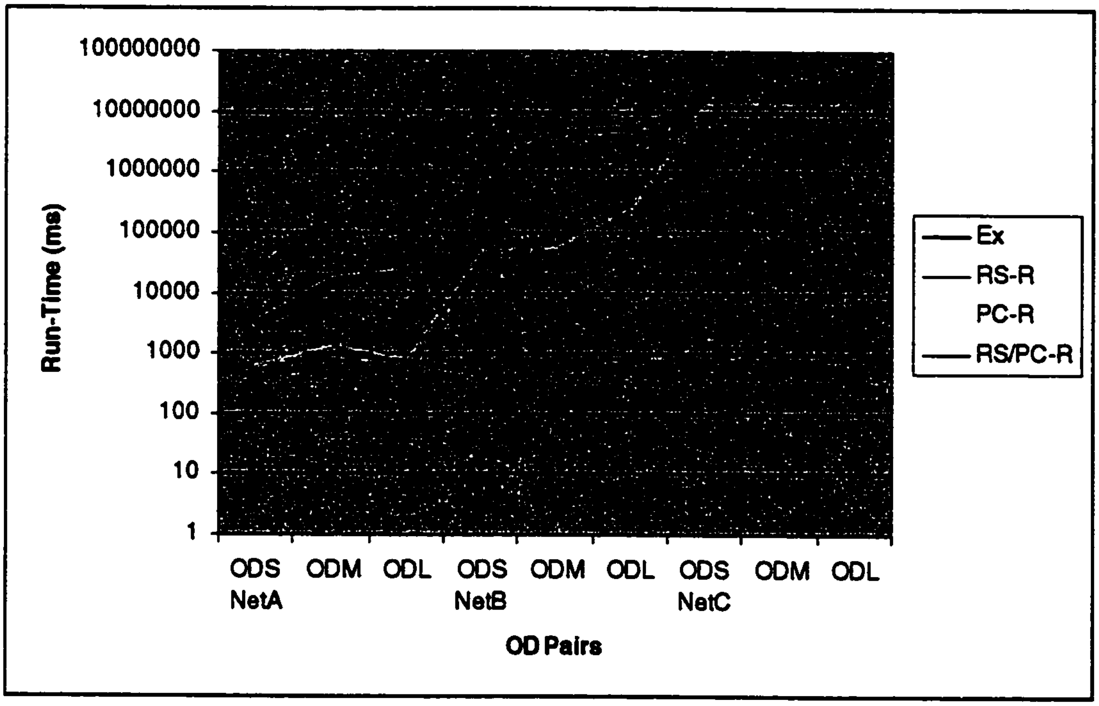


Figure 8.14: Run-time #4 (Minimizing Unavailability)

8.6. Constrained Single Objective and Multi-objective Test Cases

In approaching multi-objective cases, i.e. minimizing both cost and unavailability, the study implemented a feature that can set one objective to be minimized and the other as a maximum constraint, that is, either minimizing cost with a maximum unavailability constraint, or minimizing unavailability with a maximum cost constraint. Although the best way is to directly generate a Pareto solution set, the study's approach is indeed more practical for the use of network provisioning planners. Moreover, the Pareto solution set can actually be generated using this approach, but with extra procedures. For example, first a single objective is used to minimize unavailability without cost constraint. With the corresponding cost value of the resulting path, this cost can then be lowered by an incremental value, and set as a constraint. Then minimizing unavailability can be performed with the cost constraint just set. By successively performing this procedure, an partially sampled Pareto solution set will be generated. This procedure can also be performed using minimizing cost as the single objective.

The reason why the Pareto solution set is approximated is because if there is more than one solution within an incremental value of the constraint, it will not be generated because the incremental values are too far apart. Furthermore, if not using the exhaustive method, the solutions will all be approximate because they are not guaranteed optimal.

Note that if the incremental values are divided into 20 steps, it would require 20 runs of the simulation. However, with modification of the software prototype, the Pareto solutions set can be generated in a single run by recording all generated solutions, then determining if a particular solution is on the Pareto set. Unfortunately, this feature is not complete due to time constraints.

One way to determine if a particular solution is in the Pareto set, we can check if there exists another solution that is further minimized on one objective (or more) without any degradation on the other objective(s). We can do this by comparing the solution of interest with the solutions in the partially calculated Pareto set (the partially calculated Pareto set initially starts as an empty set, and adds the first solution point to the set, then continuously build using the following procedure). If there exists such solution in the Pareto set, then the solution of interest is not in the Pareto set. However, if there is no such solution, then the solution of interest should be put into the partially calculated Pareto set. All solutions in the set should be re-evaluated because some of them does not belongs to the Pareto set anymore because of the newly introduced solution.

The following table records the approximated Pareto solution set generated using the method described above. The ODS-1 O-D pair in Network C was used as the test case. First, the unavailability was minimized (MinU) and the corresponding cost (CminU) is determined. Then the cost (MinC) was minimized and the corresponding unavailability (UminC) is also determined. The cost difference between MinC and CminU was calculated, and divided by 20 to yield the incremental value. Then Cost Constraint column in Table 8.9 was set up. OPTICA was executed to obtain results for minimizing unavailability using the Cost Constraint column in Table 8.9. The results are recorded in the Actual Unavailability and Actual Cost columns.

	Cost Constraint	Actual Unavailability	Actual Cost
0%	333.1667	5.02E-06	299.09999
5%	324.285	5.02E-06	299.09999
10%	315.4033	5.02E-06	299.09999
15%	306.5217	5.02E-06	299.09999
20%	297.64	5.04E-06	248.73333
25%	288.7583	5.04E-06	248.73333
30%	279.8767	5.04E-06	248.73333
35%	270.995	5.04E-06	248.73333
40%	262.1133	5.04E-06	248.73333
45%	253.2317	5.04E-06	248.73333
50%	244.35	6.18E-06	229.29999
55%	235.4683	6.18E-06	229.29999
60%	226.5867	8.76E-06	205.89999
65%	217.705	8.76E-06	205.89999
70%	208.8233	8.76E-06	205.89999
75%	199.9417	8.80E-06	155.53333
80%	191.06	8.80E-06	155.53333
85%	182.1783	8.80E-06	155.53333
90%	173.2967	8.80E-06	155.53333
95%	164.415	8.80E-06	155.53333
100%	155.5333	8.80E-06	155.53333

Table 8.9: Generating Pareto Curve for ODS1 in Network C

The following figure plots the loci of a actual unavailability and actual cost coordinates obtained this way. It is the approximated Pareto solutions set curve.

