Availability Analysis of Meta-Mesh Restorable Transport Networks

by

Andres Castillo Lugo

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Engineering Management

Department of Mechanical Engineering
University of Alberta

ABSTRACT

The span-restorable meta-mesh approach was previously proposed as a novel method for improving the capacity efficiency of span restoration in a sparse network. The fundamental idea behind this method is to route lightpaths that fully transit chains of degree-2 nodes onto logical bypass spans that physically traverse the chain, but which are allowed to fail back to the anchor nodes of the chain. From the perspective of transiting lightpaths, the result is an increase in network connectivity. Previous work on the meta-mesh design considered only single-failure restorability. The work herein addresses the issue of meta-mesh dual-failure survivability by developing and evaluating two new ILP design models. The first model provides the minimum total cost to design a meta-mesh network capable of withstanding all possible dual span failures scenarios. The second model provides a maximization of the dual failure restorability by minimizing the number of non-restored working capacities with a given limit on total spare capacity investment. In addition, this work investigates an improvement of the prior meta-mesh design by allowing the existence of a logical bypass span in low priority chains in the network. Experiments are performed on six master test-case networks of various topologies and scales.

To my father…

ACKNOWLEDGEMENTS

This work would not have been possible without the support, continuous guidance, and infinite patience of my supervisor, Dr. John Doucette. Thanks John, you have set an example of excellence to me. I will forever be indebted to you.

I extend my gratefulness to my lab group member and friends, with whom I have shared many discussions. Among the many, they are Varun Sharma, Wenjing Wang, Mustafa Babadagli, Shinya Lin, Roberto Gallardo, and Mark Ruhl.

I am also indebted to Alyona Sharunova, for the great help, support, and encouragement she gave me during my studies.

I would especially like to thank my lovely family for the love, support, and constant encouragement I have gotten over the years. In particular, I would like to thank my brother Javier for taking this journey with me and go beyond borders to accomplish greater things together. To my mother Carmen, who has always stood by me like a pillar in times of need and to whom I owe my life to her constant love, encouragement, moral support and blessings.

Finally, I would like to dedicate this thesis to my father. It was you father who originally generated my interest in engineering. I still take your lessons with me, every day.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF NOMENCLATURE

$\boldsymbol{\delta_{i,j}^{p}}$ is a binary variable that is equal to 1 if the p^{th} eligible restoration route for span i uses span j, and is equal to 0 otherwise $\forall(i,j) \in \boldsymbol{S}^2, \forall p \in \boldsymbol{P}_i$.

$\boldsymbol{\zeta_{j}^{r,q}}$ is a binary variable that is equal to 1 if the q^{th} eligible working route for relation r uses span j, and is equal to 0 otherwise.

$\boldsymbol{B}$ is the total amount of spare capacity available as an investment in the network design.

$\boldsymbol{C_j}$ is the cost of each unit of capacity on span $j \in \boldsymbol{S}$.

D is the set of demand quantities for each service path relation in a network. Usually, it is indexed by r.

$\boldsymbol{d^r}$ is the amount of demands units for demand relation r.

$\boldsymbol{g^{r,q}}$ is the number of working flow allocated to the q^{th} eligible working route used for demand pair r.

$\boldsymbol{f_i^p}$ is the amount of flow routed on restoration route p for restoration of span i under a single-span failure scenario.

$\boldsymbol{f_{i,j}^p}$ is the restoration flow assigned to the pth eligible restoration route for span i when a span j has failed simultaneously.

$\boldsymbol{k_i} \in \mathbf{S_b}$ is the corresponding logical bypass span with its associated span in the network.

$\boldsymbol{N(i,j)}$ is the number of non-restored working capacities under a dual-failure scenario.

$\mathbf{P}_i$ is the set of all eligible restoration or backup routes for span $i \in \boldsymbol{S}$. Usually, it is indexed by p.

$\mathbf{Q}^r$ is the set of all eligible routes available for carrying demands for each relation r. Usually, it is indexed by q.

$\boldsymbol{R_2(i,j)}$ dual span failure restorability.

$\boldsymbol{R_2}$ weighted average of dual failure restorability on a network.

$\mathbf{S}$ is the set of all spans in a network.

$\mathbf{S}_\mathrm{b}$ set of all logical bypass spans added to the network.

$\mathbf{S}_\mathrm{c}$ set of all spans that are part of any chain in the network.

$\mathbf{S}_\mathrm{c}$ x $\mathbf{S}_\mathrm{c}$ set of all dual-failures scenarios where only chain spans are involved. In this case, the model is also extended to convert this dual-failure situation into a *logical* quadruple-failure scenario (their respective logical bypass spans).

$\mathbf{S}_\mathrm{d}$ set of spans whose end-nodes are both of degree-3 or higher. So-called *direct spans*.

$\mathbf{S}_\mathrm{d}$ x $\mathbf{S}_\mathrm{d}$ set of all dual-failure scenarios where only direct spans are involved.

$\mathbf{S}_\mathrm{d}$ x $\mathbf{S}_\mathrm{c}$ set of all dual-failures scenarios where a direct span and a chain span are involved. In this case, the model is extended to convert this dual-failure situation into a *logical* triple-failure scenario.

$\boldsymbol{s_j}$ is the number of spare capacities that is placed on span $j \in \boldsymbol{S}$.

$\boldsymbol{w_i}$ is the number of working capacity that needs to be protected on span $i \in \mathbf{S}$.

CHAPTER 1 INTRODUCTION

As demand for data services has grown over the past decades, the majority of the world's global infrastructures, businesses, governments, and social systems in general become more and more dependent on global communication systems [53]-[54]. Such growing demand for this global communication infrastructure as Internet encouraged an increase in communication technologies and services [53]. Internet of things (IoT), network cloud services, network functions virtualization (NFV), and other emerging applications and services have brought a major increase on the volume of data being transported over the core transport networks [1]. In fact, the international internet bandwidth has grown worldwide by 68% between 2014 and 2016, which was also mainly driven by the spread of mobile-broadband services [2]. Furthermore, internet video streaming and downloads have begun to take a larger share of bandwidth as it is forecasted to grow more than 81% of all consumer internet traffic by 2021 [3]. In this way, internet traffic continues to grow vigorously. In addition to this, a great stride is being made in expanding internet access across the world in order to accelerate the economic and social growth of countries [4]. All of these facts suggest that we are likely to move towards a new massive demand trend, which is forecasted to accelerate within the following years and lead to a considerable increase in communication traffic in core networks.

Transport networks (also called *backbone networks*) provide carriage for different types of communication services such as voice, data, and video [51]. They are the core of the global telecommunication service as a transmission facility-based provider [55]. Generally, these transports networks are made of fibre optic cables connected to nodal switching devices such as *optical cross-connect switches* (OXCs) and *optical add-drop multiplexers* (OADMs) [5]-[6]. In addition, this network can be segmented into *access, metro,* and *core* (*long-haul*) *networks* which are primarily based on the political, administrative, and/or geographical boundaries [7]. The access partition of the network refers to the segment between a customer location and its first communication service provider or central office. Similarly, metro networks, or metropolitan networks, connect main switching offices among the same metropolitan area. Likewise, core or long-haul networks usually connect metropolitan networks covering distances that span from a city size up to a whole continent as, for example, an undersea cable [52]. Optical networking certainly dominates in metro and long-haul networks, which can carry high amount of data traffic covering huge distances. However, the use of optical networking in the access network segment has started to grow [8]. These networks are further organized into network layers that consist of nodes and links. These nodes, in turn, usually consist of switching or cross-connect equipment and these links or spans typically consist of logical adjacencies between the equipment. In this sense, all these networks can be thought of as a stacked arrangement of one on top of one another.

Through transport networking, point-to-point transmission systems are managed to create logical or virtual network environments for other services. They operate virtually as if they do have their own dedicated transmission, but in reality, they are just one of several service layer networks supported by one underlying structure – the transport network. Internet connections, voice calls, video calls, online shopping, credit card payments, and other services are not separate physical fiber optics networks as they do not make their own way over the fiber systems. Instead, a combination of all traffic types from site to site are formed by multiplexing these into digital signal streams [9]. Figure 1.1(a) conceptually illustrates a basic terminal multiplexer that provides aggregation of traffic, which comes from low capacity physical streams (wire, cooper cables), into a single physical high capacity medium (fiber). Later on, as fiber optic transmission and eventually *Wavelength Division Multiplexing* (WDM) emerged, the bandwidth of a single fiber escalated [10]-[12]. Optical networks based on wave-division multiplexing, in particular *dense wavelength division multiplexing* (DWDM), offer huge point-to-point capacities that allowed transport networks to handle massive amounts of data traffic [13]. In fact, today's optical networks can operate at terabit per second and they can reach operating rates as high as 100 Gb/s onto a single fiber [14]. Thus, the success of these networks as a fast communication system provider enabled emerging countries to join the digital economy and improved the quality of life for people all over the word.

In fiber optics transport networking there are two components that allowed a significance improvement in their wavelength routing operations, which are the optical add/drop multiplexer (OADM) and the optical cross-connect (OXC) [53]. These components allow optical networks to be reconfigurable at the wavelength level. The OADM, also called a wavelength ADM (WADM) nodal device is capable of interconnect two optical main lines (typically referred to as East and West lines), as well as allows wavelengths to be added or dropped (remove) from a channel as other wavelength are passed through the network [5]. By this manner, aggregation of traffic coming from different sources and to be carried on city-to-city can be impressed on a single fiber. This also allows a general network control at a wavelength granularity level in the optical layer. Although this device is capable of full wavelength conversion, many OADM do not used this capability and tend to be use purely for adding and dropping wavelength channels [7]. On the other hand, a fully reconfigurable OADM (also referred to as ROADM) provides a complete wavelength routing capability [7]. In this way, an OADM can be considered as a specific type of an optical cross-connect device. Figure 1.1(b) conceptually illustrates a simple optical add/drop multiplexer. The OXC is conceptually illustrated in Figure 1.1(c) and is nodal device that is capable of cross connect any input fiber to any output fiber, perform switching in a secondary dimension, and locally add and drop channels. The importance of OXCs is that they are capable of switching channels between multiple fibers to provide the desired connectivity across the network as a difference between OADM that only allows the connection between two fibers, as for example, in a linearly connected architecture such as a bus or a ring. Because of these capabilities of switching from one wavelength to another and therefore from one fiber to another in space, OXCs devices are essential in optical mesh networking. Ultimately, both systems are of significant importance for optical networking as they can provide network survivability through optical restoration and protection by switching to alternative routes in the optical layer.

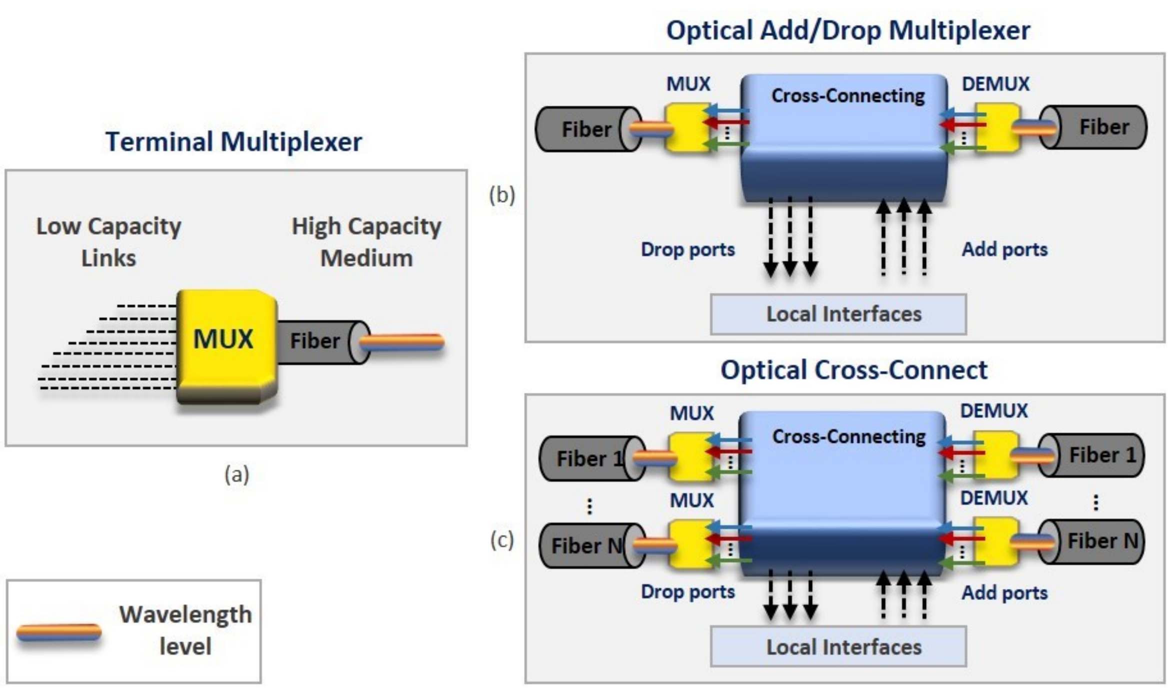


Figure 1.1 – Basic transport networks elements

With the always-increasing amount of critical data being carried over communication networks we are now dependent on the availability of these networks as on any other basic services such as electricity, water, and transportation. As a result, with so much relying on this communication infrastructure, reliability and availability of transport networks have been of significant concern [15]-[16]. Failure in these core networks, on either spans or nodes, can cause severe economic and social consequences [17], [7]. Usually, network equipment operating on the network nodes gets more physical protection and redundancy than network cables or links, which are more susceptible to environmental disruptions. Additionally, node equipment is carefully monitored as well as provided with fast restoration in case of failure [18]. Hence, span failures are what network designers and service providers are concerned about. Potential causes of span failures can be classified as engineering activities (e.g., dig-ups, maintenance, construction, etc.), natural disasters (e.g., flood, typhoon, fire, etc.), and willful deeds (e.g., terrorism activities, deliberate sabotage, etc.) [19]-[21]. Not surprisingly enough such events, in particular cable-cuts due to construction dig-ups, are the most frequent [7]. Figure 1.2 provides a brief summary of the main effects of a network service outage by reflecting the outage consequences as the outage time increase. As can be seen, the most desirable goal is to keep any communication interruption less than 50ms because any period of outage time longer than that the network will start dropping voice calls [17]. Nevertheless, in a world dominated by data services rather than voice traffic, this 50ms outage time may not be a rigid requirement anymore. In this case, network operators may be willing to tolerate up to 10s of outage time as it is after this time period that the network will start terminating data sessions. In addition, note that there are only few restoration mechanisms capable of ensuring a 50ms restoration time [7]. One would be *1+1 automatic protection switching* (APS) but this type of restoration mechanism is now typically used for the most critical services only [7]. *P-Cycles* corresponds to another type of network survivability mechanism capable of providing 50ms restoration time as combine ring speed with mesh efficiency [56]. Ultimately, any failure beyond 5 minutes would have a significant impact not only on the business sector but also on the society as a whole. Therefore, today's goal is a virtually instantaneous recovery against the most significant and frequent types of failures – spans or links failures. As a result, providing the proper amount of protection and/or restoration to this communication infrastructure is essential not only to meet customer expectations for services but also to carefully allocate network capacity and cost for services.

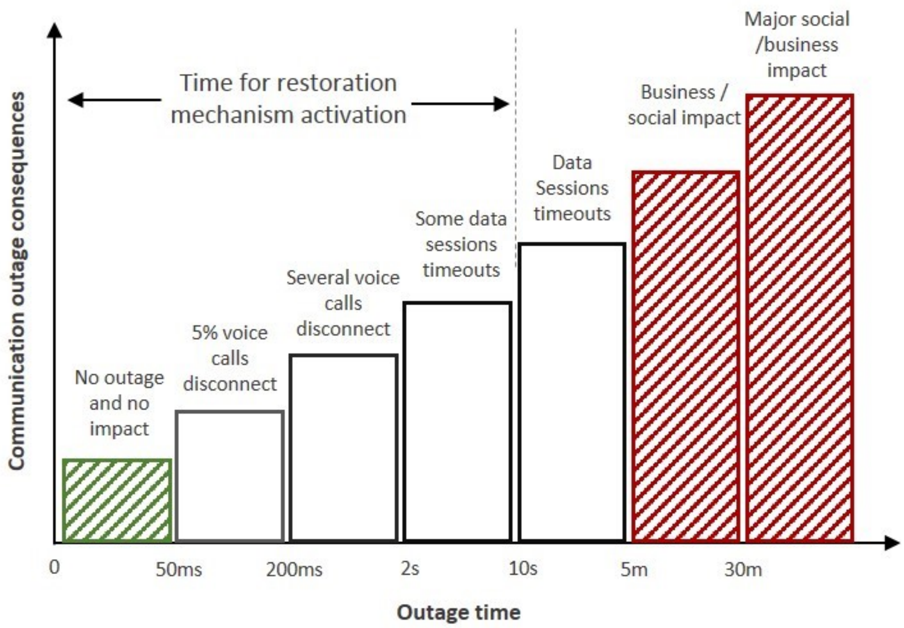


Figure 1.2 – Impact of network outage [17], [7]

Generally, when a span failure or a number of it occurs in a network, the communication path to a specific destination might become unavailable. In those cases, a network restorability mechanism has to find an alternate route around the failure to support service continuity. This is what is referred to us as *network survivability*. In this way, thanks to survivable network schemes, contemporary transport networks are provisioned to withstand full single link failure (i.e., restorability to any possible single fiber cut). Networks using such survivability schemes have been known as *fully restorable networks* or *fully protected networks* [22]. *Span restoration, path restoration, demand-wise shared protection* (DSP), *shared-backup path protection* (SBPP)*, p-cycles,* etc. are some of the examples of such restorable mechanisms [7]. Among the various network restoration and protection mechanism, span restoration is a particular attractive choice due to its simplicity and easy implementation. In span restoration, span failures are protected by a set of diverse replacement paths between the immediate end-nodes of the failed spans [27]. It is important to emphasise that the efficiency of this type of restoration mechanism is strongly dependent on highly connected network topologies (e.g., networks with high average nodal degree), where a larger degree of sharing between the immediate end nodes is allowed. Consequently, a very sparse network topology (low average nodal degree) can make the economic advantage of this restoration mechanism debateable. In an effort to provide enhanced mesh restoration capacity efficiency, the *Meta-Mesh* restoration approach was developed in [23]. In the meta-mesh method nodes of high degree (degree-3 or higher) are only considered and each degree-2 node is structured in the way that the two spans incident on it are combined into a single span [23]. Note that it is only at this level of network design that true span-restorable spare capacity sharing can arise [23]. Thus, our research aims to improve the optimal design of the meta-mesh structure by providing a greater understanding and insight into this logical topology as well as present alternatives to the conventional method, which are already in use.

Depending on the network topology and availability targets, which most of the time are explicitly specified in the *service level agreement* (SLA) [24]-[25], providing protection or restoration against a single span failure might not be sufficient for most of the communication networks. Multiple failures, even though they are less frequent, can harm network services [26]. Consider for instance that a two-fiber optic cable cut in the Bell Canada network in 2017 caused more than 4 hours outage affecting a great percentage of the traffic entering and leaving Atlantic Canada [45]. Note that multiple concurrent spans failures do not only imply that the failures occur simultaneously. It is also likely that one span fails, and prior to this span being repaired, which on average can take several hours, another span on the same network also fails. Thus, there is thus a practical reason to be concerned with how transport network restoration mechanism reacts in a dual-failure scenario. The work herein we focus our attention on meta-mesh networks and our aim is to address the basic question of how well a meta-mesh network, designed for full single-failure restorability, can withstand dual-failure scenarios, which dominate network outage after single-failure situations.

1.1 THESIS OUTLINE

The remainder of this chapter gives a brief introduction of our research topic, followed by the thesis outline.

In Chapter 2, we introduce a variety of network survivability mechanisms as well as we give a review of previous work related to the meta-mesh network topological design and optimization. In addition, we highlight the general existing methods used to perform the availability analysis in span-restorable networks as well as some basic ideas related to it. This chapter aims to provide the reader with the tools, concepts, and terminologies used throughout this thesis.

In Chapter 3, we present the motivation and research goals as well as our test networks, demand models, and computation aspects. The proposed methodology is also presented in this chapter.

In Chapter 4, we study the meta-mesh span restoration technique, which is a design method for implementing span-restoration mechanism in sparse mesh network topologies. The analysis carried out in this chapter examines and compares the previous meta-mesh design and introduces a new design insight capable of reach a greater capacity efficiency in some meta-mesh network topologies.

In Chapter 5, we develop two design models that aim to optimize the dual-failure spare capacity on meta-mesh networks. The first ILP formulation model aims to find the minimum total cost of working and spare capacity to ensure full restorability of all dual span failures scenarios. The second ILP formulation model presented in this chapter seeks to find the number of non-restored working capacities of a meta-mesh network designed for full single-failure restorability. In this way, this chapter include a formulation for capacity minimization under the constraint of complete dual-failure restorability, and a formulation for restorability maximization under a given total capacity cost budget.

In Chapter 6, we summarize this thesis and list contributions of this study, followed by a brief description of possible future research opportunities.

CHAPTER 2 NETWORK SURVIVABILITY AND NETWORK DESIGN AND OPTIMIZATION

2.1 MESH NETWORK SURVIVABILITY MECHANISMS

As we briefly discussed in Chapter 1, the growth of the Internet, the increasing dependence of essential business functions that rely on communication systems, and the general social dependencies on communication networks, all make network survivability mechanisms a now-critical aspect of a communication network design. As a result, several network protection and restoration techniques have been developed over the past years [29]-[31], [5]. These survivability schemes can be generally classified as *pre-planned* or *protection* mechanisms and *adaptive* or *restoration* mechanisms. With protection mechanism, the operation is relatively simple and self-contained. The restoration paths taken for any failure are assigned prior to failure as these paths are completely dedicated to the nodes that they protect and ready to reroute working demand flow. Therefore, whenever a failure occurs, the activation of the protection path is fast. *Demand-Wise Shared Protection* [28], *1+1 Automatic Protection Switching* (1+1 APS) [29], *P-Cycles* [30], and *1:1 Automatic Protection Switching* (1:1 APS) [29] are few examples of this type of survivability mechanism. With restoration mechanism, on the other hand, the restoration routes are found adaptively based on the failure as well as the state of the network at the time of the failure. This is because the restoration mechanisms are more flexible and easier to adapt to unanticipated changes in the network (as a failure). It can be argued that one of the main difference between these two types of network survivability schemes is that replacement paths are known in advance on protection mechanisms, meanwhile in restoration mechanisms restoration path-sets have to be found in real-time. However, in modern survivability schemes there is not a concise distinction anymore. In mesh-based network, survivability schemes can be further classified as a combination between protection and restoration mechanisms where restoration paths are fully known before a failure, but spare channels are not connected until a specific failure arises [31]. This combination or what is refer to as *pre-planned restoration mechanisms* [7] can be thought as an intermediate version between a pure restoration and a pure protection mechanism.

Another basic and arguably assumption is that finding backup paths in real-time as in restoration mechanisms is always slow and that backup paths known in advance as in protection mechanisms will always be fast and secure. This assumption is not necessarily true as intermediate restoration schemes can operate with the *distributed preplanning* (DPP) algorithm to identify backup paths before a failure occurs [7]. With DPP restoration paths can be continually discovered in advance of a failure (i.e. backup paths are known before an unexpected failure hits the network) as on protection mechanisms [32]. This algorithm uses a series of made-up failures trial and records the set of local nodes that constituted their participation (if any) in the formation of restoration paths. In addition, the efficiency and speed as well as how fast or slow the survivability schemes in the intermediate survivability category works depends on weather which real-time performance dominates: the *path finding* time or the *cross-connecting*. That is, there might be intermediate mechanisms that backup paths are fully know in advance, but restoration channels are not cross-connected until a failure arises. Thus, it is necessary to avoid oversimplifications associated between a pure protection mechanism and a pure restoration mechanisms and take advantage of the range of possibilities between these categorization, particularly on speed and availability.

Even though various survivability mechanisms are applied in mesh network designs, we only focus on and describe the ones that are employed in this thesis.

2.1.1 End-to-End Restoration and Protection Mechanisms

2.1.1.1 Automatic Protection Switching (APS)

The automatic protection switching (APS) systems is perhaps one of the simplest network survivability mechanisms [37]. Formerly, there are different ways that this mechanism can operate. 1+1 *diverse-protection* (DP) APS is a dedicated end-to-end protection mechanism where the working route is duplicated in a reserve backup route in case of failure and is routed over a different physical path. Figure 2.1 illustrates how 1+1 APS protection functions, where the traffic signals between nodes A and B are simultaneously transmitted between two paths (path 1, which corresponds to the working path is illustrated in a solid line, and path 2, which corresponds to the backup path is illustrated in dashed lines). Subsequently, 1:1 APS is a variation on 1+1 APS, with the difference that the transmission signal is not broadcasted on the second route at the same time. Under this scenario, the protection channel can be use for other purposes when is not needed. The 1:1 APS brought the M:N APS scheme, where M restoration paths provides restoration of working capacity from N routes. In this case, an example can be a 1:7 APS protection scheme where one restoration path provides protection to seven working paths (where the risk is primarily of a single failure scenario).