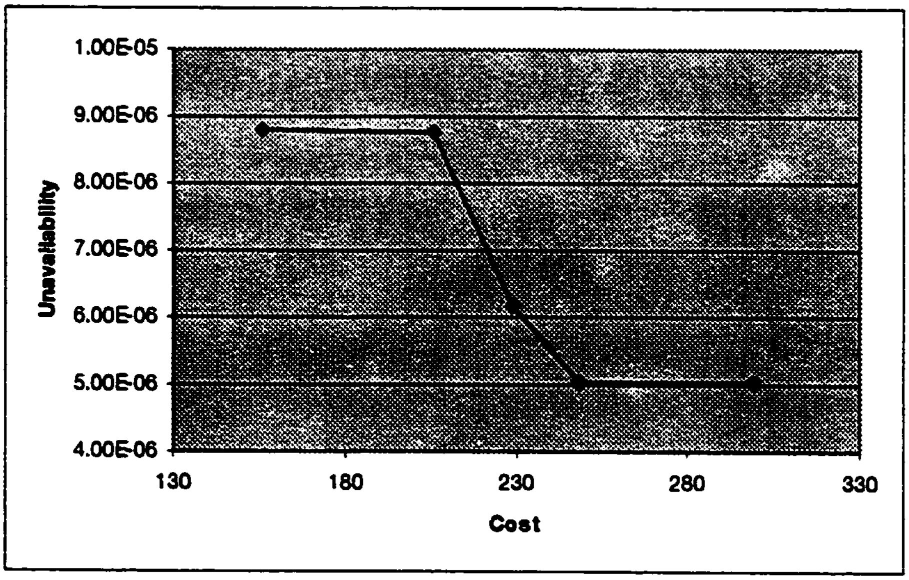


Figure 8.15: Pareto Solutions Set for ODS1 on Network C

Similarly, Figures 8.16 and 8.17 show the Pareto solutions set for ODM1 and ODL1 respectively on Network C. It can be observed that ODL has more unique data points than ODM (and ODM has more than ODS). This is because a longer path will traverse more equipment, hence there are more detailed choices when provisioning the path, hence a finer quantization of actual cost and unavailability values.

For the network planner doing 'what if' scenario analysis regarding path provisioning, there are a lot of non-obvious choices that may be missed if not using the function of optimizing single objective with constraint, or directly reading the Pareto curve. For example, in attempting to minimize the unavailability for ODS1 in Network C, the resulting path using single objective minimization is 5.02e-6 in unavailability, with 299.0999 as cost. However, if 0.3984% higher in unavailability (5.04e-6) can be tolerated, then 16.84% (248.7333) of the cost can be saved as shown in Table 8.9 and figure 8.15, by enabling a transition or jump in the path design. In this case the slight relaxation in unavailability that contributes to the major cost reduction is due to the use of different ring sequence. On the other hand, cost or unavailability does not change dramatically for using different gateway selection/configuration.

Thus, cost and unavailability of a path involve a trade-off, in which lowering the cost may raise the unavailability, and lowering the unavailability may raise the cost by forcing changes in the path construction details. The Pareto curve is very useful for the network planner in analyzing the path provisioning problem because the curve presents all the best available options, and the planner can make a more informed decision based on what requirement and preferences the service provider or the customer has.

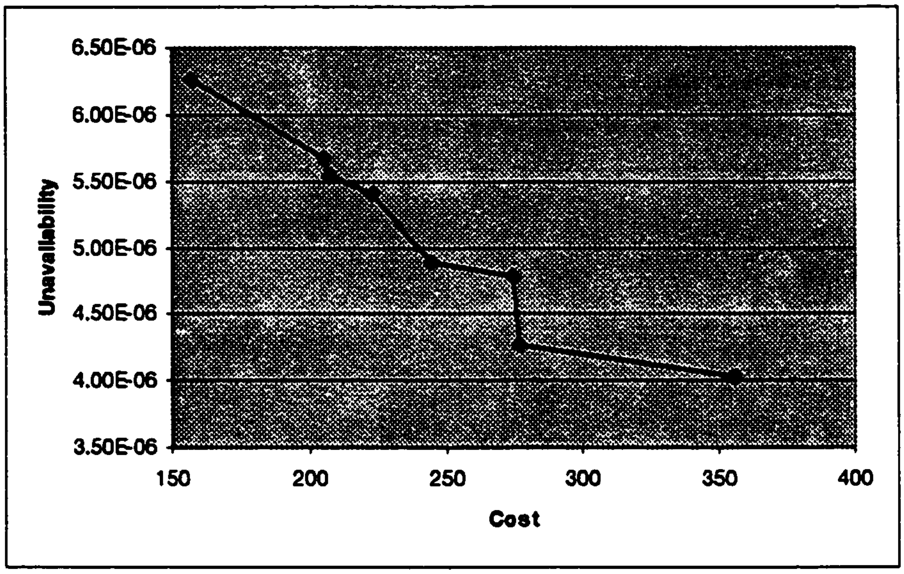


Figure 8.16: Pareto Solutions Set for ODM1 on Network C

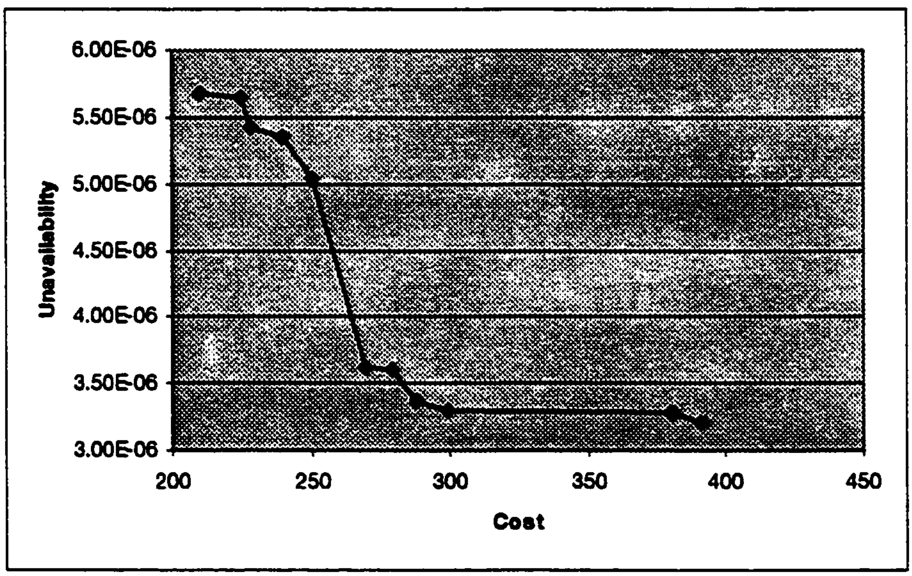


Figure 8.17: Pareto Solutions Set for ODL1 on Network C

9. Conclusions

9.1. Summary of Results and Contributions

The following is a list of major findings / contributions of this research effort:

- Demonstrated experimentally using test cases that the prior prediction that use of MN/DF combined paths can result in better cost and/or availability.
- Demonstrated and explained why the MN-only path does not necessarily always hold the absolute lowest unavailability (compared to DF-only and MN/DF combined).
- Characterized and analyzed the path provisioning problem in terms of combinatoric complexity.
- Designed, implanted and tested effective heuristic path search algorithms to generate optimal or near-optimal solutions for single objective, and single objective with constraint path provisioning optimization problems.
- Demonstrated that cost and unavailability are in a trade-off balance relationship, in which the bi-objective optimization of both parameters can be illustrated using an approximated Pareto solution curve.
- Developed the automatic path provisioning tool (OPTICA) with the proposed heuristic methods.