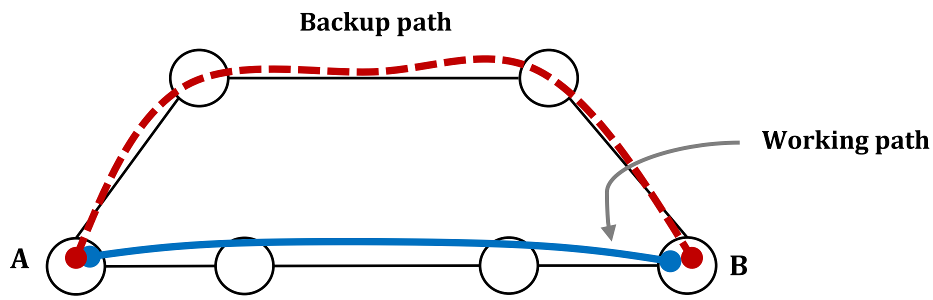


Figure 2.1 – An instance of 1+1 automatic protection switching

2.1.1.2 Shared Backup Path Protection (SBPP)

Shared backup path protection (SBPP) is another protection mechanism that in fact is very similar to the already mentioned 1:1 APS mechanism [38]. SBPP is a pre-planned end-to-end path protection mechanism that defines a single disjoint backup path for each working path in a case of a failure. Figure 2.2(a) portrays two nodes pairs sharing traffic signals, nodes A-B as well as nodes C-D, which routes their demands through a primary working path, represented by solid lines. In addition, Figure 2.2(b) illustrates each of the disjoint backup path of the mentioned nodes, which are represented by dashed lines, for restoration in the case of a failure of any span between the primary routes. In this mechanism, as on 1:1 APS, the traffic is transmitted on the working path only leaving the backup path available for other use when is not needed. It is because of this that SBPP offers great advantages on spare capacity since the backup path can be shared between different fully disjoint working paths. As can be seen in Figure 2.2(b), both backup routes share the same span S1 for their restoration in case of a span failure. Furthermore, since the protection is performed between the original-destination (O-D) pair nodes rather than between the end nodes of the failed span, this protection mechanism is slightly more efficient than span restoration [39].

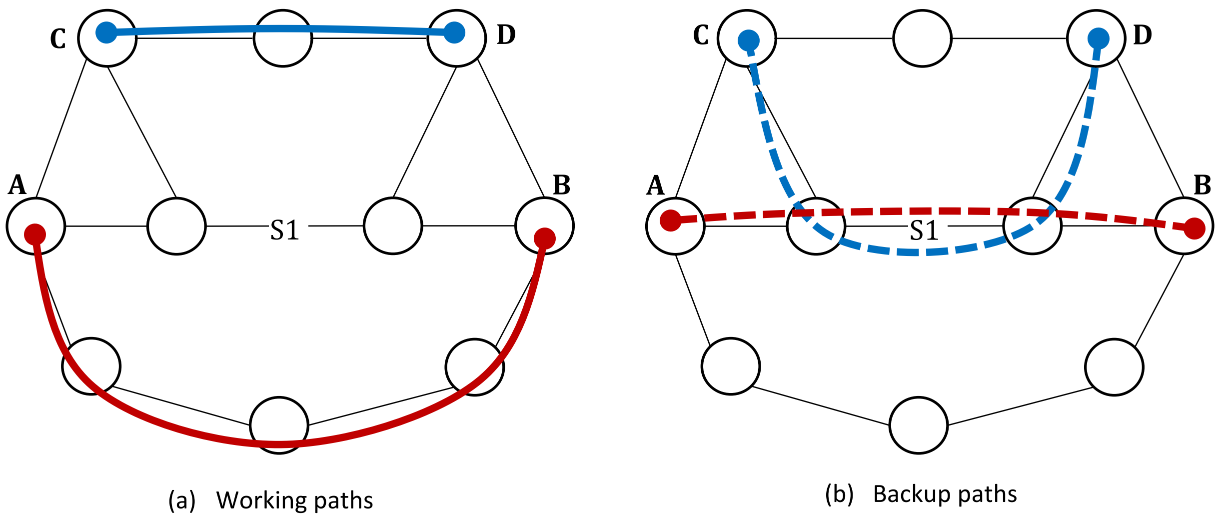
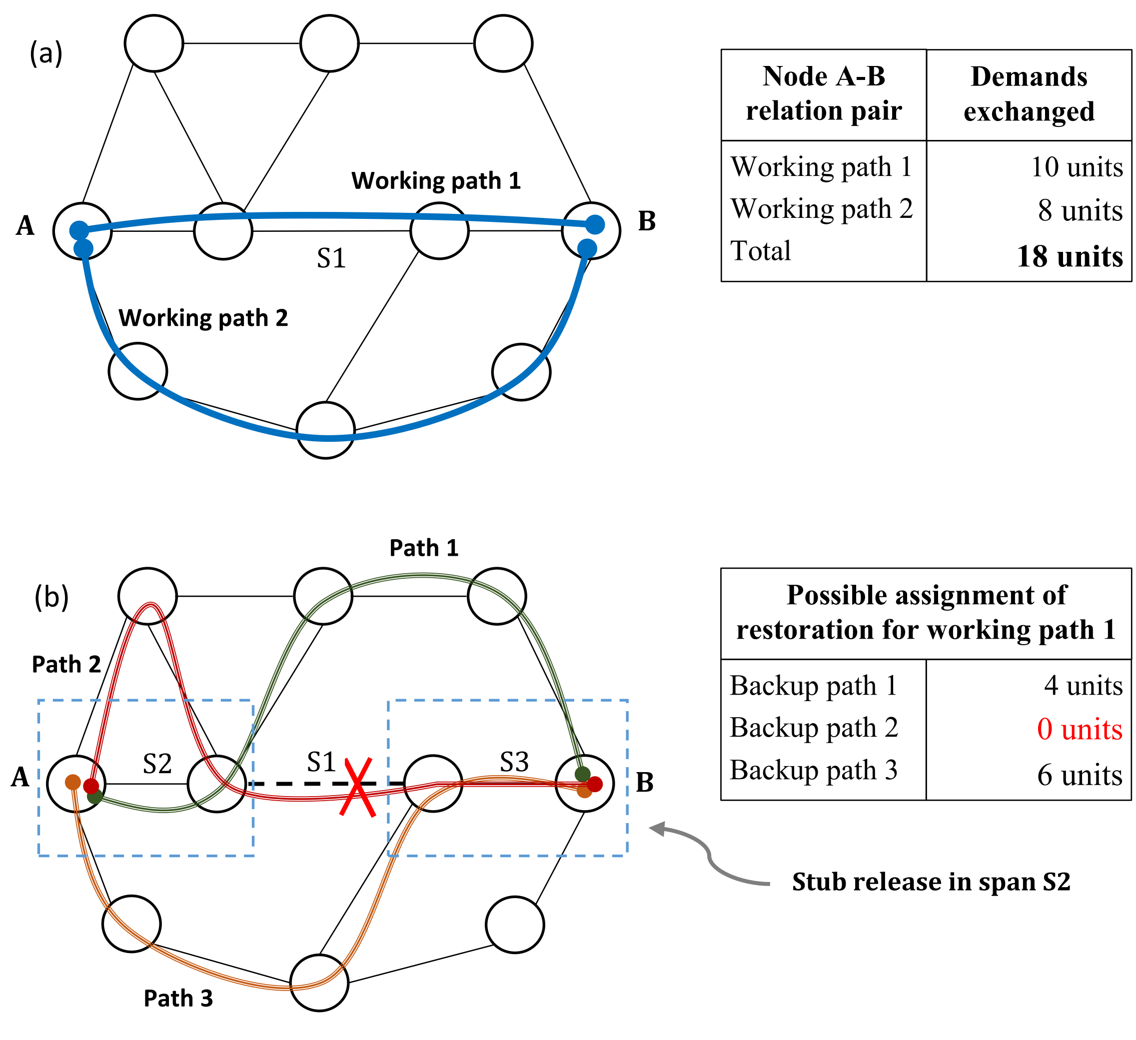


Figure 2.2 – An instance of shared backup path protection

2.1.1.3 Path Restoration

Another widespread end-to-end survivability mechanism is path restoration [40]. In path restoration, the backup paths for each working path are searched among the path's end-nodes. In this case, path restoration operates in a similar way than SBPP by offering restoration between the original and destination pair nodes. However, path restoration utilizes the *stub-release* mechanism, which allows restoration paths to use the working capacity of failed paths on spans that are not involved in the span failure [34]. In other words, the survivable portion of the affected working path is rapidly released before restoration and this allows the re-use of this capacity by restoration paths of other demands. Among all the restoration mechanisms already discussed, path restoration is the most efficient because of its stub-release mechanism. Nevertheless, their implementation is more complex than other network restoration mechanism. Figure 2.3 illustrates an example of a general path restoration framework. Figure 2.3(a) shows a node pair or "relation" between nodes A-B exchanging demands over two different working paths, which are represented by a solid line and are transiting between different spans in the network. Ten demands units are being exchanged in working path 1 and eight demands units are being exchanged in working path 2, given a total of 18 demands units. If an unexpected failure in span S1 occur, only the working path 1 is going to be affected. In this case, Figure 2.3(b) illustrates three eligible backup paths for relation pair A-B but only two of these remain feasible under this failure situation. As can be seen in Figure 2.3(b), the backup paths can effectively use the remaining portion of working capacity left over by the primary working path 1 in spans S2 and S3 as a spare capacity for their restoration paths. This stub release factor contributes to significant amount of saving in spare capacity needed for restoration [40]. Ultimately, a possible assignment of restoration flow to surviving eligible routes can be (4, 0, 6) to routes (1, 2, 3) respectively.



Node A-B relation pair	Demands exchanged
Working path 1	10 units
Working path 2	8 units
Total	**18 units**

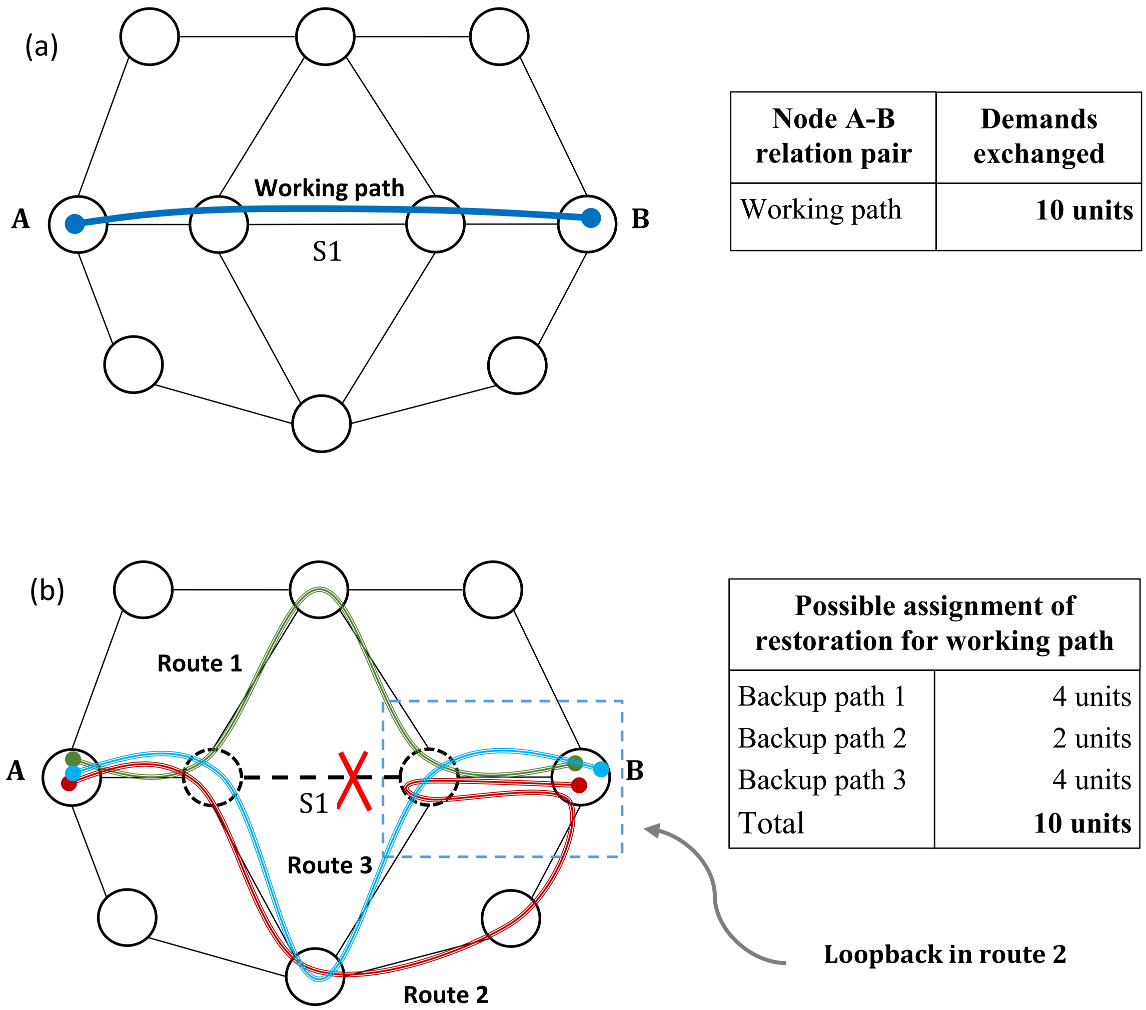
Possible assignment of restoration for working path 1	
Backup path 1	4 units
Backup path 2	0 units
Backup path 3	6 units

Figure 2.3 – An illustration of path restoration design

2.1.2 SPAN RESTORATION

Span restoration (SR) is a localized mesh restoration mechanism that uses a set of restoration routes around the immediate nodes of the break to restore the failed span [27]. Foremost among other mesh restoration schemes, span restoration offers great advantages in terms of simplicity and speed because it is only the status of each working channel on a failed span that needs to be known. For this reason, span restoration is more efficient than 1+1 APS and ring-based routing mechanism for example, as they require a significant portion of its total capacity dedicated for spare capacity to be employed when a failure occurs. However, it is less efficient that path-oriented protection or restoration mechanisms due to its localized response to network failures. Another key point is that span restoration can be achieved by incorporating distributed preplanning (DPP) algorithm where restoration paths can be continually discovered in advance of a failure [32]. Span restoration with distributing preplanning (SR-DPP) is a powerful technique that can be further classified as a preplanned restoration mechanism.

The general idea behind span restoration is illustrated in Figure 2.4. In the event of span S1 outage, different restoration paths will be "available" at that moment to seize the failure. These different restoration paths are known as restoration "path-set" for the respective failure. Note that depending of the number of working channels (i.e., w_i if span i fails) to be protected on span S1, the restoration does not need to be via a single route, nor via two-way routes, rather it can it be formed through any number of distinct routes among the path-set. In other words, the restoration of span S1 is not limited to using a single replacement route for all failed working capacity w_i. It may, as needed, involve paths on all distinct routes, up to a defined hop or distance limit predefined in the network design, in order to efficiently use the spare capacities available on other spans. With this in mind, span restoration operates at a granularity level deploying a set of reroute backup paths around the failure scenario. Figure 2.4(a) shows a working path from a node pair relation between nodes A-B exchanging 10 units of demand. As it is illustrated in Figure 2.4(b), in case of failure of span S1 three restoration routes are available to route demands allowing an efficient use of spare capacity available on other spans in the network. That is, restoration route 1 carries four units of demand, restoration route 2 carries two units of demand, and restoration route 3 carries four units of demand. In this way, all 10 units of demand are successfully restored. Note that this possible assignment of restoration flow between these different backup routes is illustrative only and it might to be the optimal solution to this problem. In addition, restoration route 2 shows a span restoration solution in which a loopback arises between node B and one of the nodes adjacent to the failed span S1.



Node A-B relation pair	Demands exchanged
Working path	**10 units**

Possible assignment of restoration for working path	
Backup path 1	4 units
Backup path 2	2 units
Backup path 3	4 units
Total	**10 units**

Figure 2.4 – An illustration of span restoration design

2.1.2.1 JOINT CAPACITY ASSIGNMENT (JCA) MODEL

The following *joint capacity assignment* (JCA) model is the basis for all network capacity design models presented in this thesis. The problem itself seeks to minimize the total cost of spare and working capacity in a span-restorable mesh network [7]. This model introduces the working path routes as a decision variable in the sense of making the relate survivability design less costly. The important point is that the total cost affects both working and spare capacity, so any reduction in total spare capacity should be related to a deviation of the working path routes from their shortest paths. Restoration routes is predefined and may arise from a hop limit factor that restraints the maximum length, in nodes, of a reroutes signal path [27]. Hence, the basic arc-path formulation of the JCA problem in a span-restorable network uses the following notation:

Sets:

S is the set of spans in the network, and is usually indexed by i when referring to a failure span and j when referring to a surviving span or to enumerate all spans in general.

$\mathbf{P}_i$ is the set of eligible restoration or backup routes for span $i \in \boldsymbol{S}$ under single-span failure scenario. Normally, it is indexed by p.

D is the set of demand quantities for each service path relation in the network. Usually, it is indexed by r.

$\mathbf{Q}^r$ is the set of all eligible routes available for carrying demands for each relation r. Usually, it is indexed by q.

Input Parameters:

C_j is the cost of each unit of capacity on span $j \in \boldsymbol{S}$.

$\delta_{i,j}^{p}$ is a binary variable that is equal to 1 if the p^{th} eligible restoration route for span i uses span j, and is equal to 0 otherwise $\forall (i,j) \in \boldsymbol{S}^2, \forall p \in \boldsymbol{P}_i$.

d^r is the amount of demands units for demand relation r.

$\zeta_j^{r,q}$ is a binary variable that is equal to 1 if the q^{th} eligible working route for relation r uses span j, and is equal to 0 otherwise.

<u>Decision Variables</u>:

s_j is the number of spare capacities that is placed on span $j \in \boldsymbol{S}$.

w_i is the number of working capacity that needs to be protected on span $i \in \mathbf{S}$.

f_i^p is the amount of flow routed on restoration route p for restoration of span i under a single-span failure scenario.

$g^{r,q}$ is the number of working flow allocated to the q^{th} eligible working route used for demand pair r.

The ILP formulation of the joint span-restorable (JCA) model itself proceeds as follows:

$$Minimize \sum_{j \in \mathbf{S}} C_j \cdot (s_j + w_j) \tag{2.1}$$

Subject to:

$$\sum_{p \in \mathbf{P}_i} f_i^p = w_i \qquad \forall i \in \mathbf{S} \tag{2.2}$$

$$s_j \geq \sum_{p \in \mathbf{P}_i} \delta_{i,j}^p \cdot f_i^p \qquad \forall (i,j) \in \mathbf{S}^2, i \neq j \tag{2.3}$$

$$\sum_{q \in \mathbf{Q}^r} g^{r,q} = d^r \qquad \forall r \in \mathbf{D} \tag{2.4}$$

$$w_i = \sum_{r \in \mathbf{D}} \sum_{q \in \mathbf{Q}^r} \zeta_j^{r,q} \cdot g^{r,q} \qquad \forall i \in \mathbf{S} \tag{2.5}$$

The objective function (2.1) minimizes the total cost of assigning working and spare capacity for each span failure in the network. Generally, it uses i to designate a failure span and j to designate other spans not involving itself as a failed element. Thus, the constraint set in (2.2) places sufficient restoration flow in all eligible restoration routes to ensure full working capacity restorability of the affected single-span failure i. Furthermore, constraint set (2.3) guarantees enough amount of spare capacity on every surviving span j for all restoration routes placed on them during all single-span failure scenarios i. Equation (2.4) ensures that the working flow assigned to each eligible working route for relation r fully routes the total demand of each relation service path. Note that this indicates that the total demand may be divided over several possible different routes as restoration routes do with spare capacity. Equation (2.5) implies that every span single-failure i working capacity must be sufficient to meet the pre-failure demands of all pair relations r that have working flow across it.

The *spare capacity assignment* (SCA) model is a basic alternative to the mentioned JCA model and seeks to only find the total spare capacity needed to assure full restorability of all single-span cut in a span-restorable mesh network. In this model, the working capacity of each span is calculated or assumed beforehand (i.e., they are given as an input). These w_i quantities can be obtained from shortest path routing of the working demands over the network graph, or from any other demand routing process. Depending of the source, this problem can also be referred to as the spare capacity placement (SCP) problem, the *spare-channel* design problem, among others [33]-[34]. This model can be obtained by setting up w_i as a parameter instead of a decision variable as well as eliminating the following sets, parameters, and decision variables from our notation: **D**, $\mathbf{Q}^r$, d^r, $\zeta_j^{r,q}$, and $g^{r,q}$, respectively. In addition, equation (2.4) and (2.5) would be removed from the JCA ILP formulation model already presented as well as the decision variable w_i from the objective function (2.1). Ultimately, the joint capacity assignment (JCA) model offers a significant reduction in total capacity with relation to the SCA design [35]-[36], and is the basis for all network capacity formulations presented in this thesis.

2.2 TOPOLOGICAL $1/(\bar{d}-1)$ LOWER BOUND REDUNDANCY

The *capacity redundancy* is a common measure of a network's efficiency, and is defined as the fraction of the total amount of spare capacity divided by the total amount of working capacity over all spans of the network [5], as in Equation (2.6). The lower bound of this capacity redundancy can provide a high-level prediction of necessary but not assuredly sufficient spare capacity to provide a span-restorable mesh network with complete single failure restorability.

$$R_{cap} = \sum_{\forall i \in S} s_i \Big/ \sum_{\forall i \in S} w_i \tag{2.6}$$

The topological lower bound is derived from the understanding of how restorability works in an isolated node within a span-restorable network [5], [36]. Consider the failure of span 1 with w_1 working capacity units adjacent to the node of degree d in Figure 2.5. From the restorability standpoint, there must be enough spare capacity in all surviving spans adjacent to the failed span in order to support complete restoration of the entire working capacity placed on the mentioned failed span 1. Likewise, each span i adjacent to the node requires the allocation of sufficient spare capacity in all surviving spans to restore all failed working capacity placed on it. This observation leads us to realize that each span of this node could have $w_i = w_1$ and spare capacity can potentially be distributed evenly on all spans. In this case, the fraction of spare and working capacity becomes our lower bound, which is simplified in Equation (2.7) [36].

$$\sum_{\forall i \in S} s_i \Big/ \sum_{\forall i \in S} w_i = {d \cdot w_1/(d-1)} \Big/ {d \cdot w_1} = \frac{1}{\bar{d} - 1} \tag{2.7}$$

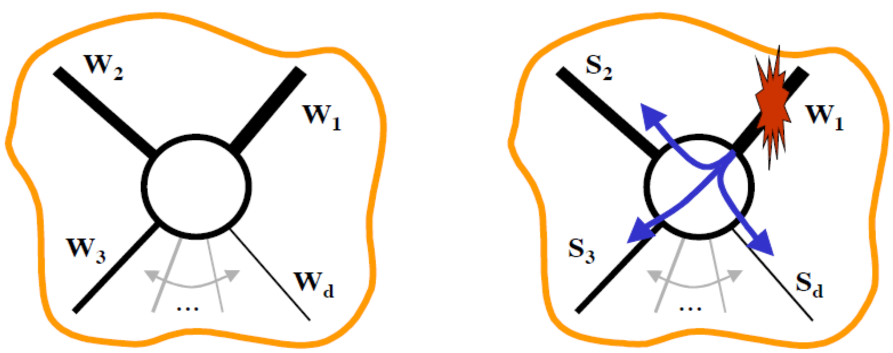


Figure 2.5 – Basis of a lower bound of redundancy derivation in a span-restorable network [5].

2.3 Meta-Mesh Network Design

Advances in WDM transmission technologies and optical switching technologies brought the ability of networks elements to allow the use of mesh restoration mechanisms on the optical networking layer [16]. In mesh-based networks, transport switch elements such as optical cross-connect switches (OXCs) and optical add-drop multiplexer (OADMs) are now more sophisticated because it is not only the add/drop function for each fiber that they provide but also because they require to switch channels among fibers in order to provide connectivity across the network. In this way, these particular optical switching elements have the ability to switch the path of a particular wavelength and reconfigure it at the node level. These technological advances allowed the development of new network restoration designs and provided new options and insights in the field of network survivability.

As it is widely acknowledged, mesh-based restoration schemes offer a greater capacity efficiency as well as operational flexibility than ring-based restoration schemes. This is because survivable ring architectures typically involve a structure of working and spare capacity that is configured in the form of a ring. In this case, lightpaths are routed through the rings, not necessarily via shortest paths, where half of the capacity is used for working traffic and half if used for backup paths [37]. Rings are typically viewed as a fast and simple protection mechanisms. On the other hand, mesh restoration approaches generally offer a greater capacity efficiency by routing working lightpaths via shortest paths and sharing working capacity amongst multiple working lightpaths [7]. In addition, while ring-based networks and restoration schemes dominated during the mid-1990s, a great strive was made on increasing the *network average nodal degree* in existing long-haul networks by acquiring more rights-of-way. The average nodal degree of a network is a transport networking terminology that refers to the average degree of all spans in a network graph. This can be calculated by $\bar{d} = 2 \cdot |S|/|N|$, where N is the total number of nodes in the network and S is the total number of spans in the network. However, achieving a greater capacity efficiency in sparse network topologies, as for example in some North American interexchange carriers (IXC) networks, represented a tremendous challenge case. In this sense, during the ring-to-mesh transition, chains subnetworks (e.g., a series of degree-2 nodes) were becoming more noticeable in this sparse network graphs. For instance, Sprint Communications' USA backbone network, shown in Figure 2.6, has an average nodal degree of approximately 2.4 [41], [57]. As can be seen, due to the sparseness of the network graph several chains are formed which make questionable the used of mesh-based restoration mechanisms in this network topology design. European networks, on the other hand, are beneficiated of mesh-based restoration mechanism due to their often have a high average nodal degree of approximately $\bar{d} > 4$ [5]. This is therefore an interested research area in optical transport networks that is focused on sparse network topologies.

Figure 2.6 – Sprint Communications' USA backbone network [41], [57] (used with permission)

2.3.1 Chain Optimized Mesh Design

As previously stated, chains are a natural feature of sparse network topologies graphs as it is evident in many areas of the Sprint Communications' USA backbone network from Figure 2.6 [41], [57]. In fact, these chains subnetworks are the main source of span restoration inefficacy on sparse networks topologies [5]. This is because the efficiency of this type of restoration mechanism is strongly linked with high-connected network graphs [23]. To explain this, we begin by showing the natural performance of span restoration in chains. By its own nature, the spare capacity destined for restoration placed on each node side must be enough to support the loop-back of the largest amount of working capacity across the entire chain [27]. This is because all working capacity on a span within the chain must be restored back through all surviving spans in the chain to the *anchor node* (a degree-3 or higher node), and then back through the network. Figure 2.7 illustrates a three-span chain subnetwork as well as a set of working and spare capacities. In this example, node 1 and node 4 act as the anchor nodes of this chain. As can be seen, the amount of spare capacity required for a single-failure restoration must be equal or exceed the working capacity of the other side of the node, except for the span with the maximum working capacity, which will need spare capacity equal to the second highest working capacity in the chain. Under these circumstances, if a failure of any span in the chain occurs (i.e., a failure in the span between node 2 and node 3 as it is illustrated), all failed working capacity is sent back or looped-back in the opposite direction until it reaches an anchor node and then back through the network. This is thus one of the main reasons of the relative inefficiency of span restoration in a sparse network graph.

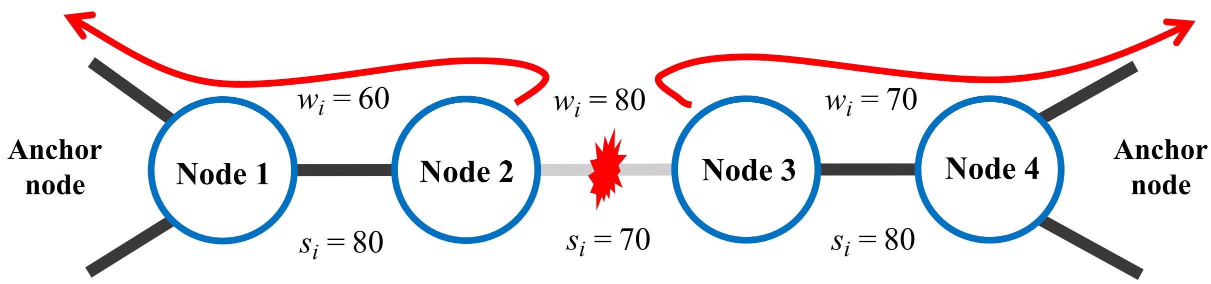


Figure 2.7 – Spare capacity requirements in a chain using span restoration

A closer look on the design of these chains will show us that there are two types of working capacities that travels across the chain itself. One would be an accumulation of working flow for some demands originating and/or terminating inside the chain and the other one is an accumulation of working flow for some demands that pass completely through the chain. Therefore, only some of the working capacity on the spans of the chain will arise from working traffic that originates and/or terminates at one of the chains within the span. Hence, a breakdown of these demands that originate and/or terminate at the chain is made [23]. In this case, if a demand is travelling along the chain in its entirety and is destined to a node outside of the chain, or at one of the anchor nodes, it is referred to as *express flow* working capacity (W_{EXP}). The remaining portion of the working capacity for one of the degree-2 nodes within the chain is referred to as *local flow* working capacity (W_{LOC}). Figure 2.8 illustrates an example of this distinction. As can be seen, the maximum amount of W_{LOC} or intra-chain working capacity would be 45 units. Therefore, 35 units of express working capacity travel entirely through the chain on their way to/from other nodes located elsewhere in the network graph. In other words, if the local working flow and total working flow values are known, any difference remaining must be express working flow.

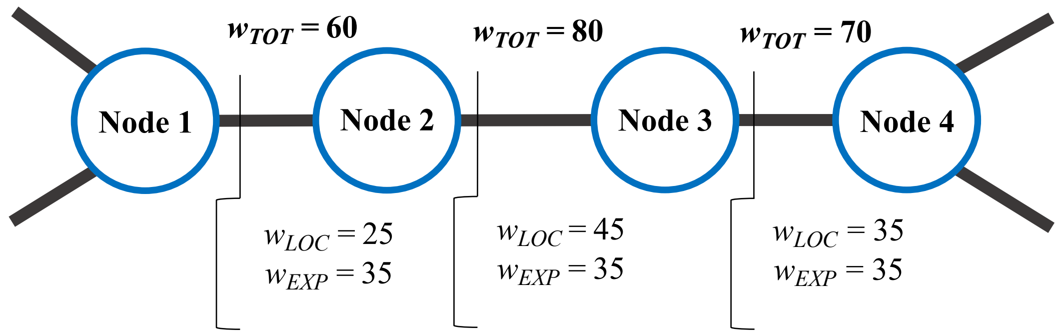


Figure 2.8 – Chain optimized design breakdown of working capacity into local and express flow

As mentioned, under normal span restoration the entire chain should have enough amount of spare capacity to assure the loopback routing of the largest amount of working flow in the chain. As a result, the normal response to any span cut under span restoration is to send back all local and express flow to the anchor nodes which is equal to $W_{TOT} = 80$ in the example above. Notice that the W_{LOC} are intra-chains working capacity demands that originate and/or terminate among the chain. There is thus no necessity to loopback the express flow to the anchor nodes because the express flow is not destined to a node inside the chain. Rather, the express working flow could be failed back right on the anchor nodes as it is shown in Figure 2.9. This yields a significant reduction of spare capacity, as it is only the intra-chains demands that are going to require spare capacity for restoration [23]. In other words, the entire chain subnetwork can be thought of as a logical express route in which these demands are travelling. As can be seen in Figure 2.9, after meeting this distinction, only 45 units of spare capacity are required instead of the 80 units originally needed. Ultimately, there must be enough spare capacity into the chain to loopback the largest amount of local working flow which corresponds to those 45 units in Figure 2.9.

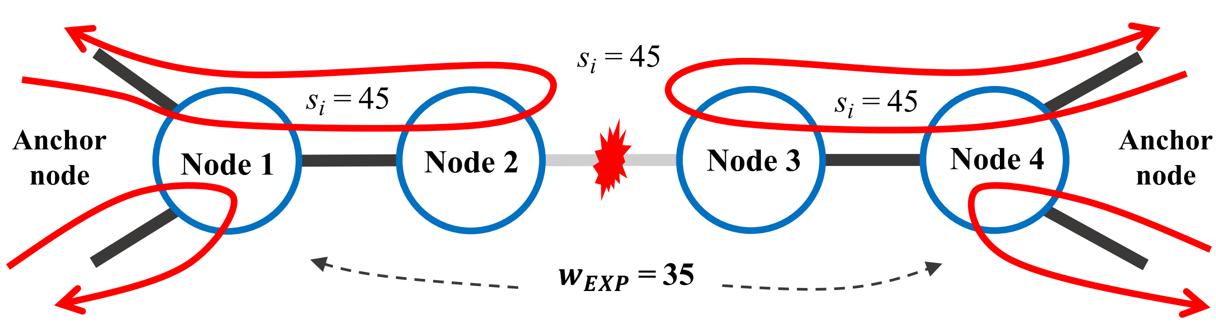


Figure 2.9 – Spare capacity requirements under the optimized chain structure

2.3.2 THE META-MESH CONCEPT

After presenting the optimized chain design and discussing its benefits, we now define the *meta-mesh* network design. The meta-mesh network corresponds to a logical structure where the spare capacity required for restoration in low-average nodal degree networks is reduced by improving the manner in which chain subnetworks or a series of degree-2 nodes are restored. In this sense, the meta-mesh is the network graph where only nodes of degree-3 or higher are considered and every degree-2 node inside a chain is combined into a single span. Figure 2.10 portrays the meta-mesh topology of the Sprint Communications' USA backbone network from Figure 2.6. Meanwhile the original backbone network has 264 nodes, 312 spans, and $\bar{d} = 2.36$, the meta-mesh topology graph has only 77 nodes, 123 spans, and $\bar{d} = 3.21$. The advantage of this topology design is that it is at this level of abstraction that the network average nodal degree increases up to $\bar{d} = 3.21$ as a difference of the original topology with $\bar{d} = 2.36$. A simple application of the $1 / (\bar{d} - 1)$ lower bound on redundancy can show us the potential of efficiency behind this design [36]-[43]. The lower bound of spare capacity redundancy of the original network in Figure 2.5 is 74%. While the meta-mesh graph in Figure 2.10 could have a lower bound of only 45%.

Figure 2.10 – The meta-mesh of the Sprint Communications' USA backbone network [58]

2.3.3 Logical Chain Bypass Span

In a closer look at the make up of the meta-mesh design, particularly after the breakdown of the local and express flows, we can observe the presence of an express route or a *logical chain bypass span* that will handle any working flow that completely travels the chain. In other words, this chain bypass span can be though of as an express route for all the W_{EXP} demands travelling through the chain. This logical bypass span represents an express routing option for working flows, in this case express flow W_{EXP}, that does not have the span restoration side effect of contributing to the loopback spare capacity requirements in the chain. For obvious reason, the length or cost of transiting the entire logical chain bypass span is going to equal to the sum of the lengths or costs of all the spans among the chain. In this way, the express flow is still routed over the same physical route (e.g., fibre optic cables) of the chain but it is not being handled by the optical add/drop multiplexers (OADMs) across the chain. Rather, it is handled by the optical cross-connects (OXCs) components at the anchor nodes. As was mentioned in Chapter 1, the importance of OXCs devices is their capability of switching between multiple fibers to provide the desired connectivity across the network as a difference between OADMs devices that only allows the connection between two fibers, as for example, in a linearly connected architecture such as a bus or a ring. As Figure 2.11 shows, a logical bypass span is allocated between the anchor nodes that is between node 1 and node 4.

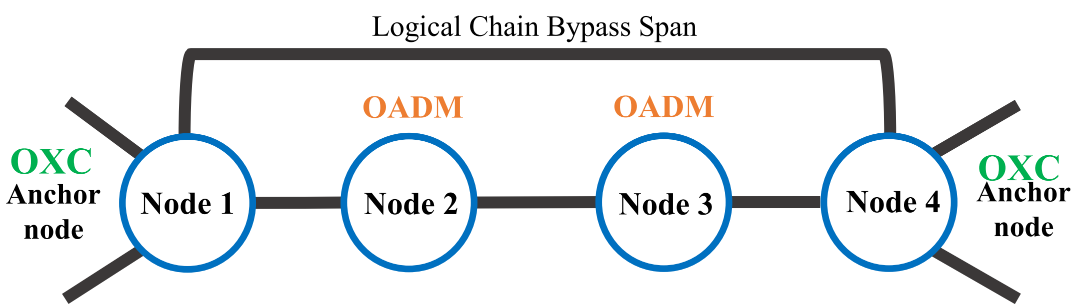


Figure 2.11 – Meta-Mesh breakdown of working capacity into local and express flow with its bypass span

From a transiting lightpath standpoint, Figure 2.12 portrays a more elaborated example of a span-restorable meta-mesh design. For simplicity, let us consider the existence of only two pair of nodes exchanging one lightpath each, pair A-B and pair A-C, on the entire network. Figure 2.12(b) illustrates the normal response in the event of span S2 outage under conventional span restoration design. As can be seen, we would need to allocate sufficient amount of spare capacity in span S1 to restore all the working capacity placed on span S2. In this case, we would require spare capacity for two lightpaths to ensure complete single failure restorability. On the other hand, the meta-mesh design would only provide spare capacity for the intra-chain lightpath (pair A-B) and the restoration of the express lightpath (the lightpath that fully transit the chain, pair A-C) would occur at the meta-mesh abstraction of the original network topology, as in Figure 2.12(d). This is because the meta-mesh model is augmented to include a logical dual span failure between the span inside the chain and its corresponding logical chain bypass span. In this case, the logical bypass span B1 corresponds to the set of spans S1, S2, and S3. So, when span S2 fails, its corresponding bypass span B1 also fails.

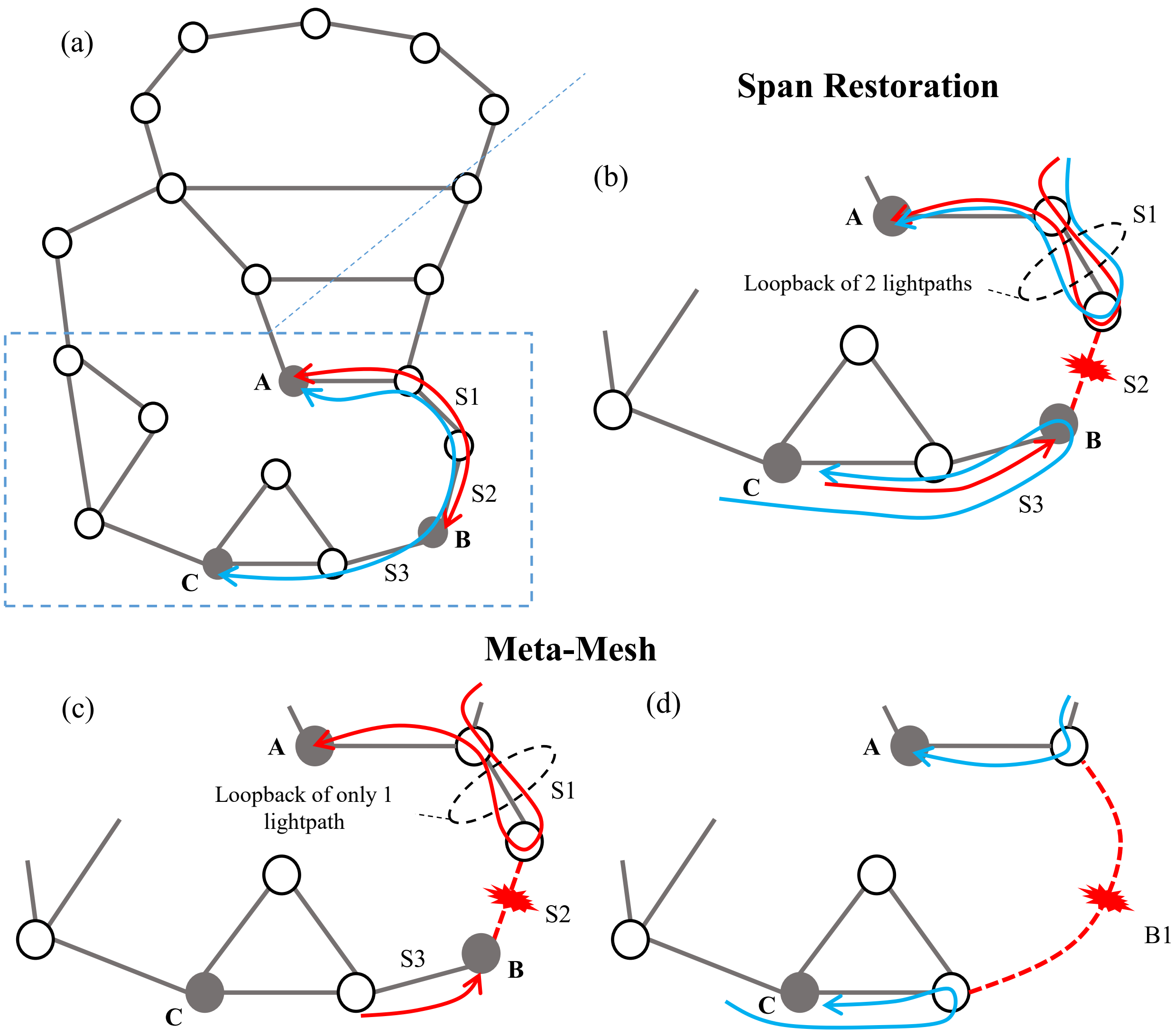


Figure 2.12 – Capacitated example of the meta-mesh design model

2.3.4 THE META-MESH ILP MODEL

By incorporating these new ideas and applying several changes to the conventional JCA design, the meta-mesh model can be obtained [5], [23]. First, the network topology file should be extended to include a logical bypass span in parallel with each chain in the network. As mentioned previously, the theoretical idea behind this is to have the option of transmitting working flow over an express route through the chain. Then, the mathematical model is augmented to convert any single span failure of the chain into a logical dual span failure scenario. In other words, if any span of the chain fails, its corresponding logical bypass span must fail as well. This is because this bypass span is no more than a logical representation of every span of the chain subnetwork. In order to represent these simultaneous logical span failures, the set of span **S** is now augmented as well as represented as follows:

<u>New sets</u>:

$\mathbf{S}_{\mathrm{d}}$ set of spans whose end-nodes are both of degree-3 or higher. So-called *direct spans*.

$\mathbf{S}_{\mathrm{b}}$ set of all logical bypass spans added to the network.

$\mathbf{S}_{\mathrm{c}}$ set of all spans that are part of any chain in the network.

<u>New Parameters</u>:

$k_i \in \mathbf{S}_{\mathrm{b}}$ is the corresponding logical bypass span with its associated span in the network. For example, if span S1, S2, S7, and S9 contain the chain whose bypass span is B2, then $k_{S1} = k_{S2} = k_{S7} = k_{S9} = B2$.

In its essence, the meta-mesh design is going to have all previous notation from the original span-restorable JCA design model formulation. The entire ILP is expressed as following:

Minimize $$\sum_{j \in \mathbf{S}} C_j \cdot (s_j + w_j) \qquad (2.8)$$

Subject to: $$\sum_{q \in \mathbf{Q}^r} g^{r,q} = d^r \qquad \forall r \in \mathbf{D} \qquad (2.9)$$

$$w_j = \sum_{r \in \mathbf{D}} \sum_{q \in \mathbf{Q}^r} \zeta_j^{r,q} \cdot g^{r,q} \qquad \forall j \in \mathbf{S} \qquad (2.10)$$

$$\sum_{p \in \mathbf{P}_i} f_i^p = w_i \qquad \forall i \in \mathbf{S} \qquad (2.11)$$

$$s_j \geq \sum_{p \in \mathbf{P}_i} \delta_{i,j}^p \cdot f_i^p \qquad \forall i \in \mathbf{S}_d \quad \forall j \in \mathbf{S} | i \neq j \qquad (2.12)$$

$$s_j \geq \sum_{p \in \mathbf{P}_i} \delta_{i,j}^p \cdot f_i^p + \sum_{p \in \mathbf{P}_{k_i}} \delta_{k_i,j}^p \cdot f_{k_i}^p \qquad \forall i \in \mathbf{S}_c \quad \forall j \in \mathbf{S} | i \neq j \neq k_i \qquad (2.13)$$

The objective function minimizes the total cost of assigning spare and working capacity for each span failure in the network. Constraints (2.9). (2.10), (2.11) are the same as that for the original JCA formulation and ensure the proper working demand routing, working flow placement, and restoration routing, respectively. Constraint (2.12) ensures sufficient amount of spare capacity on any surviving span *j* to accommodate all restoration flow routed over it for failure of any direct span *i*. Likewise, the constraint set in equation (2.13) guarantees that there is sufficient amount of spare capacity on any span *j* to carry all restoration flows placed over them for the dual-failure of any chain span *i* as well its associated logical bypass span *k(i)*. Furthermore, the eligible set of working routes $\mathbf{Q}^r$ as well as eligible restoration routes $\mathbf{P}_i$ are redefined within the augmented topology to include the chain bypass spans. In the same way, the set of restoration routes $\mathbf{P}_i$ are structured to perform the logical dual-failure combinations that now arise when a chain span fails. For the other spans in the original topology that are not part of a chain span, that is direct spans, no special consideration is made as everything for them remain unchanged.

2.4 MESH NETWORK AVAILABILITY ANALYSIS

In this section, we will introduce some network availability concepts related to this thesis. We will highlight general existing methods to perform the availability analysis in span-restorable mesh networks as well as some basic ideas related to it. However, it is important to note that the basic ideas introduced in this section will partially capture, through mathematical means, the concepts that exists inside the availability analysis in survivable communication networks. For a more extensive view of the availability analysis in mesh-restorable transport networks the reader can refer to [22].

2.4.1 INTRODUCTION

Availability is generally defined as the probability of finding a specific device or system in a working state at any time *t*. Essentially, availability differs from reliability, which is related to the likelihood of a device or a system to be in a working state for a certain time *t* without any service-affecting failure occurring. To put it differently, reliability can be though as a "mission-oriented" probabilistic measure where the mission of the system is to achieve certain time *t* in a working condition. Availability, on the other hand, is more related to a *steady state* where it is required or expected that this system has stayed in the operating state from time zero. In this way, in large period of time *t*, the availability reaches a stability or a steady state as a difference with reliability that usually decreases with time [7].

In communication networks, the *service level agreement* (SLA) encompass the primary and main source of guaranteeing service availability between a customer and the network operator [46]. In the SLA, both parties define a quantitatively performance requirements of a network connection as well as penalties that the network operator will have to suffer in case of any of these guarantees are not met. These requirements might differ from different network measurements as for example bandwidth, security, jitter, latency, and outage time. The latter, outage time, can severely harm not only customers businesses but also the network operators. Surely, from a customer's perspective, when it comes to a mission-critical service such as performance of a data center in the banking system, any downtime due to a network outage can cost millions in profits. On the other hand, from a network operator's side, provisioning spare capacity in order to achieve 100% service protection against every single combination of span failures as, for example, not only against single span failure scenarios but also in case of dual span failures situations, can be quite expensive. This cost of course is going to be attained by the chosen network survivability scheme. Network survivability is thus an inherent attribute of a network design problem and it is strongly linked to the necessity of meeting availability requirements, as cost effective as possible, of network operators as well as network users.