9.2. Topics for Further Research

9.2.1 Generating Pareto Solution Sets in OPTICA

Currently, OPTICA can only support the optimization of single objective and constrained single objective problems. As demonstrated in section 8.6, this function can be used to generate the approximate Pareto solution set. However, it has to be done with a certain number of runs of the simulation, depending on the incremental step size of one of the objectives.

Moreover, the Pareto solution set is already being traversed during each exhaustive method run of OPTICA (using other heuristic methods is viable, but will result in approximated data points). Note that for the heuristics proposed in Chapter 6, although the solution space has been substantially pruned down, quite a number of path constructions must still be evaluated in order to find the most desirable result. Since the users specify their optimization objective and constraints, the path search engine will only look for the optimal path with the most desirable parameters. When designing the system in the beginning, an attempt was made to specifically solve this type of problem (single objective with constraint); the usefulness and ability of the Pareto solution set were not realized until after the software prototype is developed and the simulations had been completed. Therefore this function was not implemented.

However, with some modification, this function could easily be integrated into the OPTICA system. With this functionality in place, the system could be even more effective than before because one run of the simulation will result in a Pareto solution set instead of a single answer.

9.2.2 Manual Path Provisioning in OPTICA

OPTICA currently does not support manual path provisioning. Manual path provisioning allows the planner to manually construct an end-to-end path using point-and-click graphical user interface. This could be a very useful feature for experienced manual planners to compare their path designs against the automatically generated designs. It is also good for re-evaluating existing paths provisioned on the current network.

9.2.3 SLA Guarantee Theory

SLA stands for service level agreement. It is an agreement between the customer and service provider which specifies the quality of service here, path unavailability guaranteed over a certain period of time. It typically includes terms and clauses stating that if the service outage time is over a certain limit, the service provider will forfeit payment, or will incur some other kind of penalty.

With the formulations in section 3.3, the theoretical path design end-to-end availability can be calculated. However, even if we assume that this is the true availability, this is still only the limiting average over an infinite observation time. Hence the problem arises: how can a value be guaranteed over a finite period of time. If the service provider wants to have a suitably low probability of breaking the SLA promise, he should advisably only set the guaranteed availability at a lower value than the calculated design availability. This is because the true availability is just a long-term average statistical value.

To demonstrate the effect of a finite time interval, assume that the service of a system of availability 99% is leased to customers (figure 9.1). Assume also that the service provider guarantees 99% service availability on all SLAs. The "Actual System" on figure 9.1 shows feasible patterns of system outages during a long period of time, corresponding to this long-term availability.

Now say that customer #1 leased the service for the entire operational period. It is obvious that the service availability just meets the SLA agreement at 99%. However, now consider a customer #2 who leased during the first half of the operational period. The service availability is 99.6%, which is actually better than the guaranteed availability. However, for customer #3, the service availability is only 98.4% and the SLA fails to meet the guarantee. In other words the finite duration availability experienced will be statistically distributed around the true long term availability.

The suggestion for further work would be to have OPTICA contain the relevant theory and import equipment MTTR and MTBF data to allow a user to generate SLA specifications with specified confidence intervals.