2.4.2 AVAILABILITY CALCULATIONS IN MESH NETWORKS

The most widely know equation to calculate the availability for a repairable system is the (2.14), where MTTF is the *mean time to failure*, and MTTR corresponds to the *mean time to repair*. In similar manner, the probabilistic complement of the availability is the unavailability (2.15), which simplifies numerical assumptions for the availability analysis in communication networks [7]. As it is well validated, in the telecommunication industry the MTTR is much smaller than the MTTF. In order to give a better explanation of this matter, consider [48] where the data shown that for 100 miles of optical cable, the component of highest failure rate, for which MTTF = 19,000 hours with a MTTR = 12 hour. Following this, a simplified form of network unavailability (2.16) was presented in [48], where A_i is the availability of the i^{th} element of a set of n elements in series and the U_i is the unavailability.

$$A = \frac{MTTF}{MTTF + MTTR} \tag{2.14}$$

$$U = 1 - A \tag{2.15}$$

$$\prod_{i=1}^{n} A_i \approx 1 - \sum_{i=1}^{n} U_i \tag{2.16}$$

In communication networks, a single-failure restorability (R_1) is defined as the average fraction of working capacity units that can be restored by a network survivable mechanism within a predefined amount of spare capacity placed on it. This of course refers to a unique set of single span failure situations where the amount of spare capacity is predefined to withstand any of these scenarios. Having stated that, some research has been done to introduce these availability calculations in mesh networks [26]. Equation (2.17) calculates the availability in a mesh network with no restoration mechanism over which a path p is provisioned over S spans. In this equation, $U_{link}^{p}(i)$ corresponds to the physical unavailability of the i^{th} link in the path. This assumption provides the definition of *link equivalent unavailability*, which is probability of finding any link not only in a failed state but also in a non-restored state by any network restoration mechanism $U_{link}^{*}(i)$ [26]. Equation (2.18) states this distinction, where a method of calculating the equivalent unavailability is presented in span-restorable network based on the capacity in the network and the particulars of the restoration mechanism. In this equation, U_S represents the physical span unavailability and S is the number of spans in the network. Given these points, we can argue that in a network designed for 100% single failure restorability ($R_1 = 1$), dual span failures have a direct influence on the network service availability because these situations dominate network outages in single-failure survivable networks. Thus, dual span failure restorability R_2 is considered. Ultimately, we will use the term of dual-failure availability or dual-failure unavailability interchangeably when referring to the availability from dual-failure situations.

$$A_{path}(d) \cong 1 - \sum_{i=1}^{S} U_{link}^{p}(i) \tag{2.17}$$

$$U_{link}^{*} = U_{s}^{2}(S-1)(1-R_2) \tag{2.18}$$

2.4.3 DUAL FAILURE NON-RESTORED WORKING CAPACITY

A dual span failure is typically denoted (*i, j*), where *i* and *j* are the two spans involved in the failure scenario. In networks designed for 100% single-failure restorability, that is $R_1(i) = 1$, there may or may not be resulting available resources that can be used for other purposes. This is mainly because in network spare capacity minimization problems some spans require more or less spare capacity than others. Although these resulting available resources cannot afford 100% dual failure restorability, that is $R_2(i, j) = 1$, usually in these cases a number of working capacity units can be restored. Therefore, it is our interest to know what is the number of *non-restored working capacities* that cannot be restored under each dual failure situation (*i, j*). This is usually denoted by $N(i,j)$. Hence, the following equation (2.19) introduces the sum of all dual failure non-restored working capacities in a network with a set of span S [7].

$$N_2 = \sum_{(i,j)\,\in\,\mathbf{S}\,|\,i\neq j} N(i,j) \tag{2.19}$$

2.4.4 DUAL FAILURE RESTORABILITY

As we previously mentioned, the restorability of a network is the average amount of failed working capacity units w_i that can be restored within the spare capacity placed on it by a specified restoration mechanism. Under this circumstance, we can define the *dual failure restorability* as the average fraction of failed working capacity units that can be restored within the spare capacity placed on it under all dual-failure span situations [7]. That is, if a network can achieve $R_2 = 1$ it will withstand any dual-failure situation that might arise on it. This can be further calculated by Equation (2.20) [49].

$$R_2(i,j) = 1 - \frac{N(i,j)}{w_i + w_j} \tag{2.20}$$

In the above equation, w_i and w_j are the amounts of working capacity units placed on span i and j, respectively. As mentioned, $N(i,j)$ corresponds to the number of non-restored working capacities that cannot be restore under each dual failure situation (i, j). Nevertheless, a preferred definition for $R_2(i,j)$ will be a weighted average of the total working capacity to be restored in each combination (i, j). Therefore Equation (2.21) aims to calculate the dual span failure weighted average restorability for networks designed with a minimum amount of spare channel capacity sufficient to yield $R_1 = 1$ [7]. Here, $|S|$ is the number of spans in the network.

$$R_2 = 1 - \left(\sum_{(i,j)\, \in S^2 |\, i \neq j} N(i,j) \,/\, 2 \cdot (|S| - 1) \cdot \sum_{i \in S} w_i \right) \tag{2.19}$$

2.4.5 DUAL SPAN FAILURE TYPES

In mesh-based communication networks there are four logical categories that describes dual failure scenarios [22].

I. Dual-failures with no interaction between their restoration routes.

II. Dual-failures with some interaction between their restoration routes. This interaction may content one of their restoration routes.

III. Dual-failures where the second failure *j* harms or damage one or more restoration routes of the first failure *i*.

IV. Dual-failures where a degree-2 node is affected.

Figures 2.13 (a) to (d) portrays these four dual failures categories. As can be seen, Figure 2.13(a) shows the case with no interaction between the failed span restoration routes. In this case, both failures are fully restorable, and therefore, the dual failure restorability will be 100%. Figure 2.13(b) illustrates the case with a partial interaction between restorations routes may arise. In this case, the fully restoration will depend on the available spare capacity between the two failures, that is if the number of working units placed on span *j* is greater than the number of working units placed on span *i*, the second span will not be fully restored. Figure 2.13(b) shows the case where the second span failure *j* entirely damages the restoration routes of the first failed span *i*. Finally, Figure 2.13(c) portrays the case where a degree-2 node is isolated due to the cuts of its adjacent spans yielding an unfeasible situation. In this case, the dual span failure restorability will be zero.

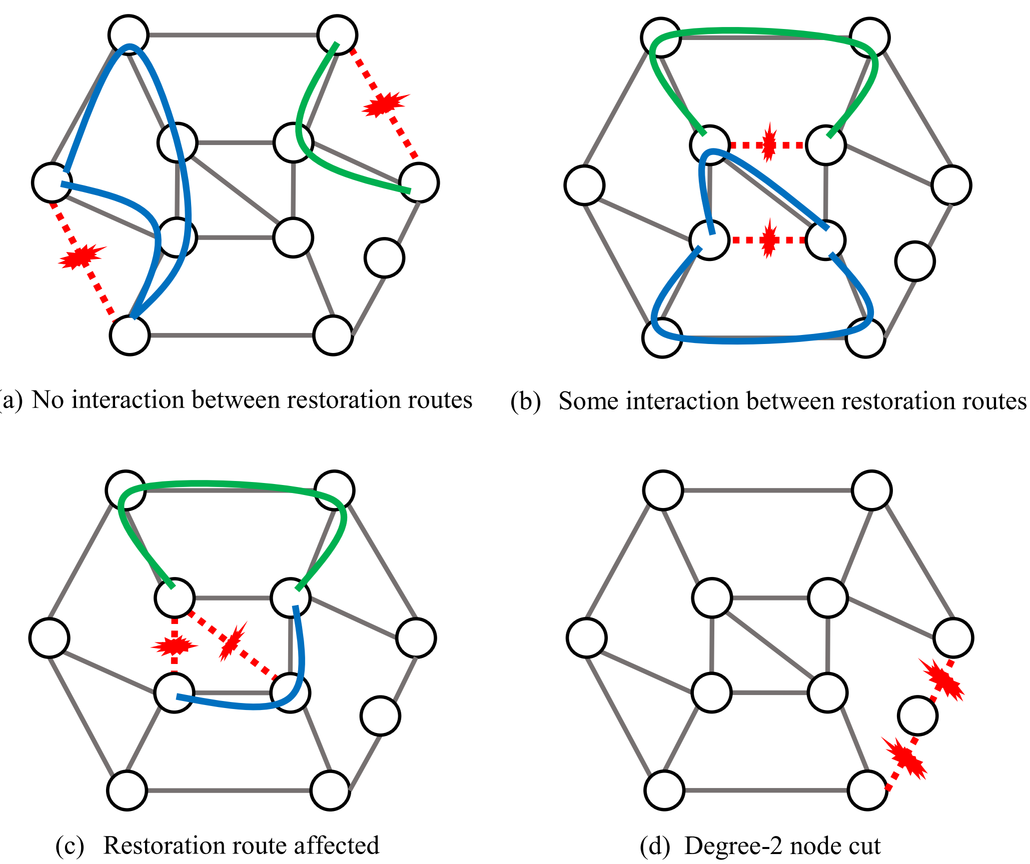


Figure 2.13 – Different types of dual-failures scenarios

CHAPTER 3 RESEARCH GOALS AND EXPERIMENTAL SET-UP

3.1 MOTIVATION AND GOALS

Generally, network operators design their transport network to be 100% restorable to any single span failure. That is, the network has sufficient amount and distribution of spare capacity so that any single span cut, or failure can be withstood without service outage. Perhaps this measure of real-time restoration or protection mechanism is enough to ensure a high network availability in some communication networks. However, dual-failure scenarios are becoming a reality threat capable of harm societies and businesses [45]. These situations motivate the analysis of the effects of dual failures on single failure restorable designs, which brings us to the central question of this thesis: "How well does a meta-mesh span-restorable network, designed for 100% restorability to single failures, actually stand up to dual span failure scenarios?". The purpose of this study is therefore to investigate and analyse how this network design behaves during dual-failure situations. To fulfill this objective, we develop an integer linear programming (ILP) model for the calculation of the meta-mesh dual-failure minimum capacity (MM-DFMC) problem as well as the calculation of the meta-mesh dual-failure maximum restorability (MM-DFMR) problem.

Another important part of this thesis is devoted to improvement of the previous meta-mesh design by offering a new insight in their topology model. This new insight is capable of offering a greater capacity efficiency in some experimental sparse network graphs previously studied in [5]. In summary, the goals of this thesis can be briefly described as following:

a) Introduce a new meta-mesh design insight capable of reaching greater capacity efficiency in some meta-mesh network topologies.
b) Develop an ILP model for the calculation of the meta-mesh dual-failure minimum capacity (MM-DFMC).
c) Develop an ILP formulation for the calculation of the meta-mesh dual-failure maximum restorability (MM-DFMR).

3.2 NETWORK TOPOLOGY MODELS

The formulation methods developed and discussed in this research thesis are implemented in a set of 124 test network topologies, which are divided into six groups or families of related networks as in [5], [23] and [36]. Each network family is created from an initial network or a *master network* with an average nodal degree of 4.0. From this master network, spans are removed one at a time in a random manner, so the network average nodal degree $\bar{d}$ is decreasing. This process is repeated until the removal of any span violate the bi-connectivity of a network node that is a network with a $\bar{d}$ equal to 2.0. Figure 3.1 illustrates an example of this procedure. As can be seen, Figure 3.1(a) start with a master network with average nodal degree equal to 4.0 and then by removing one span at a time the $\bar{d}$ is decreasing. Note that these networks topologies were created in a network research laboratory in [5] and even though they do not represent real network graphs, they have strong characteristics and qualities of real transport network topologies. It is important to emphasize that the demand matrix of each node remains constant for each set of networks.

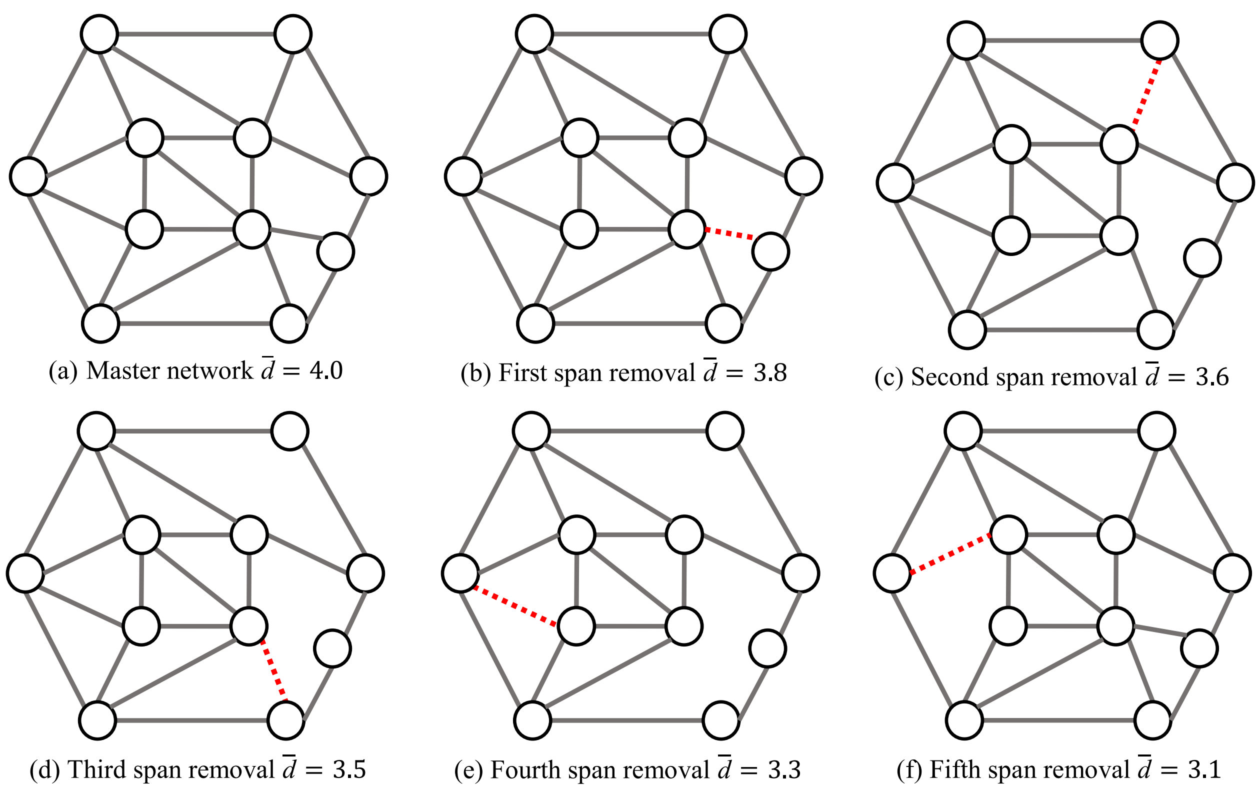


Figure 3.1 - An example of network topology creation

Each of the following networks were used for the implementation of the improved meta-mesh model as well as for the implementation of the two availability enhance models that will be presented in Chapter 4 and in Chapter 5, respectively. Figure 3.2 illustrates all six master networks (15n30s1, 20n40s1, 25n50s1, 30n60s1, 35n70s1, and 40n80s1). The remainder of these network families are fully presented in Appendix A.

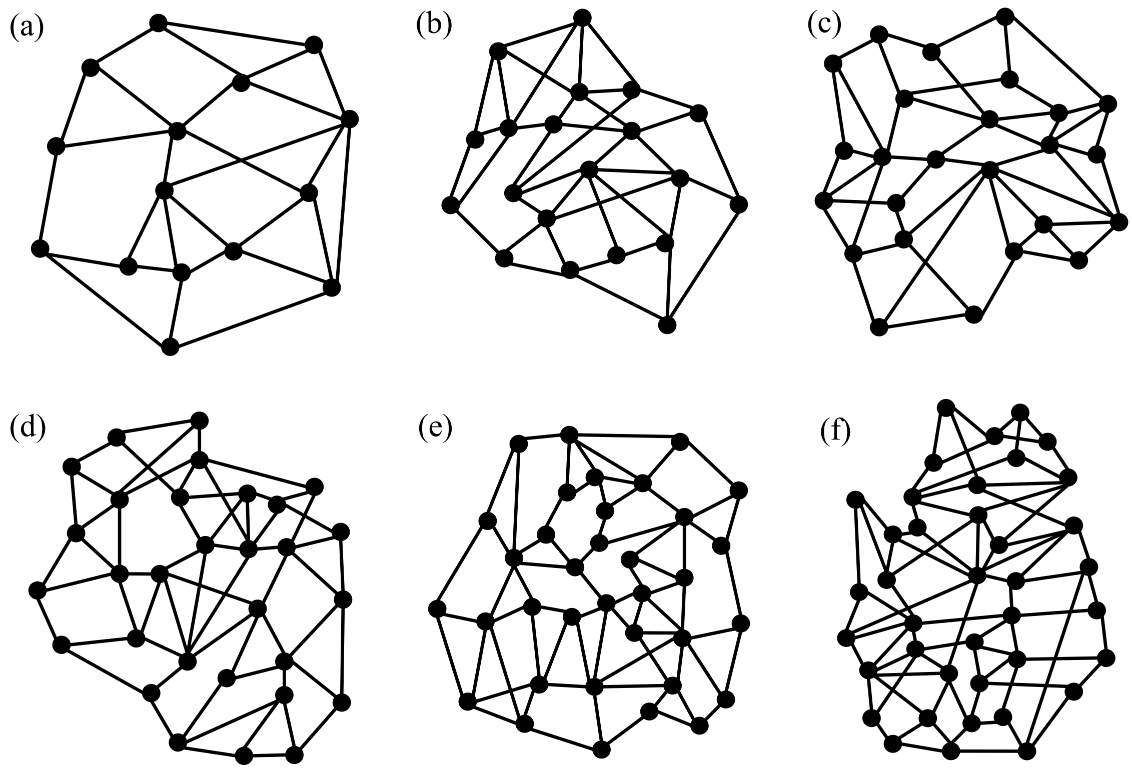


Figure 3.2 - Masters Network topologies (a) 15n30s1, (b) 20n40s1, (c) 25n50s1, (d) 30n60s1, (e) 35n70s1, (f) 40n80s1 [5]

3.3 DEMAND MODELS

The term *demand* was defined in [7] as “the unit of transmission and routing capacity used to serve any aggregations of traffic flow from the service layers”. That is, it is the working capacity of aggregated traffic coming from different sources that needs to be transported between an origin-destination (O-D) pair of nodes of the network topology. Following common practices in network survivability, the work developed in this thesis will assume that one demand unit consumes one working wavelength on each span traversed on the route of the demand between the O-D nodes. In real transport networks, the total demand for wavelengths between each O-D pair is a result of a great number of traffic types requiring service and can follow a variety of models. This thesis, follows the uniform model as the prior meta-mesh work in [23], where each origin-destination pair is assigned a demand intensity from a discrete uniform random distribution. That is, every O-D pair exchanges some uniformly randomly assigned number of the demand units from 1 to 10.

3.4 ARC-PATH ILP FORMULATIONS

Herzberg and Bye in [27], [61], proposed the *arc-path* LP formulation for the discussed spare capacity assignment (SCA) problem in a span-restorable network design, and, unless otherwise stated, this thesis will follow this arc-path type of formulation. In this formulation, the network graph is first pre-processed to find all the different eligible restoration routes for each span failure scenario. In the case of the JCA model design, an explicit enumeration of a set of eligible working routes is a pre-processed requirement as well. Note that enumeration of all distinct working and/or restoration routes is exponential in complexity with the network size. In addition, in the arc-path method, restoration and/or working routes can be obtained under direct engineering control to limit properties such as length, hop count, signal loss, etc., for each span failure situation. A proper description of this technique of generating eligible route sets is discussed herein.

3.4.1 GENERATING ELIGIBLE ROUTE SETS

As mentioned, the arc-path formulation type requires a preprocessing of the network graph to obtain the sets of distinct eligible restoration and/or working routes. Ideally, this procedure should be accomplished under an engineering control limit such as length, hop limit, etc. This is because for moderately sized network graphs such numeration of distinct eligible route sets can be very large. Therefore, there are two general approaches for generating route sets [7]:

- Route sets can be generated up to a compromise hop limit, *H*, and combined with a set of *k*-successively shortest paths found without any hop limit. Generally, this hop limit is up to 6 hops for optimal results.
- Route sets can be generated by setting a minimum amount of distinct eligible routes with a specified minimum hop limit, *H*. In addition, this hop limit can be increased until the specified target number of distinct routes is reached.

Unless otherwise stated, results in this thesis are based on the second type of enumeration for a minimum target number of eligible routes in all cases.

3.5 COMPUTATIONAL ASPECTS

All the network survivability models developed in this thesis were implemented using AMPL modeling language [59] and solved using Gurobi 6.5.0 [60] as a solver on a 12-core ACPI multiprocessor X64-based PC with Intel Xeon® CPU E5-2430 running at 2.2 GHz with 96 GB RAM. All solutions have been run with the default *mipgap* of 0.0001, meaning they are ensured to be within 0.01% of optimality.

CHAPTER 4 IMPROVED META-MESH NETWORK DESIGN

4.1 INTRODUCTION

The concept of the span-restorable meta-mesh design was mentioned in Chapter 2 where it was pointed out that this method enhanced the capacity efficiency of span restoration in a sparse network topology by improving the way in which chain subnetworks are restored [50]. The meta-mesh is not a new restoration or protection model as it employed span restoration as a network survivability mechanism. With this in mind, one important operational aspect of span restoration is that the restoration path-set is not limited to use of one replacement path to transport all failed working capacity of a specific span [7]. Rather, it can use different routes up to a specified hop or distance limit to route all failed working capacity. That is, span restoration is capable of employing several routes dividing the amount of working capacity which failed in a specific span and in this way efficiently restored. This allows an efficient sharing of spare capacity among different failure scenarios in mesh-based networks. As discussed in Chapter 3, the number of eligible restoration routes and its operational concept encompasses a limitation that have to be imposed allowing only a certain number of hops or physical length limit, H [7], [27]. In this way, by imposing this limitation, route sets would not consider choosing extremely long paths unless it is specifically required. For instance, Figure 4.1 portrays this misunderstood concept where Figure 4.1(b) shows the usual performance of span restoration. As can be seen in, instead of using a two-hop path as on Figure 4.1(a), span restoration employs multi-hop paths to overcome the span failure as on Figure 4.1(b). Similarly, the hop limit in Figure 4.1(b) is limited to $H = 3$ so that any restoration route cannot take longer paths and only those paths with $H < 3$ are considered. As an example, the path between nodes C-A-B-E-D with an $H = 4$. In addition, a minimum amount of the three shortest restoration routes can be pre-processed in a way that only those ones would be considered for restoration. The main point is that Figure 4.1(a) does not represent the way that span restoration solely works.

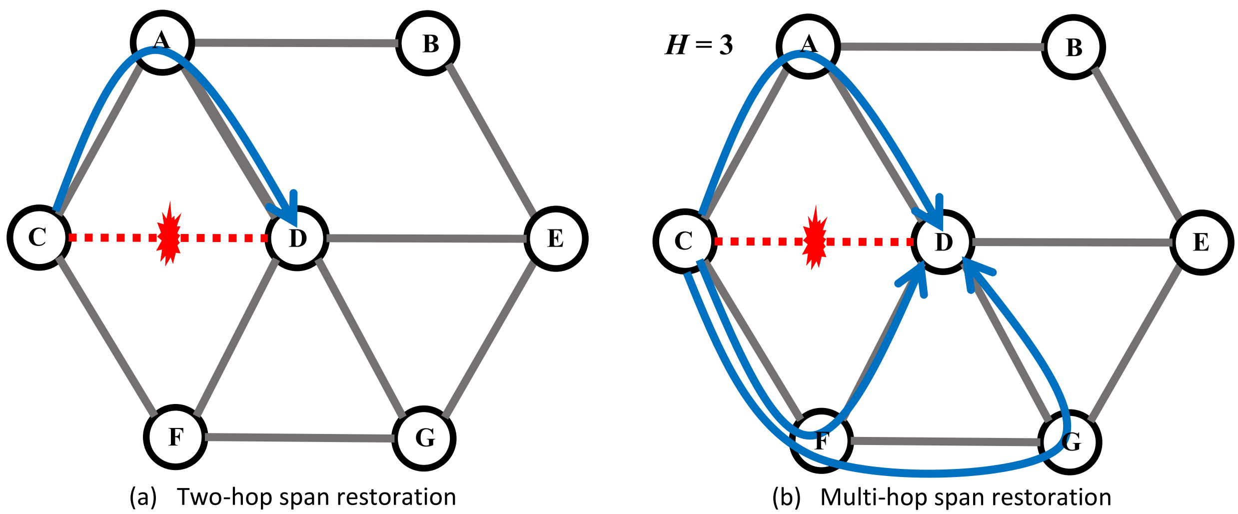


Figure 4.1 – Operational concept of span restoration

As previously stated in Chapter 3, the work developed in this thesis is based on a set of 124 network graphs divided into six network families. Test networks from the previous work were created by a systematic reduction in the network average nodal degree from a highly connected master graph. In this manner, different network graphs were originated with a random removal of one span at a time from the main or master network until it was no longer violating the bi-connectivity of a node [5]. This provided a reasonable continuous variation in the network average nodal degree $\overline{d}$. Furthermore, in order to implement the meta-mesh model, the topology file was augmented to include a logical bypass span in parallel with each chain [5], [23]. This is because this logical bypass span would serve as an express route for all the working capacity that fully transit the chain and it is destined to a node outside of the chain. The benefit of having this logical bypass span is that only those degree-3 or higher nodes (e.g., anchor nodes) would require a full OXS functionality. In contrast, the chain nodes can be handled using simpler OADM equipment (e.g., straight fibre splices or glass-through) [5].

Having stated these important points, an interesting observation was made regarding the existence of chains subnetworks that were not bypassed with a logical bypass span in the majority of the test networks experiments in the prior work [5]. Without these bypass spans, every span inside these chains will be treated as a direct span and no distinction between their local and express flows will be done. Without this, there is no difference between a span-restorable design and its augmented meta-mesh design model. One such scenario is illustrated in Figure 4.2 where the dashed lines represent the not yet bypassed chain. Note that the majority of these not bypassed spans situations are related to a chain containing one node and two spans taking a somehow triangular shape, where the cost of transiting the chain in its entirety is greater than taking the single-span between both anchor nodes. We will refer to these chains as *meta-mesh low priority chains*. This brings us to our current interest in investigating how adding new logical bypass spans in the already studied network topologies would affect its total capacity efficiency. The aim of this chapter will thus to further investigate the logical bypass span in meta-mesh networks and show how well suited and advantageous it can be for network capacity efficiency.