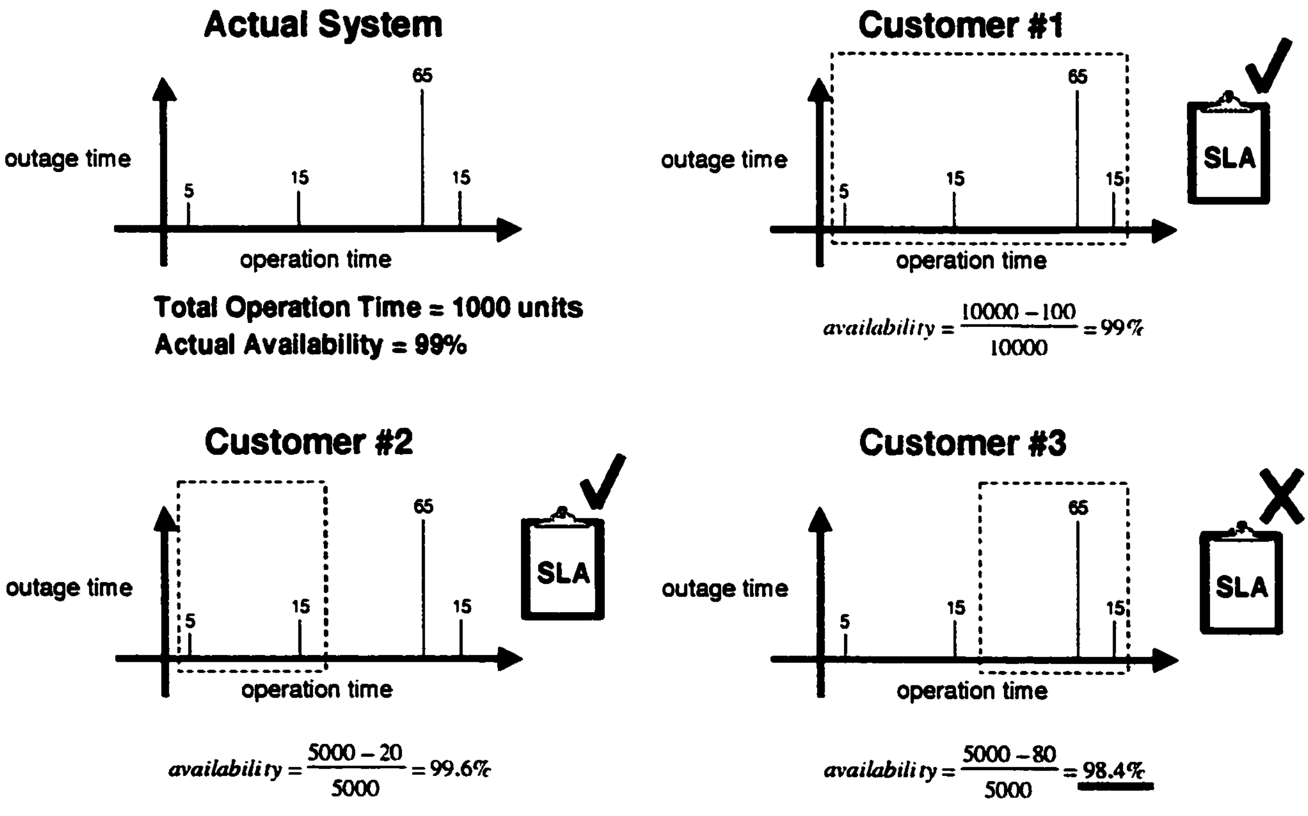


Figure 9.1: SLA of Various Time Intervals on a System

With this simple example, it can be seen that if the actual system availability of 99% is used to guarantee the SLA, there is a 50% chance of failing to meet the guarantee. With the SLA guarantee study, the provider should be able to tell, for example, given the certain actual system availability, what the adjusted availability is so that he will be able to meet the guarantee 95% of the time (figure 9.2). The SLA guarantee theory is the study of modeling the statistical behavior of availability over finite time intervals.

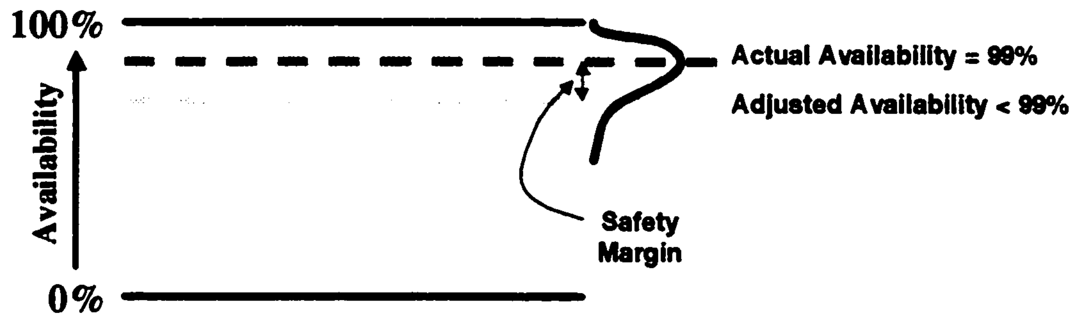


Figure 9.2: Adjusting Actual System Availability for SLA

9.2.4 Continue-Signal on Protection for Matched Nodes

The last two areas for further work are enhancements to the options available for path provisioning in a ring-based environment.

For the matched-nodes configuration, the continue-signal uses working bandwidth just like regular signals. A suggestion was made by Nortel (a TRLabs sponsor) that a study consider an alternative by using protection bandwidth for the continue-signal instead of working bandwidth. This is called "continue-signal on protection for matched nodes". Obviously this type of MN has a lower availability than regular MN, but it may be more economical to deploy. However, the regular MN inter-ring availability calculation cannot be used for this type of MN because new dual-failure scenarios are introduced (e.g. single equipment intra-ring failure plus secondary gateway failure). This study would involve the analysis of the new availability model for this type of matched nodes.

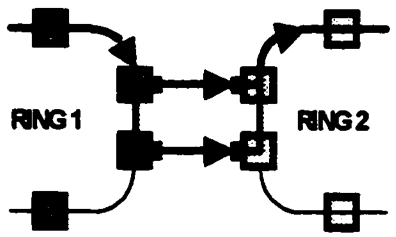


Figure 9.3: Continue-Signal using Protection Bandwidth on Matched Nodes

9.2.5 Extended Continue-Signal for Matched Nodes

For two rings with only single connectivity (i.e. only one node is common between the two rings), it is impossible to apply matched nodes on the inter-ring connection. However, with the use of an "extension signal", MN can be applied to such circumstances with single connectivity. This is called "extended matched nodes". Figure 9.4 shows two ways to apply extended continue-signal for MN.

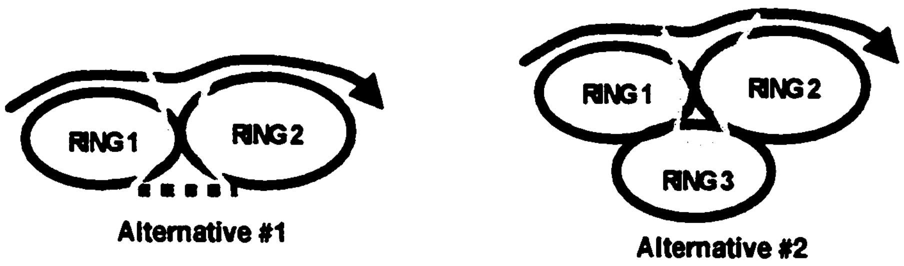


Figure 9.4: Extended Continue-Signal for Matched Nodes

The first alternative uses an external point-to-point link to connect the two secondary gateways that are physically apart. For the second alternative, the continue-signal is routed to the secondary gateway of the 1st ring, then goes through some other ring(s) before reaching the secondary gateway on the 2nd ring. In the alternative #2 example in figure 9.4, the continue-signal goes through Ring 3 before reaching Ring 2. The focus of this study possible would be to:

1. Derive new availability models for alternatives #1 and #2.
2. Design algorithms that will search for the feasibility of such extended MN opportunities in alternative #2.

9.2.6 Dual Feeding Signals on Extra Traffic

Dual feeding on extra traffic is a new proposed way to provision an end-to-end path through a ring network. As the protection capacity on a ring can be used to provision extra traffic, which will be dropped upon protection switch, we realized that the availability of this kind of path is low. However, we can provision another end-to-end path that serves the same origin and destination, but are physically diversely routed for protecting the original extra traffic path. In this case, we can eliminate most of the single element failure scenario of an extra traffic path, thus results in better availability. Moreover, it will require to develop a new availability model for this type of path to further analyze and understand its properties.

Dual feeding signals on extra traffic is an idea combining the previous two methods: the use of protection bandwidth and extended signal. The two signals of DF use only the protection bandwidth from end-to-end, and the flow of each individual signal can be routed independently, i.e., the two signals can uses two different ring sequences (figure 9.5). Any traffic routed on the protection bandwidth is called extra traffic because when there is an intra-ring failure, all the extra traffic will be dropped as a result of automatic protection switching. The focus of this study would be to:

1. Derive new availability models for this type of path.
2. Design algorithms that will search for the feasibility of such dual end-to-end extra traffic paths.