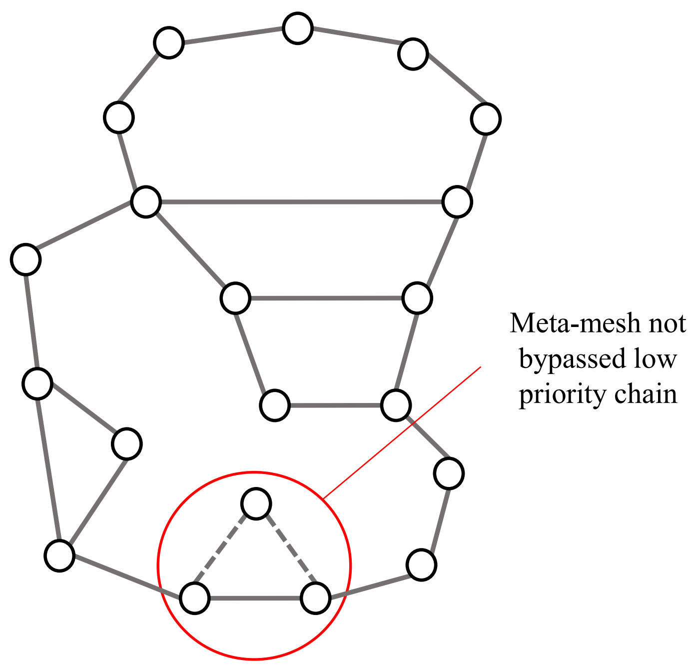


Figure 4.2 – Illustrating a meta-mesh low priority chain not bypassed

4.2 IMPROVED META-MESH DESIGN MODEL

As it is briefly mentioned, in the majority of the meta-mesh test networks experimented in the prior work [5], [23], not all chain subnetworks or series of degree-2 nodes were bypassed with a logical bypass span. Without the addition of these bypass spans to these networks, no distinction between its local and express flow can be done and therefore no saving in total and spare capacity is achieved. In fact, any restoration due to a failure in any spans inside the chain would be treated as equivalent to span restoration. In addition, the meta-mesh design interpreted these spans inside these not bypassed chains as direct spans, or spans whose end-nodes are both of degree-3 or higher, when in reality there are just spans of degree-2 nodes. The majority of these not bypassed spans situations were related to a chain containing one node and two spans forming a somehow triangular form where the cost of travelling the chain is greater than taking the path between both degree-3 nodes. The fundamental idea behind this is that, regardless of the bypass span existence, no working or restoration route would be considered for the use of that single-node or two-spans chain because the cost of wavelength per kilometer results more expensive that taking the single-span option. In other words, the span restoration design under the total cost minimization would not use the costly path option, which is the single-node chain, to allocate spare capacity to restore any span failure inside the network. This behaviour is illustrated in Figure 4.3, where the cost of travelling the single-node chain containing span S2 and span S3 is higher than using span S1. Therefore, the solver under total cost minimization would never yield a solution where working or restoration flow is placed in span S2 and span S3 rather than in span S1. Hence, since the conventional meta-mesh model produces its benefits when a significant amount of express flow is travelling through chains, a zero benefit in spare capacity reduction would be achieved. The only scenario, of course, will be in case of span S1 failure, where the model allocates the spare capacity dedicated to restore span S1 on this chain subnetwork, or for routing demands between node A and node B in case of working flow. Figure 4.3 is an illustrative example only, but it is representative of a real situation presented in some test network of [5].

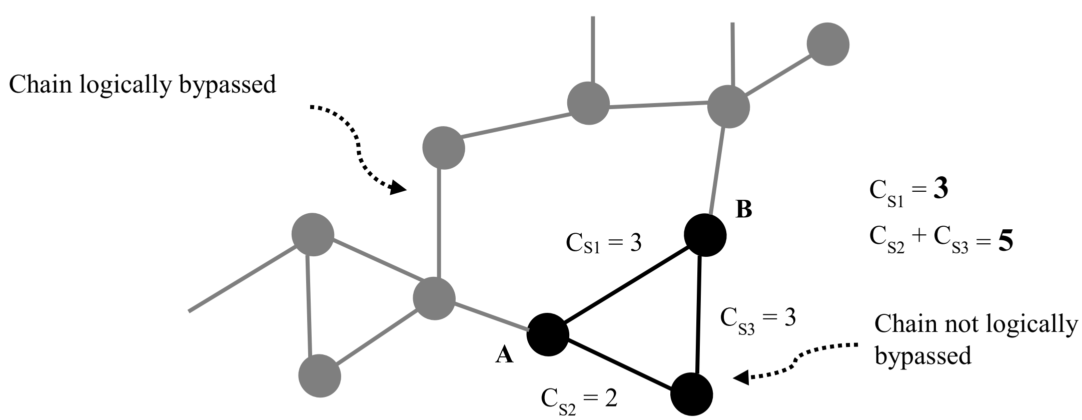


Figure 4.3 – Bypass span problem description

To further explain this, let us start by considering an ordinary span-restorable meta-mesh response to a single failure situation in a sparse network topology. Figure 4.4(a) illustrates a pair of nodes A-B exchanging demands by a two-path option (this can be via multiple paths but for simplicity we are only going to considered two). In this example, spans S2 and S3 correspond to a not bypassed low priority chain. If span S1 fails, a normal response under a meta-mesh design for the path passing through span S1 would be the rerouting of all working capacity between the immediate end nodes as portrayed in Figure 4.4(b). On the other hand, an intra-chain failure, as for instance in span S2, would require an allocation of sufficient amount of spare capacity in S3 to loopback all failed local and express working flows as illustrated in Figure 4.4(c). A creation of a logical bypass span in this chain would let those demands, that physically traverse the chain, fail all the way back to the anchor nodes and no spare capacity is therefore needed within the chain for this express flow. This is portrayed in Figure 4.4(d).

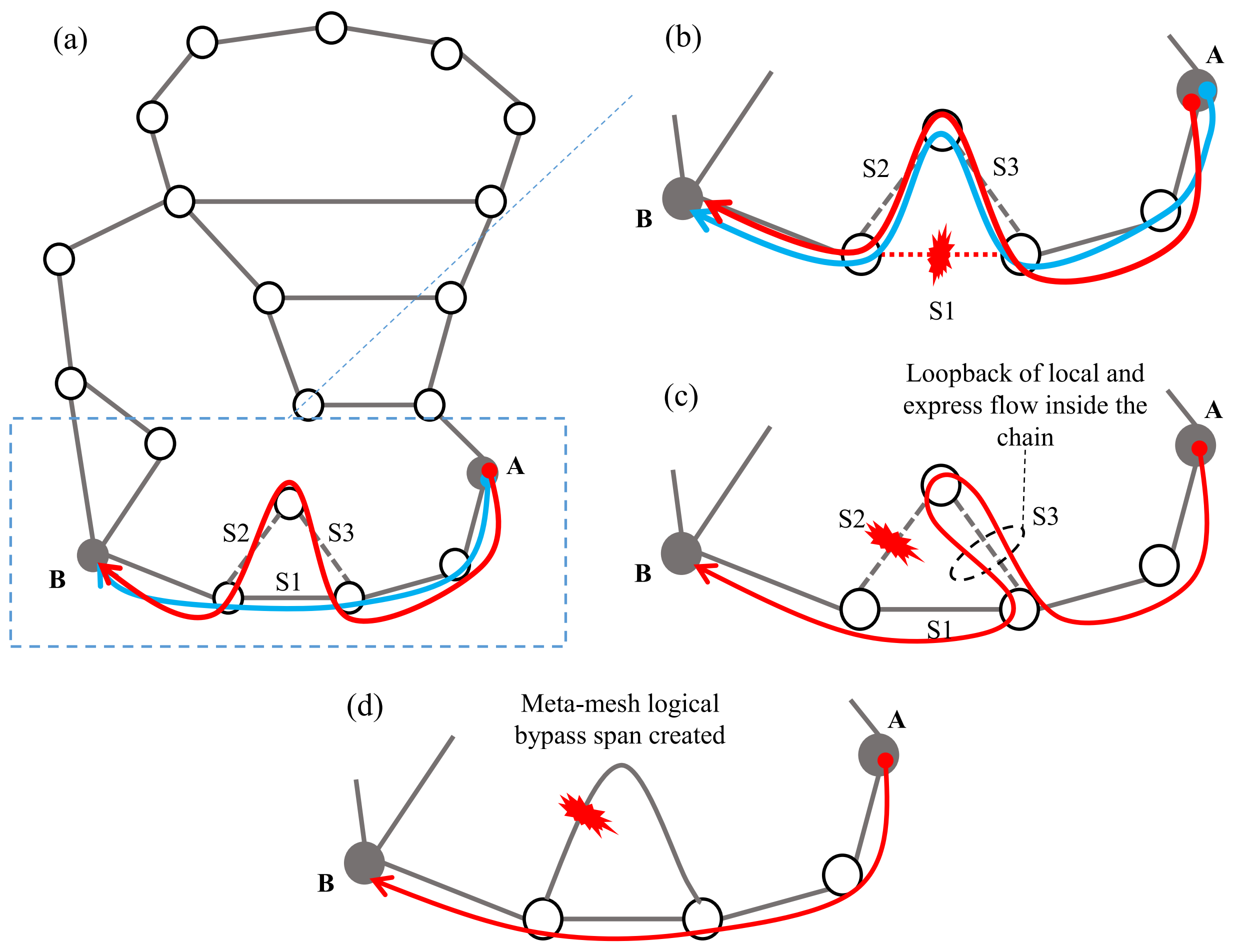


Figure 4.4 – Illustrating meta-mesh advantage in chain subnetworks

As stated, span-restorable designs use a diverse path-set to efficiently place spare capacity for restoration rather than using a single-path or two-hop path between its immediate end nodes. Indeed, span restoration rerouting occurs at a granularity level, using multiple-hop paths for restoration capacity efficiency [7]. This allows an efficient management in spare capacity allocation. With this in mind, the span-restorable meta-mesh design might use these costly single-node chains as well as the less costly single-span option to place spare capacity destined to restore spans failure scenarios that arises in the network graph. Furthermore, it must be remembered that the pre-processing number of eligible working or restoration routes are based on a hop limit (H) such that only those routes composed of H or fewer spans are considered as eligible routes. It makes sense in principle that there may be ways of routing the working demands as well as the restoration flow through these chains that somehow make the related survivability design less costly. Although the majority of the flow that the solver would place in these chains is local, that is intra-chain working capacity, it might have a room for some express flow as well. Having stressed these points, we introduce the *improved meta-mesh design* (IMM), which is not a new restoration method nor a new formulation design, rather it is a topology that arises when all chain subnetworks inside the meta-mesh network are properly bypassed allowing the full capacity efficiency of the span-restorable meta-mesh design. The general idea in this design is to add a new logical bypass span in those chains that were not logically bypassed in the experimented test networks and analyse the outcome of it. To achieve this, the sets of eligible restoration routes for each span failure were regenerated to include new logical bypass spans associated with chain subnetworks that had not been bypassed. This allows the solver to exploit any routing possibility to ensure spare capacity reduction in this design model.

4.2.1 IMPROVING THE TOPOLOGY BY ADDING NEW LOGICAL BYPASS SPANS

We now proceed to implement a new logical bypass span in those chains that were not logically bypassed inside the test networks belonging to different network families. Out of the 124 test networks from different network families, 105 of them were logically bypassed with at least one bypass span. Figure 4.5(a) and (b) encompass the most common situation founded in several networks where a chain was not bypassed and the most likely scenario where a reduction in spare capacity cost can be achieve. The reason behind this is that in a chain as Figure 4.5(c) any solver under the total minimization objective would prefer to accommodate restoration flow in this chain rather than in the single-span connecting its anchor nodes. Therefore, without any express flow routed over this chain the meta-mesh design model will act in the same manner as span restoration. The table in appendix B provides a summary of the total of logical bypass span added to each network inside each network family. The table gives a number of single-node chains, double-node chains, triple-node chains, and multiple-node chains that were not bypassed in each test network. As can be seen, the majority of the situations encountered was as the one in Figure 4.5(a).

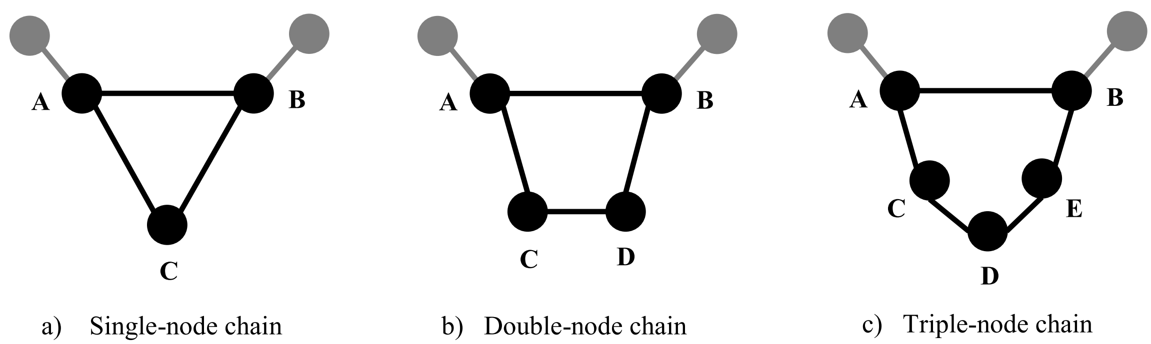


Figure 4.5 – Topological designs of not bypassed chains

4.3 EXPERIMENTAL STUDY METHOD

The ILP model implemented in this section corresponds to the same meta-mesh model that was fully described in Section 2.2.4. Prior work on the meta-mesh approach [5] studied 124 test networks that we will further employ for our following experiments. In this way, the results obtained in the original meta-mesh design will serve as a benchmark for our following experiment. As previously stated in Chapter 3, each ILP formulation model was implemented as an AMPL model [59] and solved using Gurobi 6.5.0 solver [60] on a 12-core ACPI multiprocessor X64-based PC with Intel Xeon® CPU E5-2430 running at 2.2 GHz with 96 GB RAM. The pre-processing stage of all eligible working and restoration routes and other input parameters was done on an Intel® Core™ i7 notebook with 8.0 GB of RAM, running Windows 7. Ultimately, all spare and working capacity allocations were integer.

The meta-mesh ILP model herein is based on the arc-path approach, which requires us to enumerate all eligible working and restoration routes in advance. Our following results are based on a minimum of 5 eligible route choices for routing working demand as well as a minimum of 15 distinct routes for span restoration. We increased the number of restoration routes from 10 to 15 in order to allow more sharing efficiency of permitted re-routing as in [27] as well as encourage the use of these new chains by the solver.

4.3.1 Results and Discussions

Out of the 105 logically bypassed networks, only 28 of them displayed a reduction in spare capacity. Figure 4.5 and Figure 4.6 provide a summary plot of the results in terms of total spare capacity percentage reduction relative to the original meta-mesh design versus the network average nodal degree in each test case. Each of the figures portray a scatter-plot data for independent test cases of a combination of three network families. Correspondingly, Figure 4.5 provides data for the 15n30s1, 20n40s1, and 25n50s1 network families. In the same manner, Figure 4.6 illustrates data for the 30n60s1, 35n70s1, and 40n80s1 network families.

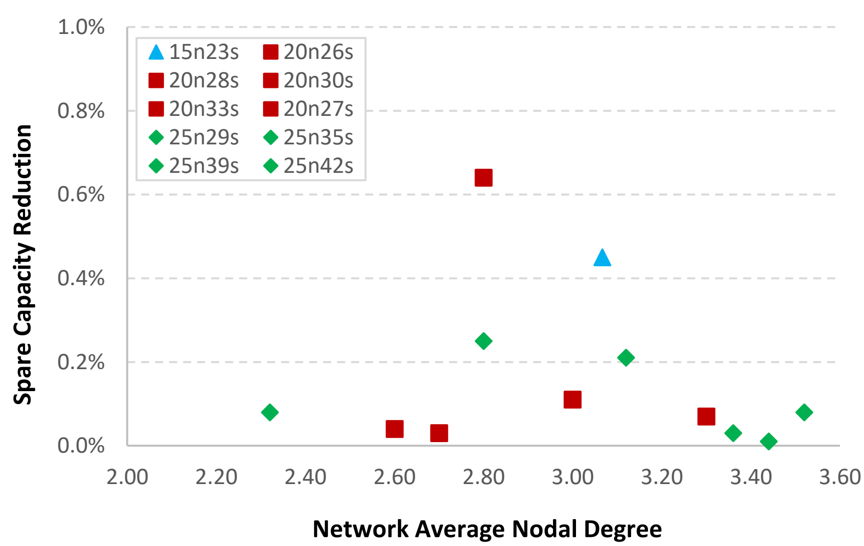


Figure 4.5 – Total spare capacity saving versus network average nodal degree in the test networks that presented spare capacity saving inside the 15n30s1, 20n40s1, and 25n50s1 network families

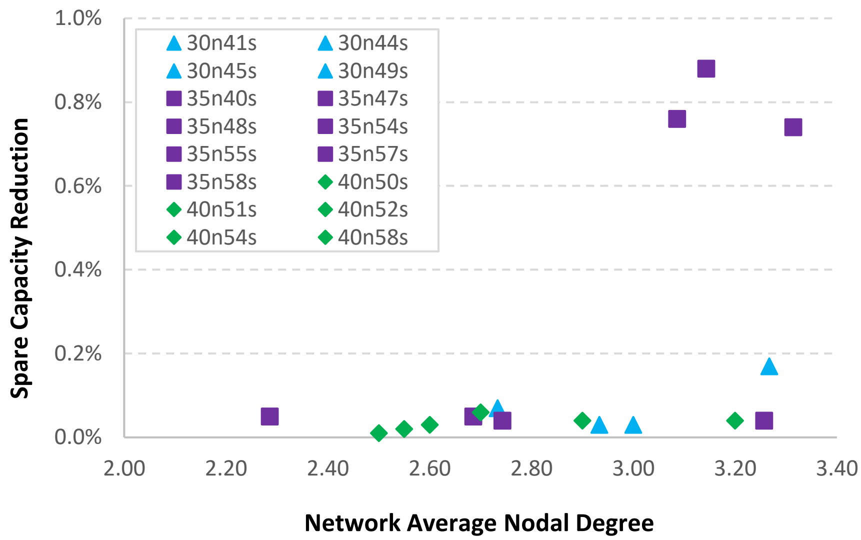


Figure 4.6 – Total spare capacity saving versus network average nodal degree in the test networks that presented spare capacity saving inside the 30n60s1, 35n70s1, and 40n80s1 network families

First, and not surprisingly, both figures showed that our improved meta-mesh model do not provide a significant reduction in spare capacity as the total average of the 28 cases is very low. In this case, some network families experienced greater spare capacity reduction than others, which is going to not only depend on the number of logical bypass spans added to the network but also on the network topology itself. That is, if the restoration model, in this case span restoration, uses or not the new bypassed chain to route lightpaths instead of the single-span option. For one network, 35n70s1-55s, the reduction in spare capacity was 0.88%, which corresponds to the maximum reduction in spare capacity experienced amongst the 28 networks. This test network contained four single-node chains that were not bypassed as well as one double-node chain that was not bypassed as it is conceptually illustrated in Figure 4.7. In addition, two test networks from the same network family, the 35n70s1-54s and the 35n70s1-58s, experienced reduction in total spare capacity needed for full single-failure restorability of 0.76% and 0.74% respectively. Both test networks contained a high number of chains that require the allocation of a new logical bypass span. In the 35n70s1-54s test network, four single-node chains as well as one double-node chain required a logical bypass span. For the case of the 35n70s1-58s, only three single-node chains required a logical bypass span. Another high improvement in spare capacity was for 0.64% in the 20n40s1-28s test network. In this case, four single-node chains were bypassed. As mentioned in Section 4.2.1, the single-node chain is the most likely scenario where a spare capacity reduction can be achieved. This is because a solver under total spare cost minimization would never choose the multiple-node chain to route working demands or restoration flow instead of the less costly single-path option. Indeed, experimental results demonstrated this assumption where 77 logically bypassed test networks did not show any reduction in total or spare capacity.

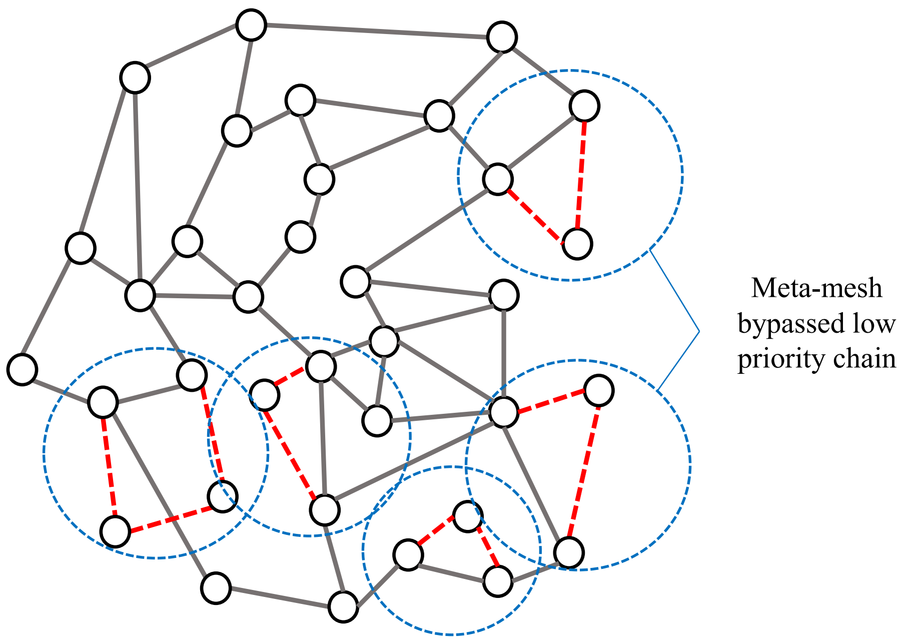


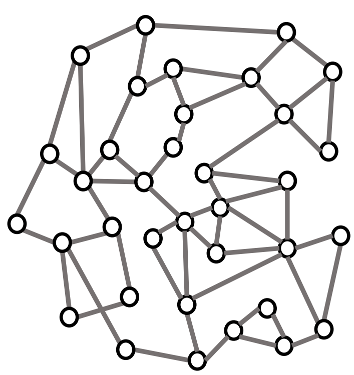
Figure 4.7 – An illustration of the 35-node and 55-span test network

Now, the significance of these results, in particular that these low amounts in spare capacity reduction are achieved by the improved meta-mesh model in those 28 test network scenarios, is questionable. However, in the modern transport network with increasing connection speed and hence with increased traffic, the costs associated with transmitting demands can be very high [55]. Note that the results of any network economic study depend on the topology of the network, the number of demands exchanged between their origin-destination node pairs, and, of course, the cost assumptions used in the study per se. Furthermore, in networks operating in full capacity, that is full potential of the bandwidth of the fiber, the capital cost of the optical layer would be dominated by the cost of the transponders, optical cross-connect (OXCs) devices and optical add/drop multiplexers (OADMs) [55]. Work developed in [55] and in [63] estimated relative costs for equipment used in an 80-wavelenght, 2,500-km optical backbone network with 10-Gb/s line rate that are shown in Table 4.1. These costs are a rough estimation due to the several assumptions that were made during their development [55].

Table 4.1 – Relative costs for an 80-wavelength, 10-Gb/s, 2,500-km optical system [55]

Element	Relative cost
Tunable transponder	1X
Tunable regenerator	1.4X
Bidirectional in-line optical amplifier	4X
Optical terminal	5X
ROADM	14X
Degree-3 ROADM-MD	21X
Degree-4 ROADM-MD	28X
Degree-5 ROADM-MD	35X
OTN grooming switch port	1.5X

However, an exact calculation of the operating cost to run a network, often referred to as OpEx, for these test networks is beyond the scope of this thesis. In this way, to oversimplify this scenario, we will assume an x10$ cost for every lightpath exchanged between each O-D node (regardless the distance and the equipment involved). This is because the aim of this discussion is to capture the idea that even a small reduction in spare capacity associated with single failure restorability can yield significant saving in economic network costs, rather than deriving absolute network costs. As it was mentioned in Chapter 3, the work developed in this thesis used an uniform model to illustrate real transport network total demand for wavelengths between each O-D pair. That is, every origin-destination pair exchanges some uniformly randomly assigned number of demand units from one to ten. Thus, if we look closely at the total spare capacity cost required to ensure full single failure restorability in the 35-node and 55-spans test network, we realize that a total saving of approximate $100,000 in the event of a failure can be achieved. Figure 4.8 shows more details of this example.