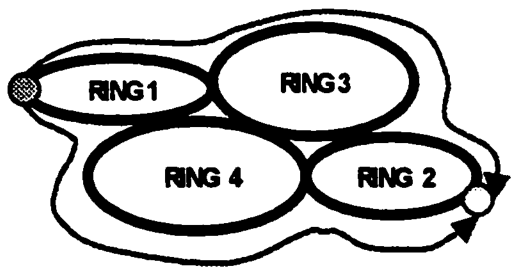


Figure 9.5: Dual Feeding Signals on Extra Traffic

10. References

1. W.E. Falconer, "Service Assurance in Modern Telecommunication Networks", IEEE Communications Magazine, June 1990, pp. 32-39.

2. J.C. MacDonald, "Public Network Integrity – Avóiding a Crisis of Trust", IEEE Journal of Selected Areas in Communications, vol.12, no.1, January 1994, pp.5-12.

3. W.D.Grover, "Availability, Reliability and Restorability", University of Alberta, EE681 course notes, pp.1-3.

4. W.D.Grover, "Alternative Routing Strategies for High Availability Paths in SONET Ring Networks", TRLabs, January 1998.

5. W.D.Grover, "Resource Management for Fault Tolerant Path Structures in SONET Ring Networks", JNSM 1997, June 1st 1998.

6. Nortel (Northern Telecom), SONET 101, October 1996, http://www.nortel.com/broadband.

7. Nortel (Northern Telecom), Introduction to SONET Networking, October 1996, http://www.nortel.com/broadband.

8. M. I. Henig, "The shortest path problem with two objective functions", European Journal of Operational Research 25, 1985, p. 281-291.

9. J. M. Jaffe, "Algorithms for Finding Paths with Multiple Constraints", Networks, Vol. 14, 1984, p.95-116.

10. P. Hansen, "Bicriterion Path Problems", In Multiple criteria decision making: Theory and Application., ed. G. Fandel and T. Gal. Lecture Notes in Economics and Mathematical Systems, 177, Springer-Verlag, p.109-127.

11. M. To, P. Neusy, "Unavailability Analysis of Long-Haul Networks", IEEE Journal on Selected Areas in Communications, Vol.12, No.1, January 1994.

12. B. Pelegrin, P. Fernandez, "On the Sum-Max Bicriterion Path Problem", Computers & Ops. Res. Vol. 25, No. 12, p.1043-1054, 1998.

13. E. Q. Martins, "On a Multicriteria Shortest Path Problem", European Journal of Operational Research 16, 1984, p.236-245.

14. D. J. White, "The Set of Efficient Solutions for Multiple Objective Shortest Path Problems", Computers & Ops. Res. Vol. 9, No. 2, p.101-107, 1982.

15. J. Current, C. ReVelle, J. Cohon, "An Interactive Approach to Identify the Best Compromise Solution for Two Objective Shortest Path Problems", Computers Opns Res. Vol. 17, No. 2, p.187-198, 1990.

16. Y. P. Aneja, V. Aggarwal, K. P. K. Nair, "Shortest Chain Subject to Side Constraints", Networks, Vol. 13, 1983, p.295-302.

17. G. Y. Handler, I. Zang, "A Dual Algorithm for the Constrained Shortest Path Problem", Networks, Vol. 10, p.293-310, 1980.

18. J. E. Beasley, N. Christofides, "An Algorithm for the Resource Constrained Shortest Path Problem", Networks, Vol. 19, 1989, p.379-394.

19. J. Y. Yen, "Finding the K Shortest Loopless Paths in a Network", Management Science, Vol. 17, No. 11, July, 1971.

20. Sara Baase, "Computer Algorithms: Introduction to Design and Analysis", Addison Wesley, Second Edition, 1988.

21. W.D. Grover, J.B. Slevinsky, M.H. MacGregor, "Optimized Design of ring-based Survivable Networks", Can. J. Elect. & Comp. Eng., Vol. 20, No. 3, 1995.

22. M. Labbe, G. Laporte, P. Soriano, "Covering a Graph with Cycles", Computers Ops Res. Vol. 25, No.6, pp.499-504, 1998.

23. S. Khanna, "A Polynomial Time Approximation Scheme for the SONET Ring Loading Problem", Bell Labs Technical Journal, Spring 1997.

24. W.D. Grover, T.D. Bilodeau, B.D. Venables, "Near Optimal Spare Capacity Planning in a Mesh Restorable Network", Globecom 1991.

25. M. Herzberg, S. Bye, "An Optimal Spare-Capacity Assignment Model for Survivable Networks with Hop Limits", Globecom 1994.

26. B.D. Venables, W.D. Grover, M.H. MacGregor, "Two Strategies for Spare Capacity Placement in Mesh Restorable Network", ICC 1993.

27. A. Antonopoulos, "Planning Telecommunication Transport Networks: Metrication, Analysis and Design", Ph.D. thesis for Department of EEE, University College London, December 1997.

28. X.P. Yan, P.D. Ye, "Survivability Study of SDH Self-Healing Ring", Beijing University of Posts and Telecommunications, Beijing 100088.

29. L. Nederlof, K. Struyve, C. O'Shea, H. Misser, Y. Du, B. Tamayo, "End-to-End Survivable Broadband Networks", IEEE Communications Magzine, September 1995.

30. P. Conn, T. Flanagan, R. Quintal, "End-to-end survivability", Telecommunication Access Networks Technology and Service Trends, Elsevier Science Publishers B.V. (North-Holland), 1991.

31. J. Baudron, A. Khadr, F. Kocsis, "Availability and Survivability of SDH Networks", Alcatel CIT Villarceaux, France, Alcatel SEL Stuttgart, Germany.

Appendix A: Test Case Parameters

A.1. Standard Network Interface Format (SNIF)

The following files are the SNIF files of all the test cases networks. All the nodes and spans information is recorded in these text files.

```
Date:
File Name: NetworkA.snif
Network: NetworkA
Program:

Node  Xcoord               Ycoord
1     39.34359848686663    60.65107357877351
2     19.779423517502256   69.78102189781022
3     9.12781714529276     60.65107357877351
4     9.345196867174586    40.0
5     20.21418296126591    54.56444136608237
6     59.77729234375832    50.65160637220949
7     39.778357930630285   46.95615110021845
8     50.429964302839785   30.435292237199633
9     21.30108157067504    19.566306143108314
10    29.561511002184446   10.4363578240716
11    39.56097820874846    20.218445308753793
333
Span  NodeA NodeB Distance  Working    Spare
1     1     2     7.0       0     0
2     1     3     9.0       0     0
3     1     4     9.0       0     0
4     1     5     6.0       0     0
5     1     6     8.0       0     0
6     1     8     6.0       0     0
7     2     3     5.0       0     0
8     3     5     4.0       0     0
9     4     5     6.0       0     0
10    4     8     9.0       0     0
11    4     9     6.0       0     0
12    5     6     11.0      0     0
13    5     7     6.0       0     0
14    5     8     9.0       0     0
15    5     9     8.0       0     0
16    5     11    10.0      0     0
17    6     8     4.0       0     0
18    7     8     5.0       0     0
19    8     9     6.0       0     0
20    8     11    5.0       0     0
21    9     10    3.0       0     0
22    9     11    4.0       0     0
23    10    11    4.0       0     0
```

Date:
File Name: NetworkB.snif
Network: NetworkB
Program:

Node	Xcoord	Ycoord
1	28173.10401459854	58743.46715328467
2	37042.26459854015	59630.38321167883
3	40412.54562043796	52002.90510948905
4	58683.01642335766	51293.37226277372
5	46088.80839416058	49696.92335766424
6	36864.88138686131	45439.72627737226
7	26399.271897810217	45439.72627737226
8	31188.618613138686	31958.602189781024
9	39170.86313868613	39940.84671532847
10	52474.60401459854	41892.062043795624
11	60102.08211678832	44907.57664233577
12	63827.129562043796	37989.631386861314
13	50700.77189781021	35683.6496350365
14	62408.063868613135	28233.55474452555
15	53538.90328467153	31249.069343065694
16	48217.406934306564	29120.470802919706
17	39703.01277372263	26459.722627737225
18	31720.76824817518	18477.47810218978
19	39170.86313868613	16703.64598540146
20	47862.640510948906	19009.627737226278
21	52119.837591240874	19719.160583941608
22	57086.56751824817	23266.824817518253
23	57441.33394160584	17590.562043795624
24	61875.91423357664	24685.890510948906
25	65955.72810218978	23621.59124087591
26	65423.57846715328	17413.17883211679
27	59747.31569343065	14929.813868613142
28	47330.49087591241	11027.383211678833
29	47507.87408759124	15284.580291970808
30	33671.983576642335	12269.06569343066

Span	NodeA	NodeB	Distance	Working	Spare
0	1	2	30.0	0	0
1	1	3	35.0	0	0
2	1	7	45.0	0	0
3	2	4	60.0	0	0
4	2	3	30.0	0	0
5	4	11	35.0	0	0
6	3	5	30.0	0	0
7	3	4	50.0	0	0
8	7	6	35.0	0	0
9	7	8	60.0	0	0
10	7	3	50.0	0	0
11	6	5	45.0	0	0
12	6	9	30.0	0	0
13	5	11	40.0	0	0
14	10	5	35.0	0	0
15	11	10	40.0	0	0
16	11	12	35.0	0	0
17	12	10	30.0	0	0
18	12	13	35.0	0	0

19	12	14	50.0	0	0
20	10	8	80.0	0	0
21	10	13	35.0	0	0
22	9	13	45.0	0	0
23	13	15	35.0	0	0
24	9	15	50.0	0	0
25	9	17	55.0	0	0
26	8	18	50.0	0	0
27	8	17	55.0	0	0
28	17	18	45.0	0	0
29	17	16	30.0	0	0
30	17	19	30.0	0	0
31	16	15	30.0	0	0
32	16	21	30.0	0	0
33	14	24	20.0	0	0
34	14	22	30.0	0	0
35	14	15	30.0	0	0
36	24	25	20.0	0	0
37	19	20	35.0	0	0
38	19	28	35.0	0	0
39	19	30	30.0	0	0
40	19	18	30.0	0	0
41	30	18	35.0	0	0
42	30	29	45.0	0	0
43	27	28	55.0	0	0
44	27	26	45.0	0	0
45	28	23	50.0	0	0
46	22	25	40.0	0	0
47	26	23	45.0	0	0
48	29	20	20.0	0	0
49	20	21	25.0	0	0
50	9	18	75.0	0	0
51	9	20	80.0	0	0
52	9	10	40.0	0	0
53	9	8	35.0	0	0
54	23	29	50.0	0	0
55	21	22	25.0	0	0
56	25	26	30.0	0	0
57	22	23	30.0	0	0
58	21	23	35.0	0	0

Date:
File Name: NetworkC.snif
Network: NetworkC
Program:

Node	Xcoord	Ycoord	
1	15.0	112.0	
2	34.0	109.0	
3	50.0	105.0	
4	62.0	99.0	
5	81.0	101.0	
6	101.0	101.0	
7	114.0	94.0	
8	120.0	94.0	
9	124.0	96.0	
10	130.0	101.0	
11	130.0	95.0	
12	133.0	88.0	
13	135.0	81.0	
14	138.0	80.0	
15	143.0	83.0	
16	141.0	88.0	
17	146.0	89.0	
18	151.0	87.0	
19	150.0	84.0	
20	157.0	82.0	
21	166.0	92.0	
22	165.0	99.0	
23	172.0	105.0	
24	180.0	108.0	
25	184.0	106.0	
26	189.0	95.0	
27	193.0	100.0	
28	194.92656587473002		105.28617710583153
29	197.0	104.0	
30	195.0	97.0	
31	194.0	94.0	
32	193.06047516198703		92.22354211663067
33	189.0	92.0	
34	191.0	89.0	
35	187.0	89.0	
36	187.0	86.0	
37	188.7062634989201		84.75917926565874
38	187.0	80.0	
39	184.0	82.0	
40	183.0	79.0	
41	180.0	79.0	
42	178.0	76.0	
43	179.0	73.0	
44	180.0	70.0	
45	182.0	70.0	
46	183.0	66.0	
47	179.0	66.0	
48	176.0	60.0	
49	173.0	55.0	
50	168.0	50.0	
51	168.0	45.0	

52	170.0	40.0	
53	170.0	32.0	
54	176.0	25.0	
55	179.46900219101605		17.141273752251443
56	181.0	9.0	
57	175.0	9.0	
58	170.0	18.0	
59	170.0	23.0	
60	162.0	31.0	
61	168.0	31.0	
62	170.0	25.0	
63	148.6889848812095		29.605831533477314
64	146.0	32.0	
65	140.0	29.0	
66	137.0	29.0	
67	135.0	25.0	
68	130.0	21.0	
69	125.0	26.0	
70	132.0	29.0	
71	132.0	26.0	
72	120.0	26.0	
73	117.0	23.0	
74	112.0	22.0	
75	105.0	16.0	
76	103.0	9.0	
77	100.0	9.0	
78	98.0	20.0	
79	100.0	26.0	
80	100.0	36.0	
81	104.0	26.0	
82	82.0	31.0	
83	63.0	31.0	
84	62.0	36.0	
85	45.0	36.0	
86	40.0	40.0	
87	18.0	46.0	
88	30.0	55.0	
89	16.0	46.0	
90	18.0	42.0	
91	18.0	39.0	
92	16.0	38.0	
93	15.0	41.0	
94	16.0	43.0	
95	14.0	48.0	
96	16.0	50.0	
97	13.0	50.0	
98	10.0	50.0	
99	7.0	56.0	
100	6.0	64.0	
101	5.0	67.0	
102	9.0	68.0	
103	10.0	65.0	
104	12.0	61.0	
105	12.0	57.0	
106	9.0	72.0	
107	7.0	75.0	
108	7.0	80.0	