- Total spare capacity original meta-mesh = 1165870
- Total spare capacity improved meta-mesh = 1156990
- Estimation of the original meta-mesh cost = $11,655,700.00
- Assumed savings = **$102,570.00**

Figure 4.8 – Details of approximate savings in the 35-node and 55-span test network

What we want to emphasize is that for the high amount of capital expenditures as well as operational expenditures of backbone networks, even a small reduction is spare capacity can yield significance reduction in network economic cost, so it is reasonable to focus on this aspect [55]. In addition, Figure 4.9 provides a histogram of the total capacity cost breakdown needed in the original meta-mesh design and in the improved meta-mesh for the 35-node and 55-span test case, which was the network that experienced the greatest reduction in spare capacity. As it was expected, more working capacity is allocated inside chain spans and, of course, the logical bypass span, which is proportional to the reduction in working capacity into the direct spans (e.g., degree-3 or higher nodes). This is because the model is not longer considering those spans that were not bypassed as direct spans of the network (recall Section 4.2) and it is properly assigning working capacity into chain subnetworks. Ultimately, the relative benefit is also close to zero in those networks that contain low not logically bypassed chains and the conventional meta-mesh design cannot be improved upon.

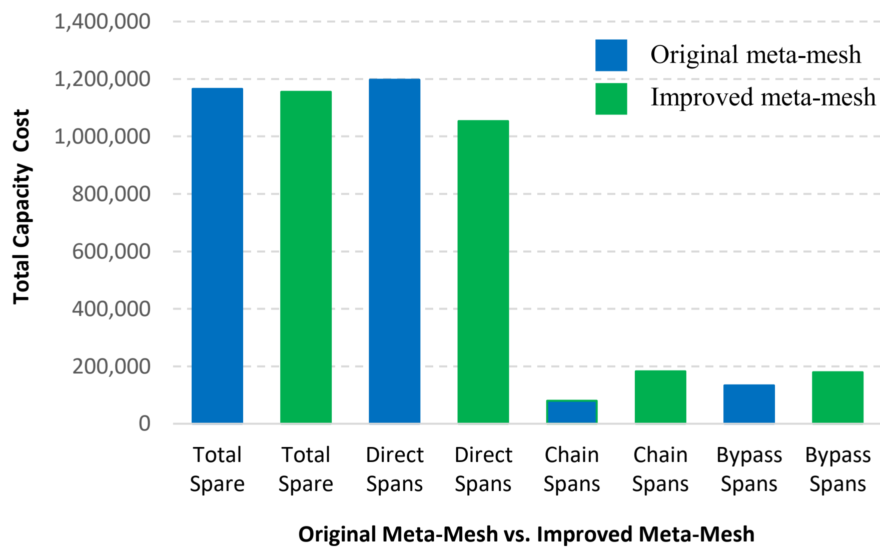


Figure 4.9 – Breakdown of the original meta-mesh design versus the improved meta-mesh design

CHAPTER 5 HIGH AVAILABILITY META-MESH CAPACITY DESIGN[1]

5.1 INTRODUCTION

As stated in previous chapters, most networks operators utilizing survivability mechanism are designed to ensure service continuity (100% restoration) in the event of any single span failure scenario. As a result, single span failures are not anymore a significant contributor to network outages [26]. And although simultaneous dual failures can arise due to unforeseen common causes of failure or shared risk link groups [62], such situations can be avoided via proper design of the network's topology. Work in [26] showed that it is two independent failures overlapping in time, where a second failure occurs before repair of an initial failure is complete, that will be the major contributor of network outage (unavailability). Therefore, the study and analysis of these situations in transport network are of significant interest for network operators. In addition, considering the consequences of these scenarios in a real backbone networks [45], motivated us to pursue this study and understanding of dual-failure restorability in meta-mesh networks designs. Certainly, protection of transport networks against single-failure situations can be a norm for some network operators but protection of these networks against dual-failure situations can be a luxury for some of them [42]. The meta-mesh served as a refined method of using the span restoration in sparse network topologies and allowed to achieve greater capacity efficiency for single span failure restorability [23]. Thus, it is important now to study how this design can be enhanced in order to achieve dual failure restorability.

Prior work on mesh-based span-restorable dual failure availability [26] demonstrated that it is difficult to predict dual failure restorability analytically. This is because the dual span failure restorability of each span failure pair depends on the specifics spans involved in the failure, the failure sequence, the graph topology, the exact working and spare capacities, and the assumed

[1] This chapter is adapted from the conference paper: A. Castillo, J. Doucette, "Dual-Failure Availability Analysis of Meta-Mesh Networks," *International Workshop on Resilient Networks Design and Modeling* (RNDM) 2018, to be submitted, May 2018.

restoration process. Thus, three technical models were developed which analyzed how different levels of adaptability affected the network dual failure restorability. These models presented in [26] were referred to as *static restoration preplans*, *partly-adaptive behaviour*, and *fully-adaptive behaviour*. The static restoration preplans model restored each span failure of a dual span failure scenario as if each were an isolated single span failure situation. That is, this model follows a predefined single failure restoration plan, in which if there is not enough allocation of spare capacity to support the complete restoration of both spans, the restoration routes of the second span are excluded. In the partly-adaptive behaviour model, there is an acknowledgement of the spare capacity used by the restoration routes of the first failed span, and therefore, any second span failure would adapt to the changes in available spare capacity in the network. In addition, if any restoration route of the first failed span traverses the second failed span, the restoration mechanism accumulates the working capacity requirements for the second failed span with the failed restoration routes of the first span failure. The fully-adaptive behaviour model is completely aware and able to adapt to both failed spans restoration routes spare capacity requirements as the partly-adaptive behaviour. However, if any restoration route of the first failed span is damaged by the second failed span, a spare capacity withdraw of those restoration routes is made and the restoration mechanism will try to find new restoration routes between the end-nodes. Having stated these points, one of the major contributions in [26] was that mesh-based span-restorable networks, designed for full single span failure restorability, are able to support a high average proportion of working capacity against dual span failures, which was especially true in the fully-adaptive model. Therefore, the work herein will follow this fully-adaptive technical model.

In this chapter, we introduce two meta-mesh network capacity formulations in order to enhance dual-failure restorability and hence availability in this specific design model. A natural first exercise is simply to design a meta-mesh network capable of withstanding all possible combinations of dual-failure situations, that is a network capable of assuring $R_2(i, j) = 1$. A next approach is to analyse how high an average dual failure restorability can be achieved on a meta-mesh network designed to outstand only single-failure scenarios, and if we add some extra spare capacity as a budget how much this would have to be in order to achieve a reasonable amount of dual-failure restorability. We first introduce the two ILP design models, followed by results and discussions.

5.2 META-MESH DUAL-FAILURE MINIMUM CAPACITY

We now develop a specific optimization model, in which we can explore the trade-offs and opportunities to design a meta-mesh network capable of withstanding all possible dual span failure scenarios. As mentioned in Chapter 3, dual failures scenarios are not merely theoretical [45], and so there is a practical reason to be concerned about providing dual failure restoration in real transport networks. Therefore, this type of scenario [45] opens the door to investigate the design of ensuring complete dual failure restorability in a mesh-based network. The *Meta-Mesh Dual-Failure Minimum Capacity* (MM-DFMC) formulation seeks to find the minimum total of working and spare capacity costs to guarantee full restorability of all dual span failure scenarios in a meta-mesh network. That is, this model shows us what is the minimum cost of achieving $R_2(i,j) = 100\%$ in meta-mesh restoration. We note here that the dual failure of any two spans within an individual chain is inherently not restorable, any nodes between the failed spans (and lightpaths originating from them) will be isolated. We, therefore, remove those dual span failures situations inside chains, which contain nodes of degree-2. Figure 5.1 illustrates the three types of dual failure situation contained in a meta-mesh network. Figure 5.1(a) portrays an original sparse network graph. Then, Figure 5.1(b) to Figure 5.1(d) illustrate the three types of meta-mesh dual span failure scenarios involved in our formulation tests, which are the following:

$\mathbf{S}_\mathrm{d} \times \mathbf{S}_\mathrm{d}$ set of all dual-failure scenarios where only direct spans are involved.

$\mathbf{S}_\mathrm{d} \times \mathbf{S}_\mathrm{c}$ set of all dual-failures scenarios where a direct span and a chain span are involved. In this case, the model is extended to convert this dual-failure situation into a *logical* triple-failure scenario.

$\mathbf{S}_\mathrm{c} \times \mathbf{S}_\mathrm{c}$ set of all dual-failures scenarios where only chain spans are involved. In this case, the model is also extended to convert this dual-failure situation into a *logical* quadruple-failure scenario (their respective logical bypass spans).

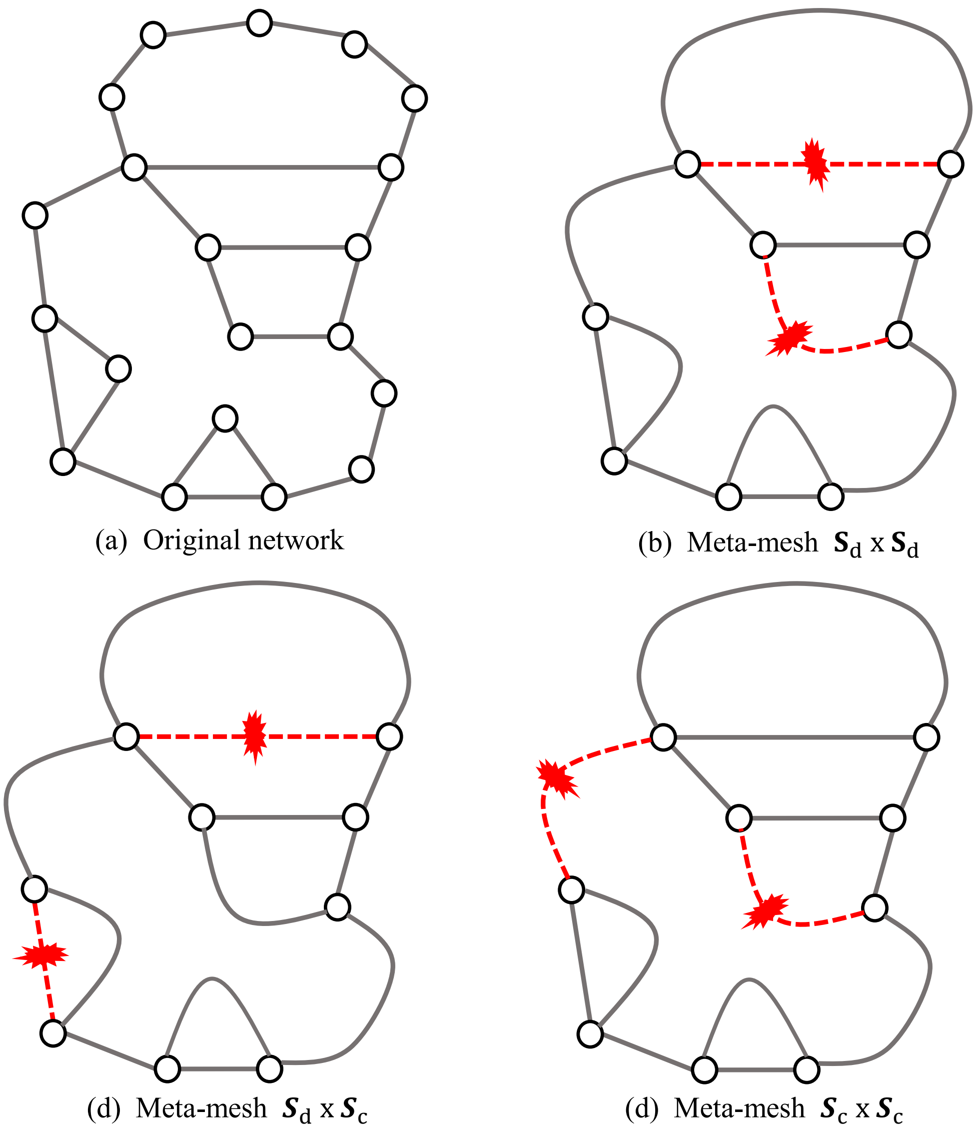


Figure 5.1 – Meta-Mesh dual failure scenarios

In our new ILP model, we used all previous JCA and meta-mesh ILP formulations already discussed in Chapter 2. That is, constraints from (2.8) to (2.12). In this way, we ensure restoration against any single-failure scenario inside chain subnetworks. Likewise, the objective function in this model is going to be the same, which minimizes the total cost of placing spare capacity in the network. Therefore, the formulation and the only new variable to introduce at this moment are:

$f_{i,j}^{p}$ is the restoration flow assigned to the pth eligible restoration route for span i when a span j has failed simultaneously

MM-DFMC: $$Minimize \sum_{j \in \boldsymbol{S}} C_j \cdot (s_j + w_j) \tag{5.1}$$

Subject to:

$$\sum_{p \in \boldsymbol{P}_i} f_{i,j}^{\ p} = w_i \qquad \forall (i,j) \in \boldsymbol{S_d^2} \mid i \neq j \tag{5.2}$$

$$\sum_{p \in \boldsymbol{P}_i \mid \delta_{i,k}^p = 0} f_{i,j}^{\ p} = w_i \qquad \forall (i,j) \in \boldsymbol{S_c} x \boldsymbol{S_d} \mid i \neq j, k = k(i) \tag{5.3}$$

$$\sum_{p \in \boldsymbol{P}_j \mid \delta_{j,k}^p = 0} f_{j,i}^{\ p} = w_j \qquad \forall (i,j) \in \boldsymbol{S_c} x \boldsymbol{S_d} \mid i \neq j, k = k(i) \tag{5.4}$$

$$\sum_{p \in \boldsymbol{P}_k \mid \delta_{k,i}^p = 0} f_{k,j}^{\ p} = w_k \qquad \forall (i,j) \in \boldsymbol{S_c} x \boldsymbol{S_d} \mid i \neq j, k = k(i) \tag{5.5}$$

$$\sum_{p \in \boldsymbol{P}_i \mid \delta_{i,k}^p = 0, \delta_{i,l}^p = 0} f_{i,j}^{\ p} = w_i \qquad \forall (i,j) \in \boldsymbol{S_c^2} \mid i \neq j, k = k(i), \quad l = l(j) \tag{5.6}$$

$$\sum_{p \in \boldsymbol{P}_k \mid \delta_{k,i}^p = 0, \delta_{k,l}^p = 0} f_{k,j}^{\ p} = w_k \qquad \forall (i,j) \in \boldsymbol{S_c^2} \mid i \neq j, k = k(i), \quad l = l(j) \tag{5.7}$$

$$\sum_{p \in \boldsymbol{P}_l \mid \delta_{l,j}^p = 0, \delta_{l,k}^p = 0} f_{l,i}^{\ p} = w_l \qquad \forall (i,j) \in \boldsymbol{S_c^2} \mid i \neq j, k = k(i), \quad l = l(j) \tag{5.8}$$

$$\sum_{p \in \boldsymbol{P}_i \mid \delta_{i,j}^p = 1} f_{i,j}^{\ p} = 0 \qquad \forall (i,j) \in \boldsymbol{S_d^2} \mid i \neq j \tag{5.9}$$

$$\sum_{p \in \boldsymbol{P}_i \mid \delta_{i,j}^p = 1} f_{i,j}^{\ p} = 0 \qquad \forall (i,j) \in \boldsymbol{S_c} x \boldsymbol{S_d} \mid i \neq j, k = k(i) \tag{5.10}$$

$$\sum_{p \in \boldsymbol{P}_j \mid \delta_{j,i}^p = 1} f_{j,i}^{\ p} = 0 \qquad \forall (i,j) \in \boldsymbol{S_c} x \boldsymbol{S_d} \mid i \neq j, k = k(i) \tag{5.11}$$

$$\sum_{p \in \boldsymbol{P}_k \mid \delta_{k,j}^p = 1} f_{k,j}^{\ p} = 0 \qquad \forall (i,j) \in \boldsymbol{S_c} x \boldsymbol{S_d} \mid i \neq j, k = k(i) \tag{5.12}$$

$$\sum_{p \in \boldsymbol{P}_i \mid \delta_{i,j}^p = 1} f_{i,j}^{\ p} = 0 \qquad \forall (i,j) \in \boldsymbol{S_c^2} \mid i \neq j, k = k(i), \quad l = l(j) \tag{5.13}$$

$$\sum_{p \in \boldsymbol{P}_k \mid \delta_{k,j}^{p}=1} f_{k,j}^{\ p} = 0 \qquad \forall (i,j) \in \boldsymbol{S}_{\boldsymbol{c}}^{\boldsymbol{2}} \mid i \neq j,\ k = k(i),\ \ l = l(j) \qquad (5.14)$$

$$\sum_{p \in \boldsymbol{P}_l \mid \delta_{l,i}^{p}=1} f_{l,i}^{\ p} = 0 \qquad \forall (i,j) \in \boldsymbol{S}_{\boldsymbol{c}}^{\boldsymbol{2}} \mid i \neq j,\ k = k(i),\ \ l = l(j) \qquad (5.15)$$

$$s_w \geq \sum_{p \in \boldsymbol{P}_i} \delta_{i,w}^{\ p} \cdot f_{i,j}^{\ p} + \sum_{p \in \boldsymbol{P}_j} \delta_{j,w}^{\ p} \cdot f_{j,i}^{\ p} \qquad \forall (i,j) \in \boldsymbol{S}_{\boldsymbol{d}}^{\boldsymbol{2}}\ x\ \boldsymbol{S} \mid i \neq j \qquad (5.16)$$

$$s_w \geq \sum_{p \in \boldsymbol{P}_i} \delta_{i,w}^{\ p} \cdot f_{i,j}^{\ p} + \sum_{p \in \boldsymbol{P}_j} \delta_{j,w}^{\ p} \cdot f_{j,i}^{\ p} + \sum_{p \in \boldsymbol{P}_k} \delta_{k,w}^{\ p} \cdot f_{k,j}^{\ p} \qquad \forall (i,j) \in \boldsymbol{S}_{\boldsymbol{c}} x \boldsymbol{S}_{\boldsymbol{d}} x \boldsymbol{S} \mid i \neq j, k = k(i) \qquad (5.17)$$

$$s_w \geq \sum_{p \in \boldsymbol{P}_i} \delta_{i,w}^{\ p} \cdot f_{i,j}^{\ p} + \sum_{p \in \boldsymbol{P}_j} \delta_{j,w}^{\ p} \cdot f_{j,i}^{\ p} + \sum_{p \in \boldsymbol{P}_k} \delta_{k,w}^{\ p} \cdot f_{k,j}^{\ p} + \sum_{p \in \boldsymbol{P}_l} \delta_{l,w}^{\ p} \cdot f_{l,i}^{\ p} \qquad \forall (i,j) \in \boldsymbol{S}_{\boldsymbol{c}} x \boldsymbol{S}_{\boldsymbol{d}} x \boldsymbol{S} \mid k = k(i), l = l(j) \qquad (5.18)$$

The objective function pursues to minimize the total cost of assigning spare and working capacity for each single or dual span failure in the network. Equation (5.2) asserts the restorability of each direct span failure *i* under all direct dual-failure situations, that is where only direct spans or spans with degree-3 or higher are involved $\mathbf{S}_{\mathrm{d}}^{\mathbf{2}}$. In the same way, all set of constraints (5.3), (5.4), and (5.5) ensure the restorability of each span failure *i* under all dual-failure situations where only a span inside a chain subnetwork and a direct span are involved, that is $\mathbf{S}_{\mathrm{d}}$ x $\mathbf{S}_{\mathrm{c}}$. Similarly, equations (5.6), (5.7), and (5.8) ensure full restoration of all dual-failures scenarios where only spans in chains are involved, that is $\mathbf{S}_{\mathrm{c}}^{\mathbf{2}}$. Note that during the dual-failure $\mathbf{S}_{\mathrm{d}}$ x $\mathbf{S}_{\mathrm{c}}$ situation the mathematical model is extended to convert dual physical cuts into the corresponding logical triple-failure scenario between the chain span and its associated logical bypass span. Likewise, during the dual-failure $\mathbf{S}_{\mathrm{c}}^{\mathbf{2}}$ situation the mathematical model is augmented but at the same time it translates dual physical cuts into a logical quadruple-failure scenario between both chain spans and their associated logical bypass spans. Constraint (5.9) guarantees that direct span *j* cannot be used for restoration of direct span *i* in case of dual-failure situation (*i*, *j*), but it can support restoration flow when it is not a part of the failure scenario. Similarly, constraints (5.10), (5.11), and (5.12) ensure that in the scenario of a chain span failure and a direct span failure $\mathbf{S}_{\mathrm{d}}$ x $\mathbf{S}_{\mathrm{c}}$ span *j* can support any amount of restoration flow when is not part of the failure scenario, but it cannot be used for restoration of span *i* under that specific (*i*, *j*) failure situation. In the same manner, constraints (5.13), (5.14), and (5.15) do the same that the previous constraints but at the level of chain spans

$\mathbf{S}_c^2$. Finally, all set of constraints (5.16), (5.17), and (5.18) guarantee enough amount of spare capacity in all spans in the network for the restoration of any dual span failure under $\mathbf{S}_\mathrm{d}^2$, $\mathbf{S}_c^2$, and $\mathbf{S}_\mathrm{d}$ x $\mathbf{S}_\mathrm{c}$ failure scenarios. The AMPL model developed is detailed in Section D.1 of Appendix D.

5.3 META-MESH DUAL-FAILURE MAXIMUM RESTORABILITY

A relevant concern of designing a meta-mesh network capable of withstanding all possible dual failure situation, and therefore, providing $R_2(i, j)$ = 100% is the inevitable high cost related with it. In fact, this will be confirmed by the results that follow. From the network operator standpoint, it may be questionable the allocation of a high amount of spare capacity to be capable of supporting these types of dual failure situations. It is logical, therefore, to investigate what is the highest average of dual failure restorability that can be achieved with a specific amount of spare capacity placed on a meta-mesh network design. That is, if we allow a certain increase in spare capacity investment in the network, how high our average dual failure restorability would be? Therefore, we introduce the *Meta-Mesh Dual-Failure Maximum Restorability* (MM-DFMR) design, which seeks to find the minimum number of non-restored working capacities $N(i, j)$ of a meta-mesh network designed to support single-failure scenarios. In this way, we are maximizing our dual failure restorability by minimizing the number of non-restored working capacities. This is far more meaningful and practical for network operators, in which they know in advance how much average $R_2(i, j)$ they can expect in a dual failure event. The MM-DFMR ILP formulation uses all previous JCA and meta-mesh ILP formulation mentioned in Chapter 2. That is, equations from (2.8) to (2.12). Likewise, the model uses the same parameters and variables applied before with the exception of the following added notation:

$N(i, j)$ is the number of non-restored working capacities under a dual-failure scenario.

B is the total amount of spare capacity available as an investment in the network design.