109	10.0	90.0
110	10.0	94.0
111	10.0	97.0
112	13.0	101.0
113	13.0	75.0
114	27.0	75.0
115	45.0	75.0
116	70.0	75.0
117	70.0	84.0
118	60.0	86.0
119	60.0	95.0
120	70.0	64.0
121	69.27645788336933	58.21922246220302
122	72.0	57.0
123	72.0	51.0
124	64.0	51.0
125	62.0	47.0
126	114.0	90.0
127	102.0	86.0
128	102.0	81.0
129	105.0	75.0
130	112.0	65.0
131	105.0	65.0
132	101.0	59.0
133	100.0	49.0
134	95.0	40.0
135	85.0	39.0
136	85.0	47.0
137	127.0	86.0
138	130.0	73.0
139	123.0	75.0
140	120.0	80.0
141	115.0	80.0
142	127.0	64.0
143	116.0	64.0
144	140.0	53.0
145	122.0	46.0
146	108.0	51.0
147	104.0	49.0
148	142.88336933045358	55.52375809935205
149	144.95680345572353	54.07235421166306
150	145.0	41.0
151	133.0	35.0
152	110.0	35.0
153	148.0	38.0
154	155.0	46.0
155	150.0	50.0
156	140.0	60.0
157	140.0	65.0
158	134.0	70.0
159	140.0	71.0
160	139.0	75.0
161	154.0	77.0
162	150.0	71.0
163	150.0	66.0
164	146.0	66.0
165	146.0	61.0

166	144.0	60.0
167	154.90928725701943	72.52591792656587
168	157.0	70.0
169	161.0	70.0
170	164.0	63.0
171	160.0	58.0
172	156.0	58.0
173	155.32397408207342	55.10907127429805
174	170.0	65.0
175	165.0	55.0
176	161.0	51.0
177	159.0	67.0
179	175.22894168466522	70.2451403887689
180	162.0	82.0
181	167.0	78.0
182	170.0	78.0
183	173.0	76.0
184	173.0	81.0
185	177.0	81.0
186	180.0	85.0
187	177.0	94.0
188	171.0	95.0

Span	NodeA	NodeB	Distance	Working	Spare
1	1	2	437.0	0	0
2	2	3	385.0	0	0
3	3	4	306.0	0	0
4	4	5	412.0	0	0
5	5	6	431.0	0	0
6	6	7	348.0	0	0
7	7	8	137.0	0	0
8	8	9	78.0	0	0
9	9	10	169.0	0	0
10	10	11	132.0	0	0
11	11	12	179.0	0	0
12	12	13	158.0	0	0
13	13	14	63.0	0	0
14	14	15	145.0	0	0
15	15	16	92.0	0	0
16	16	17	101.0	0	0
17	17	18	123.0	0	0
18	18	19	66.0	0	0
19	19	20	155.0	0	0
20	20	21	288.0	0	0
21	21	22	116.0	0	0
22	22	23	138.0	0	0
23	23	24	146.0	0	0
24	24	25	75.0	0	0
25	25	26	241.0	0	0
26	26	27	108.0	0	0
27	27	28	136.0	0	0
28	28	29	50.0	0	0
29	29	30	152.0	0	0
30	30	27	96.0	0	0
31	30	26	120.0	0	0
32	26	31	90.0	0	0
33	26	33	86.0	0	0

34	31	32	66.0	0	0
35	32	33	95.0	0	0
36	33	34	86.0	0	0
37	34	35	89.0	0	0
38	33	35	75.0	0	0
39	35	36	80.0	0	0
40	36	37	45.0	0	0
41	37	38	73.0	0	0
42	38	39	56.0	0	0
43	36	39	79.0	0	0
44	39	40	54.0	0	0
45	40	41	48.0	0	0
46	41	42	62.0	0	0
47	42	43	56.0	0	0
48	43	44	56.0	0	0
49	44	45	45.0	0	0
50	45	46	72.0	0	0
51	46	47	75.0	0	0
52	44	47	69.0	0	0
53	47	48	115.0	0	0
54	48	49	131.0	0	0
55	49	50	140.0	0	0
56	51	52	100.0	0	0
57	52	53	169.0	0	0
58	53	54	163.0	0	0
59	54	55	148.0	0	0
60	55	56	136.0	0	0
61	56	57	97.0	0	0
62	57	58	187.0	0	0
63	58	59	148.0	0	0
64	59	60	246.0	0	0
65	60	61	150.0	0	0
66	61	62	110.0	0	0
67	54	62	121.0	0	0
68	55	59	210.0	0	0
69	59	62	28.0	0	0
70	60	63	329.0	0	0
71	63	64	80.0	0	0
72	64	65	147.0	0	0
73	65	66	42.0	0	0
74	66	67	113.0	0	0
75	67	68	100.0	0	0
76	68	69	130.0	0	0
77	70	71	70.0	0	0
78	67	71	53.0	0	0
79	69	72	75.0	0	0
80	72	73	66.0	0	0
81	73	74	83.0	0	0
82	74	75	213.0	0	0
83	75	76	153.0	0	0
84	76	77	31.0	0	0
85	77	78	227.0	0	0
86	78	79	126.0	0	0
87	79	80	220.0	0	0
88	80	81	243.0	0	0
89	74	81	222.0	0	0
90	79	81	71.0	0	0

91	80	82	425.0	0	0
92	82	83	409.0	0	0
93	83	84	94.0	0	0
94	84	85	383.0	0	0
95	85	86	152.0	0	0
96	86	87	485.0	0	0
97	87	88	319.0	0	0
98	86	88	368.0	0	0
99	50	51	74.0	0	0
100	87	89	55.0	0	0
101	89	94	74.0	0	0
102	94	90	56.0	0	0
103	90	91	68.0	0	0
104	91	92	62.0	0	0
105	92	93	71.0	0	0
106	93	94	42.0	0	0
107	89	96	63.0	0	0
108	89	95	47.0	0	0
109	95	96	79.0	0	0
110	95	97	54.0	0	0
111	97	98	97.0	0	0
112	98	99	153.0	0	0
113	99	100	164.0	0	0
114	100	101	58.0	0	0
115	101	102	109.0	0	0
116	102	103	53.0	0	0
117	103	104	142.0	0	0
118	104	105	101.0	0	0
119	96	105	203.0	0	0
120	100	103	117.0	0	0
121	102	106	105.0	0	0
122	106	107	49.0	0	0
123	107	108	132.0	0	0
124	108	109	196.0	0	0
125	109	110	96.0	0	0
126	110	111	84.0	0	0
127	111	112	122.0	0	0
128	1	112	310.0	0	0
129	106	113	79.0	0	0
130	113	114	333.0	0	0
131	114	115	368.0	0	0
132	115	116	595.0	0	0
133	116	117	193.0	0	0
134	117	118	254.0	0	0
135	118	119	175.0	0	0
136	4	119	88.0	0	0
137	116	120	231.0	0	0
138	120	121	148.0	0	0
139	121	122	41.0	0	0
140	122	123	305.0	0	0
141	123	124	184.0	0	0
142	124	125	94.0	0	0
143	84	125	247.0	0	0
144	7	126	109.0	0	0
145	126	127	267.0	0	0
146	127	128	106.0	0	0
147	128	129	179.0	0	0