Nevertheless, the objective function changes to aim the minimization of the number of non-restored working capacities in the meta-mesh network design. The AMPL model developed is detailed in Section C.1 of Appendix C. The MM-DFMR ILP formulation is the following:

MM-DFMR:

$$Minimize \sum_{(i,j)\in \mathbf{S}^2 \mid i\neq j} N(i,j) \tag{5.19}$$

Subject to:

$$N(i,j) = w_i + w_j - \left(\sum_{p\in \boldsymbol{P_i}} f_{i,j}^{\ p} + \sum_{p\in \boldsymbol{P_j}} f_{j,i}^{\ p}\right) \quad \forall (i,j) \in \boldsymbol{S_d} x \boldsymbol{S} \mid i \neq j \tag{5.20}$$

$$N(i,j) = w_i + w_j + w_k - \left(\sum_{\substack{p\in \boldsymbol{P_i} \\ \mid \delta_{i,k}^p = 0}} f_{i,j}^{\ p} + \sum_{\substack{p\in \boldsymbol{P_j} \\ \mid \delta_{j,k}^p = 0}} f_{j,i}^{\ p} + \sum_{\substack{p\in \boldsymbol{P_k} \\ \mid \delta_{k,i}^p = 0}} f_{k,j}^{\ p}\right) \quad \forall (i,j) \in \boldsymbol{S_c} x \boldsymbol{S_d} \mid i \neq j, k = k(i) \tag{5.21}$$

$$N(i,j) = w_i + w_j + w_k + w_l - \left(\sum_{\substack{p\in \boldsymbol{P_i} \\ \mid \delta_{i,k}^p = 0, \\ \delta_{i,l}^p = 0}} f_{i,j}^{\ p} + \sum_{\substack{p\in \boldsymbol{P_j} \\ \mid \delta_{j,k}^p = 0, \\ \delta_{j,l}^p = 0}} f_{j,i}^{\ p} + \sum_{\substack{p\in \boldsymbol{P_k} \\ \mid \delta_{k,i}^p = 0, \\ \delta_{k,l}^p = 0}} f_{k,j}^{\ p} + \sum_{\substack{p\in \boldsymbol{P_l} \\ \mid \delta_{l,j}^p = 0, \\ \delta_{l,k}^p = 0}} f_{l,i}^{\ p}\right) \quad \forall (i,j) \in \boldsymbol{S_c^2} \mid i \neq j, k = k(i), \ l = l(j) \tag{5.22}$$

$$\sum_{p\in \boldsymbol{P}_i} f_{i,j}^{\ p} \leq w_i \quad \forall (i,j) \in \boldsymbol{S_d^2} \mid i \neq j \tag{5.23}$$

$$\sum_{p\in \boldsymbol{P}_i \mid \delta_{i,k}^p = 0} f_{i,j}^{\ p} \leq w_i \quad \forall (i,j) \in \boldsymbol{S_c} x \boldsymbol{S_d} \mid i \neq j, k = k(i) \tag{5.24}$$

$$\sum_{p\in \boldsymbol{P}_j \mid \delta_{j,k}^p = 0} f_{j,i}^{\ p} \leq w_j \quad \forall (i,j) \in \boldsymbol{S_c} x \boldsymbol{S_d} \mid i \neq j, k = k(i) \tag{5.25}$$

$$\sum_{p\in \boldsymbol{P}_k \mid \delta_{k,i}^p = 0} f_{k,j}^{\ p} \leq w_k \quad \forall (i,j) \in \boldsymbol{S_c} x \boldsymbol{S_d} \mid i \neq j, k = k(i) \tag{5.26}$$

$$\sum_{p\in \boldsymbol{P}_i \mid \delta_{i,k}^p = 0, \delta_{i,l}^p = 0} f_{i,j}^{\ p} \leq w_i \quad \forall (i,j) \in \boldsymbol{S_c^2} \mid i \neq j, k = k(i), \ l = l(j) \tag{5.27}$$

$$\sum_{p\in \boldsymbol{P}_k \mid \delta_{k,i}^p = 0, \delta_{k,l}^p = 0} f_{k,j}^{\ p} \leq w_k \quad \forall (i,j) \in \boldsymbol{S_c^2} \mid i \neq j, k = k(i), \ l = l(j) \tag{5.28}$$

$$\sum_{p\in \boldsymbol{P}_l \mid \delta_{l,j}^p = 0, \delta_{l,k}^p = 0} f_{l,i}^{\ p} \leq w_l \quad \forall (i,j) \in \boldsymbol{S_c^2} \mid i \neq j, k = k(i), \ l = l(j) \tag{5.29}$$

$$\sum_{p \in \boldsymbol{P}_i \mid \delta_{i,j}^p = 1} f_{i,j}^{\ p} \ = \ 0 \qquad \forall (i,j) \in \boldsymbol{S}_{\boldsymbol{d}}^{\mathbf{2}} \mid \ i \neq j \tag{5.30}$$

$$\sum_{p \in \boldsymbol{P}_i \mid \delta_{i,j}^p = 1} f_{i,j}^{\ p} \ = \ 0 \qquad \forall (i,j) \in \boldsymbol{S}_{\boldsymbol{c}} x \boldsymbol{S}_{\boldsymbol{d}} \mid i \neq j, k = k(i) \tag{5.31}$$

$$\sum_{p \in \boldsymbol{P}_j \mid \delta_{j,i}^p = 1} f_{j,i}^{\ p} \ = \ 0 \qquad \forall (i,j) \in \boldsymbol{S}_{\boldsymbol{c}} x \boldsymbol{S}_{\boldsymbol{d}} \mid i \neq j, k = k(i) \tag{5.32}$$

$$\sum_{p \in \boldsymbol{P}_k \mid \delta_{k,j}^p = 1} f_{k,j}^{\ p} \ = 0 \qquad \forall (i,j) \in \boldsymbol{S}_{\boldsymbol{c}} x \boldsymbol{S}_{\boldsymbol{d}} \mid i \neq j, k = k(i) \tag{5.33}$$

$$\sum_{p \in \boldsymbol{P}_i \mid \delta_{i,j}^p = 1} f_{i,j}^{\ p} \ = \ 0 \qquad \forall (i,j) \in \boldsymbol{S}_{\boldsymbol{c}}^{\mathbf{2}} \mid i \neq j, k = k(i), \quad l = l(j) \tag{5.34}$$

$$\sum_{p \in \boldsymbol{P}_k \mid \delta_{k,j}^p = 1} f_{k,j}^{\ p} = 0 \qquad \forall (i,j) \in \boldsymbol{S}_{\boldsymbol{c}}^{\mathbf{2}} \mid i \neq j, k = k(i), \quad l = l(j) \tag{5.35}$$

$$\sum_{p \in \boldsymbol{P}_l \mid \delta_{l,i}^p = 1} f_{l,i}^{\ p} \ = \ 0 \qquad \forall (i,j) \in \boldsymbol{S}_{\boldsymbol{c}}^{\mathbf{2}} \mid i \neq j, k = k(i), \quad l = l(j) \tag{5.36}$$

$$s_w \geq \sum_{p \in \boldsymbol{P}_i} \delta_{i,w}^{\ p} \cdot f_{i,j}^{\ p} + \sum_{p \in \boldsymbol{P}_j} \delta_{j,w}^{\ p} \cdot f_{j,i}^{\ p} \qquad \forall (i,j) \in \boldsymbol{S}_{\boldsymbol{d}}^{\mathbf{2}} \ x \ \boldsymbol{S} \mid i \neq j \tag{5.37}$$

$$s_w \geq \sum_{p \in \boldsymbol{P}_i} \delta_{i,w}^{\ p} \cdot f_{i,j}^{\ p} + \sum_{p \in \boldsymbol{P}_j} \delta_{j,w}^{\ p} \cdot f_{j,i}^{\ p} + \sum_{p \in \boldsymbol{P}_k} \delta_{k,w}^{\ p} \cdot f_{k,j}^{\ p} \qquad \forall (i,j) \in \boldsymbol{S}_{\boldsymbol{c}} x \boldsymbol{S}_{\boldsymbol{d}} x \boldsymbol{S} \mid i \neq j, k = k(i) \tag{5.38}$$

$$s_w \geq \sum_{p \in \boldsymbol{P}_i} \delta_{i,w}^{\ p} \cdot f_{i,j}^{\ p} + \sum_{p \in \boldsymbol{P}_j} \delta_{j,w}^{\ p} \cdot f_{j,i}^{\ p} + \sum_{p \in \boldsymbol{P}_k} \delta_{k,w}^{\ p} \cdot f_{k,j}^{\ p} + \sum_{p \in \boldsymbol{P}_l} \delta_{l,w}^{\ p} \cdot f_{l,i}^{\ p} \qquad \forall (i,j) \in \boldsymbol{S}_{\boldsymbol{c}} x \boldsymbol{S}_{\boldsymbol{d}} x \boldsymbol{S} \mid k = k(i), l = l(j) \tag{5.39}$$

$$\sum_{j \in \boldsymbol{S}} C_j \ \cdot \ \left(s_j \ + \ w_j\right) \leq B \tag{5.40}$$

As mentioned earlier, the objective function (5.19) asserts to minimize the number of non-restored working capacities. Note that we aim to maximize the R_2 of this network design by minimizing the sum $N(i,j)$ over all dual-failure scenarios. Equation (5.20) aims to express the quantity of non-restored working capacities that are not restored in case of a dual failure situation on two direct spans $\mathbf{S}_{\mathrm{d}}^{\mathbf{2}}$ (i,j). In the same way, Equation (5.21) defines the number of $N(i,j)$ over a dual failure on a chain subnetwork and a direct span $\mathbf{S}_{\mathrm{d}} \times \mathbf{S}_{\mathrm{c}}$. Likewise, Equation (5.22) asserts the number of $N(i,j)$ over a dual failure situation on two chain spans $\mathbf{S}_{\mathrm{c}}^{\mathbf{2}}$ (i, j). Equation (5.23) pursues to assign as much of the restoration flow as it can to restore as many working channels as it can over a dual failure situation on two direct spans (i, j). In the same manner, Equations from (5.24) to (5.26) ensure that the number of restoration paths assigned for restoration of span i in a dual failure $\mathbf{S}_{\mathrm{d}} \times \mathbf{S}_{\mathrm{c}}$ situation is at most equal to the number of working channels to be restored. Equations from (5.27) to (5.29) aim to achieve the same thing than the last equation but for a dual failure situation on two chain spans. Constraints from (5.30) to (5.36) force the exclusion of using span j for restoration of span i over a dual failure situation (i, j) during these dual failures scenarios $\mathbf{S}_{\mathrm{d}}^{\mathbf{2}}$, $\mathbf{S}_{\mathrm{c}}^{\mathbf{2}}$, and $\mathbf{S}_{\mathrm{d}} \times \mathbf{S}_{\mathrm{c}}$ but allow to use it when is not part of the failure scenario. Constrains set from (5.37) to (5.39) assert an adequate amount of spare capacity for every dual failure situation over $\mathbf{S}_{\mathrm{d}}^{\mathbf{2}}$, $\mathbf{S}_{\mathrm{c}}^{\mathbf{2}}$, and $\mathbf{S}_{\mathrm{d}} \times \mathbf{S}_{\mathrm{c}}$ scenarios that the budget-limited formulation chooses to cover. Finally, constraint (5.40) enforces a budget limit of spare capacity allowed to use.

5.4 EXPERIMENTAL SET-UP

As mentioned in Chapter 3, we used a set of six network families that were previously used in the prior meta-mesh analysis [5]. Each formulation was implemented as an AMPL model and solved using Gurobi 6.5.0 solver on a 12-core ACPI multiprocessor X64-based PC with Intel Xeon® CPU E5-2430 running at 2.2 GHz with 96 GB RAM. All results are based on a default mipgap of 0.0001, meaning they are ensured to be within 0.01% of optimal. Pre-processing for eligible restoration and working routes and other input parameters was done on an Intel® Core™ i7 notebook with 8.0 GB of RAM, running Windows 7. Also, all spare and working were integer.

All the ILP models presented in this thesis are based on the arc-path formulation type [7], which requires a preprocessing of the network graph to represent the sets of eligible restoration and working routes. Typically, this process is done by generating all distinct routes up to a hot limit *H*, or by setting a minimum number of the closest distinct eligible restoration and working routes of each span [27]. The later approach is the one used in the development of this thesis. However, during the development of the experiments, some special topology design considerations arose. This is explained in the following section.

5.4.1 TOPOLOGY DESIGN CONSIDERATIONS

It is worthwhile to remember that dual failures of any two spans within an individual chain are inherently not restorable, as a network must be at least tri-connected in order to achieve full dual failure restorability, R_2 (i, j). Therefore, dual failures are logically removed from chain subnetworks. However, note that we allowed single span failures inside these chains to exploit all savings in spare capacity of the meta-mesh design. This is because this model produces its best benefits where there are significant express capacity flows through chains.

Besides the above unfeasibility scenario, there are two more situations where a not feasible solution arose. We observed from our preliminary investigations that another subset of dual failures of all three types ($\mathbf{S_d x S_d}$, $\mathbf{S_d x S_c}$, and $\mathbf{S_c x S_c}$) could disconnect the network graph into two sections and in some networks into three sections. Furthermore, an insufficient diversity of eligible working and restoration routes options yielded unfeasibility results. These situations will be better explained and discussed in the coming section.

5.4.1.1 NETWORK DISCONNECTION

When designing and evaluating both presented network survivability schemes, a great number of them were presenting a network disconnection scenario because of some specific dual span failure situation. Out of the 124 test networks, 69 of them were disconnected. In fact, we had cases where a dual-disconnection and even a triple-disconnection showed up. Each disconnected test network required an independent DFMC and DFMR ILP design to isolate the disconnection per se. Table 5.1 displays the number of disconnection events in each network family used during experiments of both integer linear programming formulation models. As can be seen, 53 test networks presented single-disconnection, 13 presented dual-disconnection, and only 3 of them presented triple-disconnection.

Table 5.1 – Number of disconnection events in each network family

Network family	Single-disconnection	Dual-disconnection	Triple-disconnection
15n30s1	–	–	–
20b40s1	12 out of 15	1 out of 15	–
25n50s1	3 out of 19	6 out of 19	–
30n60s1	11 out of 24	–	3 out of 24
35n70s1	16 out of 23	1 out of 23	–
40n80s1	11 out of 34	5 out of 34	–

It is clear that this problem can be overcome by logically removing dual failures on those spans causing disconnection. In this way, the only failure that is allowed in these spans are single-failure scenarios. Figure 5.2 illustrates a classic network disconnection scenario where a dual failure situation on a direct span and in any span inside a chain subnetwork leads to a not feasible solution. This example is similar in nature to a test network inside the 20n40s1 network family.

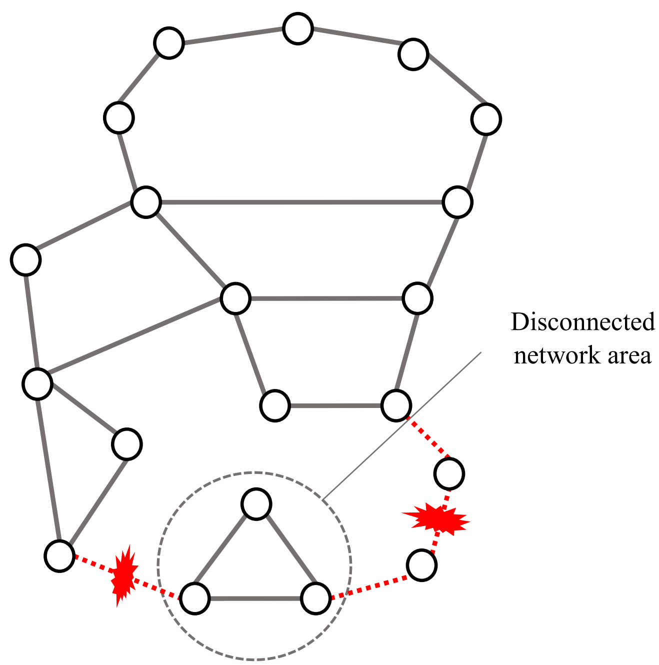


Figure 5.2 – Illustrating dual-failure infeasibility in a 20n26s meta-mesh network

5.4.1.2 Increasing the Number of Eligible Restoration and Working Routes

In survivable span-restorable network designs, the number of restoration and working path-sets is often pre-defined and pre-calculated based up to a compromise shortest route hop limit, *H*, or on a minimum amount of the shortest eligible routes as it was described in Chapter 3. A usual policy is to define a double amount of restoration routes than working routes [7]. In other words, if we define a minimum number of 10 restoration routes for each span failure we are going to have at least 5 working routes for each origin and destination (O-D) demand pair. In the previous meta-mesh single-failure analysis [5], the entire network tests cases were pre-processed with at least 5 shortest working routes between each O-D node pair. In the same manner, all the pre-processed eligible restoration routes were enumerated base on the 10 shortest routes between the immediate end-nodes.

With this in mind, a great number of test networks topologies presented an unfeasibility problem because of an insufficient number of eligible routes. That is, the preprocessed number of eligible working and restoration routes were not enough to find at least one path between their origin and destination pair to route their demands (in case of working flow) or their restoration capacity (in case of restoration flow). Figure 5.3 illustrates an example where a dual failure situation leads to a not feasible solution in the 20n26s test network. In this case, each dashed line represents a different working path option to route the particular demand between this O-D pair (node A and node B). Notice that if a failure in the specified spans arises they do no have any path left to route their demands. The same idea applies for restorations routes, where we had to increase their number because the solver could not find any eligible restoration route. This was overcome by increasing the number of eligible working routes for the problematic demand or the number of eligible restoration routes for the problematic failure scenario, as the case may be, until infeasibility is repaired. Ultimately, the exact number of working and restoration routes used in each network family can be found in Appendix B.

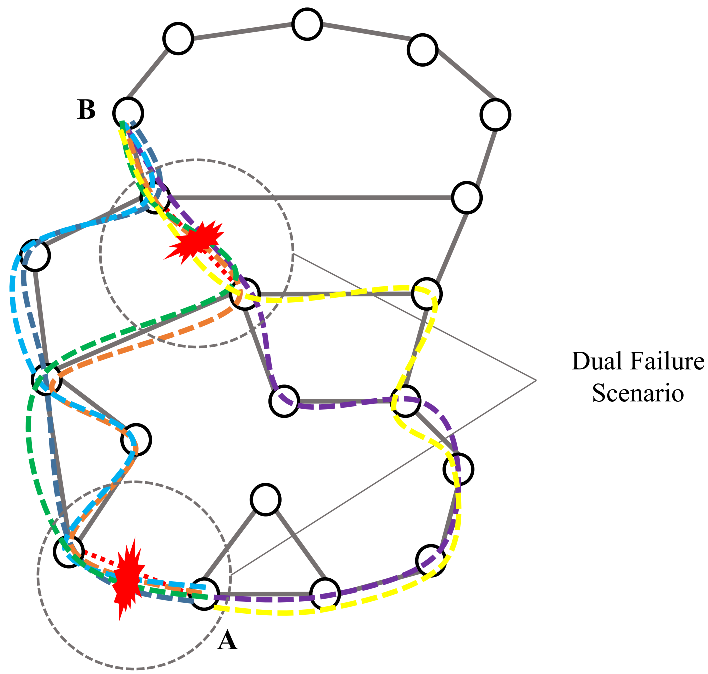


Figure 5.3 – Illustrating dual-failure working routes infeasibility in a 20n26s meta-mesh network

5.5 Result and Discussion

5.5.1 MM-DFMC Results and Discussion

Results from the MM-DFMC design experiments are presented and discussed in this section. Figures 5.4 to 5.9 illustrate the normalized spare capacity cost of optimally designed networks of the various families designed to be 100% single-failure as well as 100% dual-failure restorable using the meta-mesh design restoration. Each figure provides data for a single network family, and each data point represents the total of spare capacity required to route all demands between each O-D pair as well as provides full single-failure or dual-failure restoration for the member of the family with the indicated average nodal degree using the single-failure meta-mesh or meta-mesh dual-failure minimum capacity design. Each solid line represents the total spare cost of achieving single-failure and dual-failure restorability. As expected, these results demonstrate that the price of strictly assuring dual failure restorability, that is $R_2(i, j) = 100\%$, in these network families is quite high relative to the requirement for single-failure restorability only. In addition, these figures show that total spare cost decreases relatively uniformly as the average nodal degree increases, however, there are significant plateaus and peaks that are related to the topology design of the network, which led to not feasible solutions. This leads to some interesting understanding about how the topology design of a network responds to the relative costs of working and spare capacity. Results for the network families of 15n30s1, 20n40s1, 25n50s1, and 30n60s1 took less than one minute to solve, except for the most richly connected networks from the 35n70s1 and 40n80s1 network family that actually took a couple of minutes to solve.

In the 15n30s network family, it appears that one would have to invest 1.7 to 3.6 times as much spare capacity as otherwise to obtain full dual failure restorability. This particular network family does not present any unfeasibility issues other than the increase on the number of eligible restoration routes up to 20 in a few networks topologies. However, the entire test networks inside this network family did not required the increase of the number of eligible working paths to route their demands. This means that we used the same amount of five working options as in the prior work [5]. The same trend appeared for the 20-node network family, where the amount of spare capacity needed for dual failure restorability is between 1.6 and 3.0 times higher than the quantity needed for full single-failure restorability. A particular problem in this network family was the unfeasibility issue due to network disconnection where 13 out of 15 networks exhibited at least disconnection event arising from the scenarios described in Section 5.4.1.1. Specifically, 13 out of 15, presented network disconnection because of the sparseness of the network topology. In fact, one network topology, 20-node and 24-spans, presented dual-disconnection, which is reflected in the graph on a peak in the spare capacity costs. Similarly, it was quite common that we needed to increase the number of eligible working and/or restoration routes per demand or failure scenario, in some cases up to 20 eligible routes per scenario. This unfeasibility issue is also reflected in the mentioned graph where the dual-failure meta-mesh spare capacity line in illustrate some peaks and downs as the average nodal degree increases. These irregularities are directly related to the difference in the number of eligible working and restoration routes options between the test networks inside this family. Although we used the same number of working and restoration routes for our single-failure as well as dual-failure analysis, the latter is obligated to use long-path options to allocate restoration flow because of the dual span failure and disconnection event. In contrast, the line from the single-failure scenario (benchmark) shows a smooth decreasing transition as the average nodal degree increases. However, we note that both of these problems arise simply because of the sparse nature of the networks in question, not due to the meta-mesh approach.

One surprise was that in some networks, and in particular with the sparsest member of the 25-node family, full dual span failure restorability was possible with only a very small increase in spare capacity investment. More specifically, the 25-node network with $\bar{d}$ = 2.16 required only a 20% increase in spare capacity to provide full dual failure restorability. However, we note here that such extreme cases are likely due to the fact that we consider only dual failure scenarios where full restorability is even possible; as discussed in Section 5.2, we do not consider dual failure scenarios that disconnect the network or isolate nodes within chains. Despite this, the rest of the networks inside this family required the same high amount of spare capacity, as much as 4 times, to ensure $R_2 = 1$. Furthermore, 6 networks out of 19 presented a double-disconnection problem as well as it was required to increase the number of working and restoration routes up to 30 in order to they were capable of yielding a feasible solution.

The 30n60s network family required an investment of 1.6 to 3.5 times of spare capacity to achieve full dual-failure restorability. However, this network family contained unfeasibility issues of triple-disconnection, presented on three networks, as well as required an increase on the number of eligible working and restoration routes up to 60 in some networks. The difference between the number of eligible working and restoration routes used inside this family goes between 5 working routes and 10 restoration routes up to 60 of each. Note that this does not mean that all 60 eligible routes were used, rather, simply that none of the 59 shortest could lead to a feasible solution. This is of course, reflected on the Figure 5.6 in a series of alternations on the spare capacity line of the dual-failure meta-mesh design. Finally, the most richly connected network families of 35n70s and 40n80s presented some interesting results. In terms of spare capacity investment, on the 35n70s network family, it appears that we would have to invest between 1.9 to 3.7 times more to obtain full dual failure restorability. In this network family, 17 out 23 networks presented single-disconnection and only one of them presented dual-disconnection. In this particular network that presented dual-disconnection, 35n53s, as well as in the 35n54s network, we had to increase the number of working and restoration routes up to 240 in order to achieve a feasible solution. This was somehow surprising because, although we increased the number of eligible restoration and working routes before up to 60, it turns out that we must have 240 shortest eligible working and restoration routes to achieve a feasible solution in these networks. In this way, the number of eligible working and restoration routes used in this network family ranged from 5 working routes and 10 restoration routes up to 240 working routes and 240 restoration routes. As in previous network family graphs, this is the reason behind those peaks and downs on the spare capacity cost lines for the 35n70s network family. Ultimately, in the 40n80s network family, we required between 1.9 and 3.2 times more spare capacity in order to ensure full dual-failure restorability. As in other network families, 16 out of 34 networks were disconnected, and 5 of them presented double-disconnection. As opposed to the previous network family, in this family it was only required to increase the number of working paths up to 30 and the number of eligible restoration paths up to 20.