148	116	129	746.0	0	0
149	129	130	319.0	0	0
150	130	131	123.0	0	0
151	131	132	189.0	0	0
152	132	133	214.0	0	0
153	80	133	353.0	0	0
154	80	134	144.0	0	0
155	134	135	330.0	0	0
156	135	136	196.0	0	0
157	125	136	513.0	0	0
158	9	11	114.0	0	0
159	8	137	213.0	0	0
160	12	137	106.0	0	0
161	13	138	216.0	0	0
162	138	139	190.0	0	0
163	139	140	125.0	0	0
164	140	141	109.0	0	0
165	69	70	168.0	0	0
166	129	141	336.0	0	0
167	138	142	178.0	0	0
168	142	143	225.0	0	0
169	130	143	60.0	0	0
170	142	144	336.0	0	0
171	144	145	389.0	0	0
172	145	146	403.0	0	0
173	146	147	74.0	0	0
174	133	147	93.0	0	0
175	144	148	79.0	0	0
176	148	149	28.0	0	0
177	149	150	241.0	0	0
178	150	151	299.0	0	0
179	151	152	509.0	0	0
180	80	152	228.0	0	0
181	70	151	129.0	0	0
182	150	153	89.0	0	0
183	150	154	229.0	0	0
184	153	154	318.0	0	0
185	51	154	288.0	0	0
186	154	155	116.0	0	0
187	149	155	110.0	0	0
188	148	156	142.0	0	0
189	156	157	118.0	0	0
190	157	158	149.0	0	0
191	138	158	87.0	0	0
192	157	159	132.0	0	0
193	159	160	75.0	0	0
194	14	160	136.0	0	0
195	20	161	108.0	0	0
196	161	162	148.0	0	0
197	162	163	80.0	0	0
198	163	164	59.0	0	0
199	164	165	132.0	0	0
200	165	166	111.0	0	0
201	148	166	103.0	0	0
202	161	167	90.0	0	0
203	167	168	69.0	0	0
204	168	169	78.0	0	0

205	169	170	154.0	0	0
206	170	171	130.0	0	0
207	171	172	59.0	0	0
208	172	173	38.0	0	0
209	149	173	218.0	0	0
210	170	174	103.0	0	0
211	174	175	205.0	0	0
212	175	176	124.0	0	0
213	176	177	78.0	0	0
214	173	177	90.0	0	0
215	174	178	78.0	0	0
216	178	179	82.0	0	0
217	43	179	90.0	0	0
218	20	180	92.0	0	0
219	180	181	153.0	0	0
220	181	182	60.0	0	0
221	182	183	73.0	0	0
222	42	183	118.0	0	0
223	183	184	69.0	0	0
224	184	185	58.0	0	0
225	185	186	85.0	0	0
226	186	187	195.0	0	0
227	187	188	132.0	0	0
228	26	187	265.0	0	0
229	21	188	115.0	0	0
230	14	20	431.0	0	0
231	159	162	196.0	0	0
232	49	175	174.0	0	0
233	64	153	126.0	0	0
234	61	53	105.0	0	0
235	87	90	97.0	0	0
S0	30	31	1.0	0	0

A.2. Test Cases O-D Pairs

The following table contains all the O-D pairs used in our simulations.

	1	2	3	1	2	3	1	2	3
	2-6	3-4	6-7	2-4	2-9	1-10	2-10	2-7	2-11
	1-6	2-10	24-27	2-8	4-16	12-28	1-28	7-25	4-30
	1-82	46-168	52-145	2-146	17-63	24-48	3-52	24-85	46-104

Table 10.1: Simulation O-D Pairs

Appendix B: Matlab Files for Calculation Verification

B.1. Td_1 and Td_2 Formulation Files

The following is the Td_1 formulation in matlab format:

```
function ans=td1(s,w,sw1,w1,us1,un1)

ncost=1;
scost=0.033333;
t1=sw1*w1*us1^2;
t2=(((w+1)*sw1)+((s-w-1)*w1))*un1*us1;
t3=(w+1)*(s-w-1)*un1^2;

cost=w1*scost+(w+1)*ncost

ans=t1+t2+t3;
```

The following is the Td_2 formulation in matlab format:

```
function ans=td2(wa,wb,wa1,wb1,us1,un1)

ncost=1;
scost=0.033333;
t1=wa1*wb1*us1^2;
t2=(((wa+1)*wb1)+((wb+1)*wa1))*un1*us1;
t3=(wa+1)*(wb+1)*un1^2;

cost=(wa+wb)*ncost+(wa1+wb1)*scost

ans=t1+t2+t3;
```

B.2. Verification Calculation Files

The following six matlab files are the six verification cases mentioned in section 8.3. Function named *mn( )* is the basis file for all MN cases and function named *df( )* is the basis file for all DF cases.

```
function ans = mn()
us1=0.000001;
un1=0.00001;
una=0.00001;
k=6;
r1=td1(5,1,400,100,us1,un1)
r2=td1(4,1,300,100,us1,un1)
r3=td1(5,1,400,100,us1,un1)
r4=td1(8,1,700,100,us1,un1)
r5=td1(7,1,600,100,us1,un1)
r6=td1(5,1,400,100,us1,un1)
```

```
ee=2*(una^2+2*una*unl)
inter=2*(k-1)*(unl^2+2*una^2+4*una*unl)

ans=r1+r2+r3+r4+r5+r6+ee+inter;




function ans = df()
usl=0.000001;
unl=0.00001;
una=0.00001;
k=6;
r1=tdl(5,1,400,100,usl,unl)
r2=tdl(4,1,300,100,usl,unl)
r3=tdl(5,1,400,100,usl,unl)
r4=tdl(8,1,700,100,usl,unl)
r5=tdl(7,1,600,100,usl,unl)
r6=tdl(5,1,400,100,usl,unl)

ee=2*(una^2+2*una*unl)
inter=2*(k-1)*(unl^2+2*una^2+4*una*unl)

ans=r1+r2+r3+r4+r5+r6+ee+inter;
```