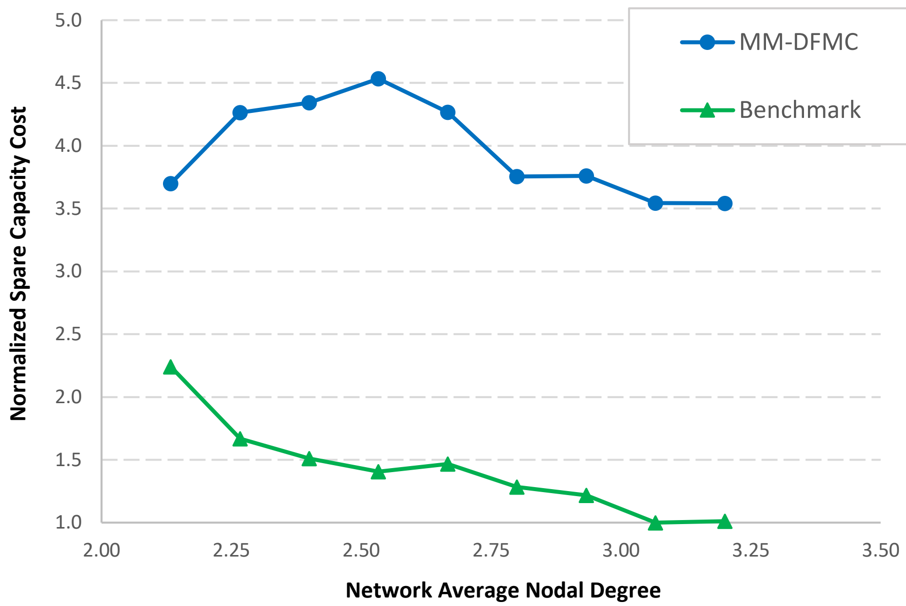


Figure 5.4 – Normalized MM-DFMC and Meta-Mesh spare capacity cost on the 15n30s1 network family

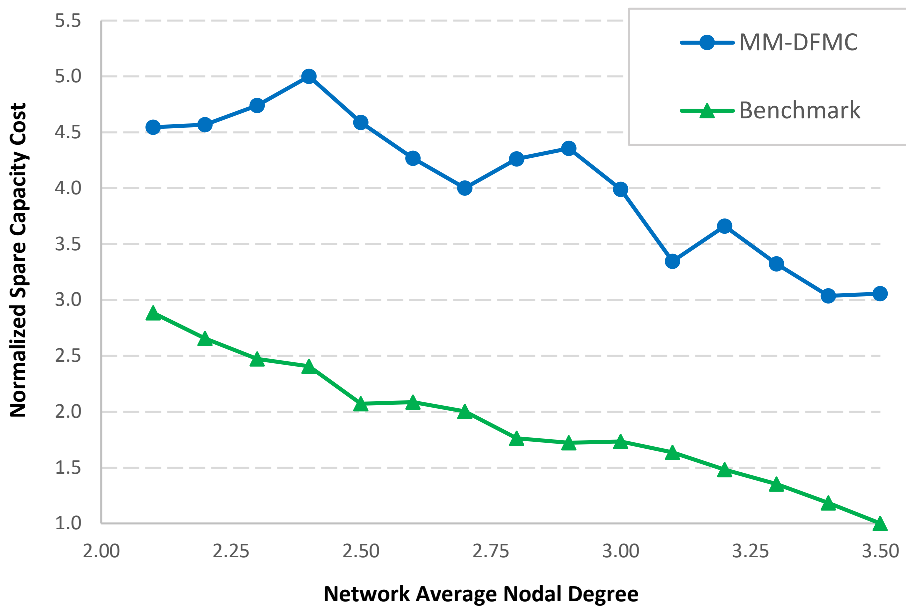


Figure 5.5 – Normalized MM-DFMC and Meta-Mesh spare capacity cost on the 20n40s1 network family

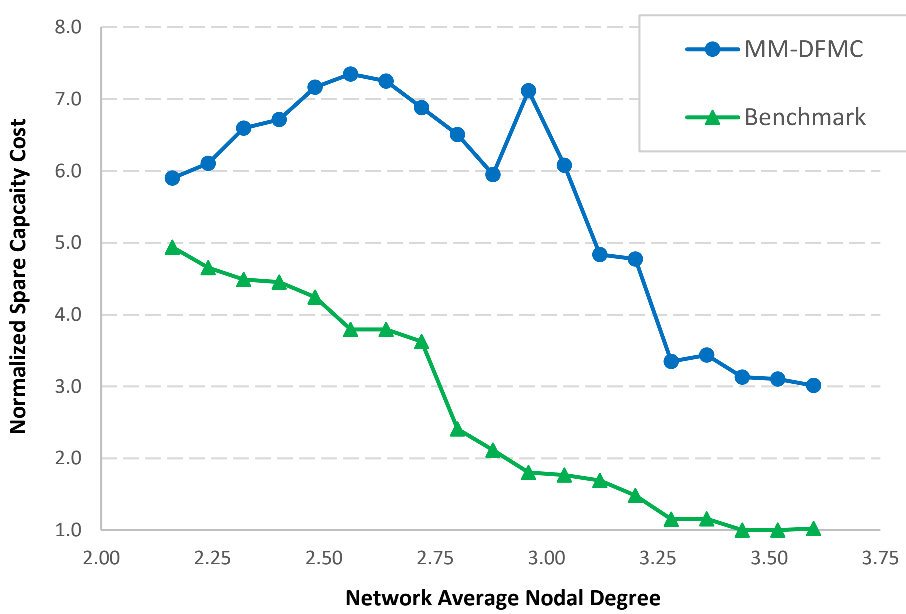


Figure 5.6 – Normalized MM-DFMC and Meta-Mesh spare capacity cost on the 25n50s1 network family

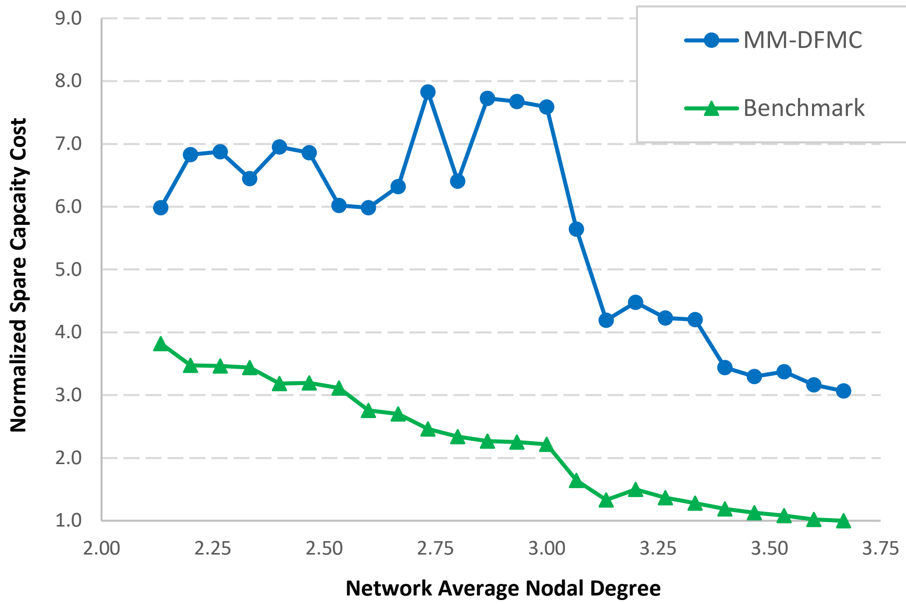


Figure 5.7 – Normalized MM-DFMC and Meta-Mesh spare capacity cost on the 30n60s1 network family

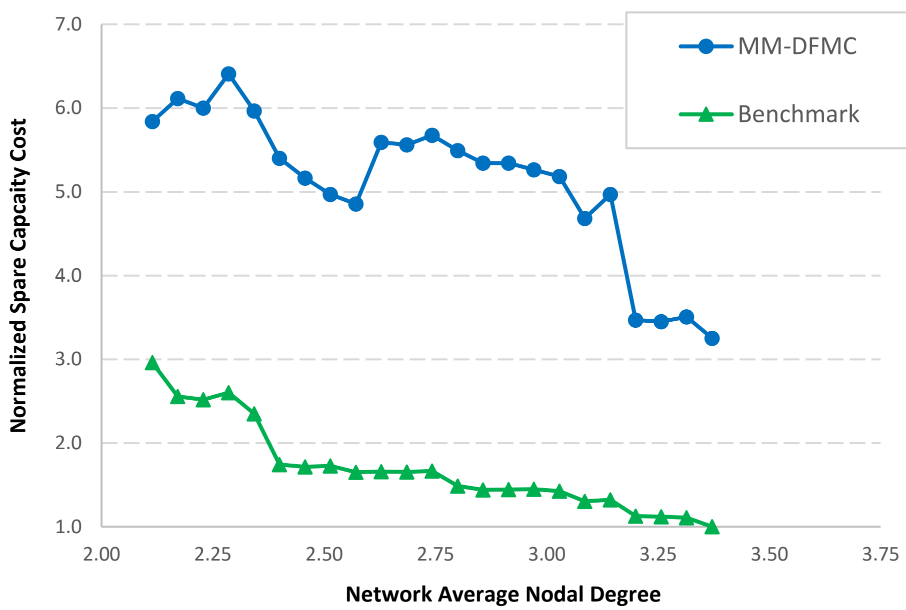


Figure 5.8 – Normalized MM-DFMC and Meta-Mesh spare capacity costs on the 35n70s1 network family

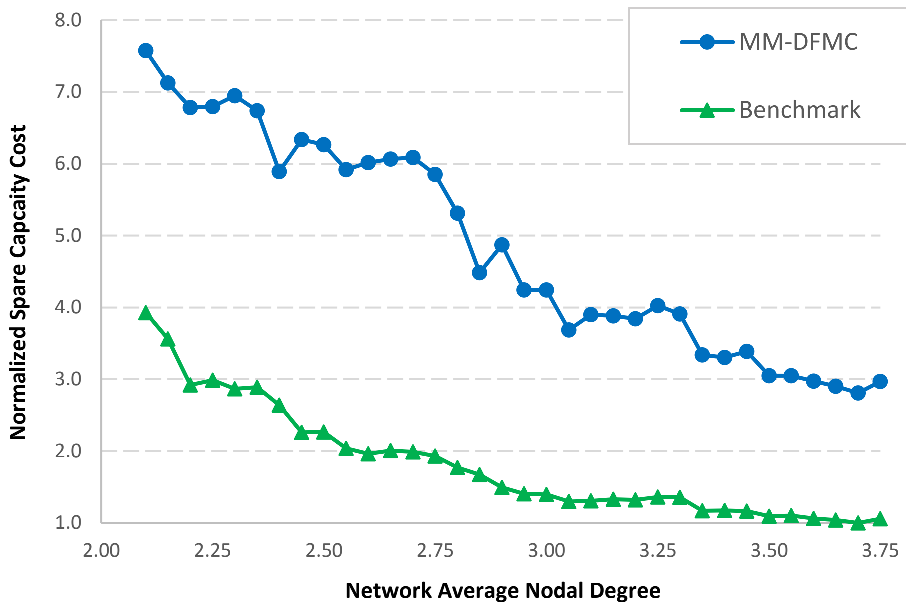


Figure 5.9 – Normalized MM-DFMC and Meta-Mesh spare capacity costs on the 40n80s1 network family

Similarly, Figure 5.10 through Figure 5.15 portray the normalized total capacity cost of designing a meta-mesh network to withstand all dual-failure scenarios, except for those dual failure situations that are inherently not restorable. As for the graphs above, each figure provides data for a single network family, and each data point represents the total (working and spare) capacity required to routes all demands between each origin and destination pair as well as provides full single-failure or dual-failure restoration for the member of the family with the indicated average nodal degree using the single-failure meta-mesh or dual-failure meta-mesh minimum capacity model. Each solid line represents the total (working and spare) cost of achieving single-failure and dual-failure restorability. Evidently, these results prove that providing dual failure survivability in sparse networks topologies is exceedingly difficult.

In addition, table 5.2 portrays the total results with the MM-DFMC formulation model for each of the six network families. As you can see, this table shows the total spare capacity and total (working and spare) needed to ensure full single and dual failure restorability using the meta-mesh model. Note that when a full dual failure restorability is ensured, a full single-failure restorability is also implied. This table also illustrates the relative increase of the MM-DFMC design compared to the single-failure meta-mesh design formulation model.

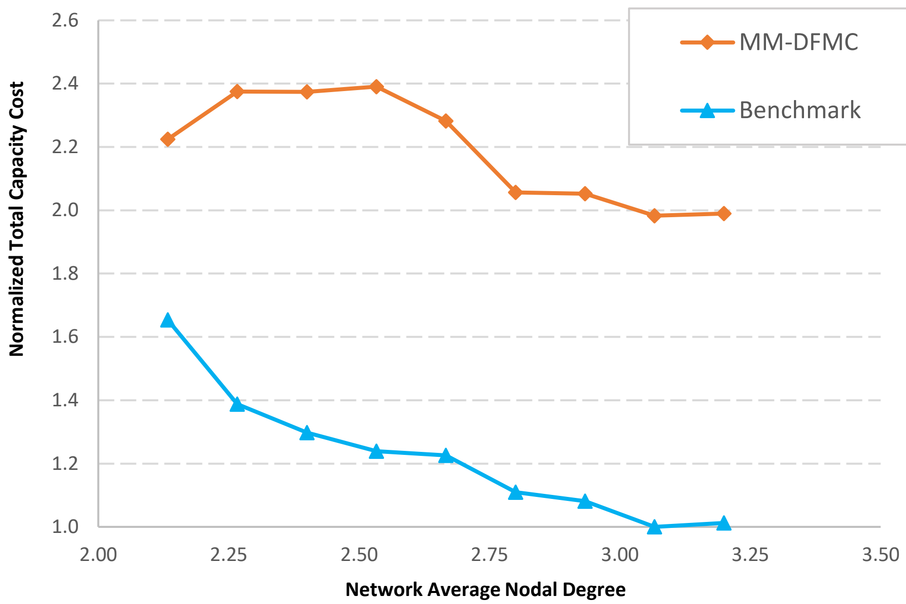


Figure 5.10 – Normalized MM-DFMC and Meta-Mesh total capacity cost on the 15n30s1 network family

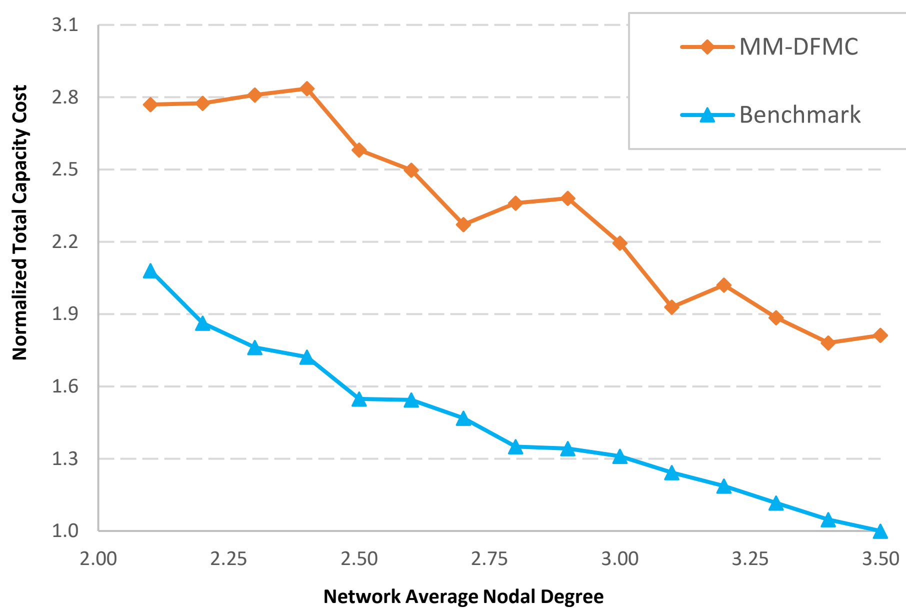


Figure 5.11 – Normalized MM-DFMC and Meta-Mesh total capacity cost on the 20n40s1 network family

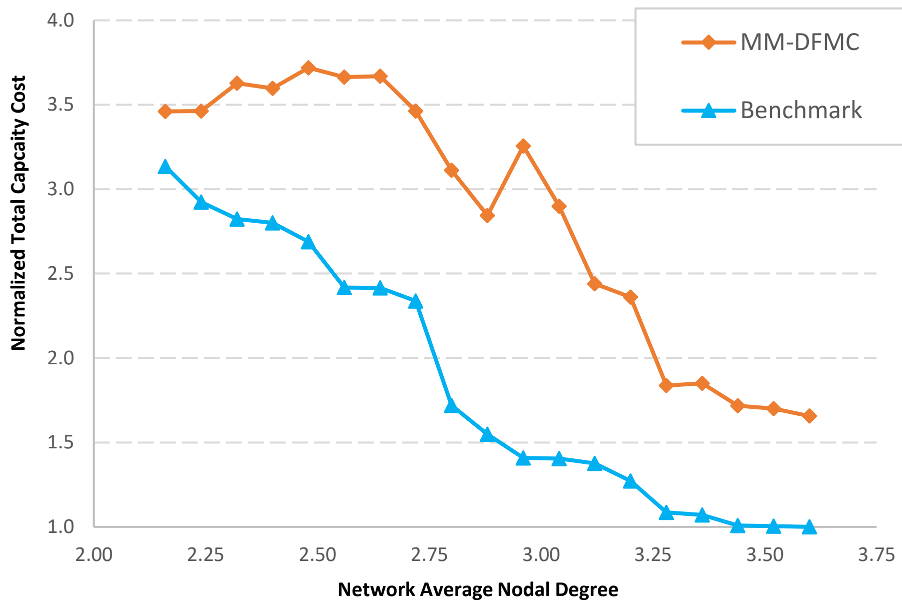


Figure 5.12 – Normalized MM-DFMC and Meta-Mesh total capacity cost on the 25n50s1 network family

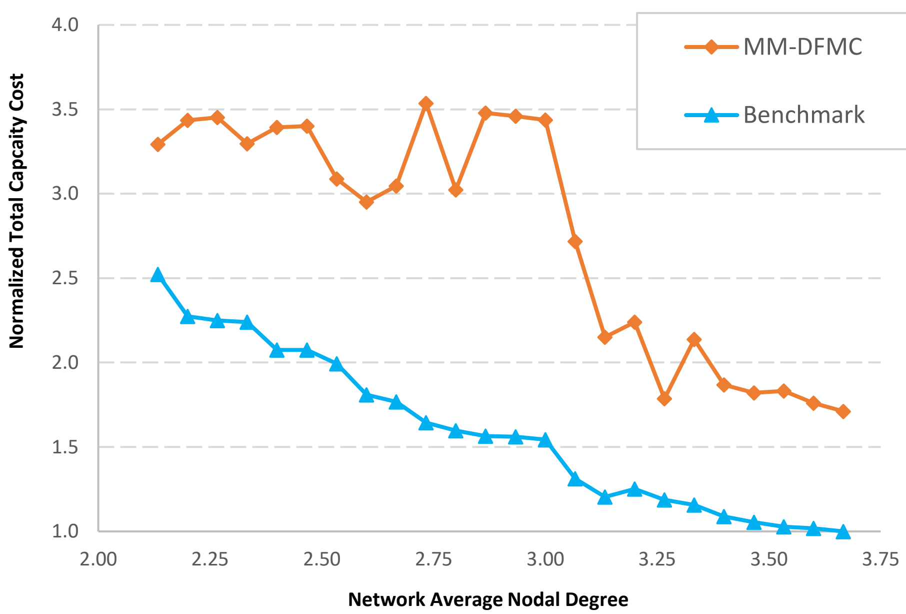


Figure 5.13 – Normalized MM-DFMC and Meta-Mesh total capacity cost on the 30n60s1 network family

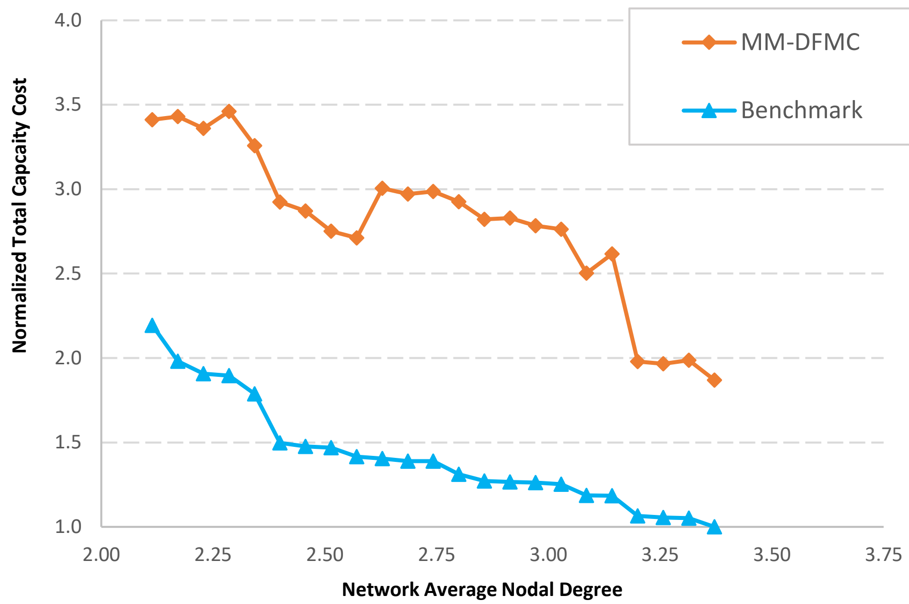


Figure 5.14 – Normalized MM-DFMC and Meta-Mesh total capacity costs on the 35n70s1 network family

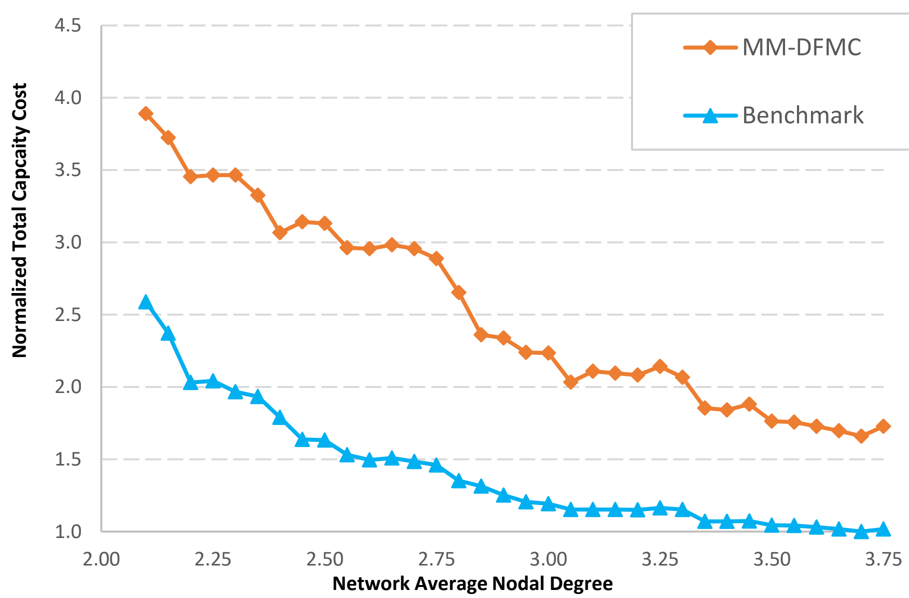


Figure 5.15 – Normalized MM-DFMC and Meta-Mesh total capacity costs on the 40n80s1 network family

Table 5.2 – MM-DFMC experiments results

Network Family	Test Network	Total Spare Capacity			Total Capacity (working and spare)		
		R_1 = 100% (Meta-Mesh)	R_2 = 100% (MM-DFMC)	Capacity increase (R_2/R_1)	R_1 = 100% (Meta-Mesh)	R_2 = 100% (MM-DFMC)	Capacity increase (R_2/R_1)
15n30s1	15n16s	322431	532277	**1.65**	635373	854838	**1.34**
	15n17s	240129	613398	2.55	533064	912572	1.71
	15n18s	217274	624719	2.87	498715	912181	1.82
	15n19s	202369	652284	3.22	476156	918565	1.92
	15n20s	211214	614090	2.90	471070	876711	1.86
	15n21s	184831	540161	2.92	426465	790108	1.85
	15n22s	175337	540917	3.08	415419	788689	1.89
	15n23s	143871	509967	**3.54**	384224	761927	1.93
	15n24s	145415	509636	3.50	388944	764309	**1.96**
20n40s1	20n21s	659820	1039810	**1.57**	1261820	1679390	**1.33**
	20n22s	607481	1045140	1.72	1129330	1682610	1.48
	20n23s	565727	1084690	1.91	1068634	1703290	1.59
	20n24s	550355	1144160	2.07	1044520	1719540	1.64
	20n25s	474039	1049580	2.21	938319	1565540	1.66
	20n26s	477064	976189	2.04	936023	1514780	1.61
	20n27s	457935	915193	1.99	890148	1377160	1.54
	20n28s	402998	975262	2.42	818978	1431380	1.74
	20n29s	394267	996606	2.52	814025	1443890	1.77
	20n30s	396729	912811	2.30	794262	1330490	1.67
	20n31s	374435	765329	2.04	753587	1169800	1.55
	20n32s	338779	837828	2.47	719894	1225700	1.70
	20n33s	309657	760514	2.45	676868	1143090	1.68
	20n34s	270690	694658	2.56	635546	1079960	1.69
	20n35s	228788	699483	**3.05**	606426	1099210	**1.81**
25n50s1	25n27s	1377850	1645470	**1.19**	2710180	2991190	**1.10**
	25n28s	1298070	1703370	1.31	2526945	2992920	1.18
	25n29s	1251760	1839260	1.46	2440820	3135950	1.28
	25n30s	1241660	1873250	1.50	2421100	3109750	1.28
	25n31s	1184120	1999290	1.68	2323690	3215030	1.38
	25n32s	1058210	2050340	1.93	2089100	3167390	1.51
	25n33s	1057910	2022740	1.91	2088040	3172860	1.51
	25n34s	1010650	1918850	1.89	2020540	2992760	1.48
	25n35s	671861	1815090	2.70	1486130	2689670	1.80
	25n36s	589865	1659630	2.81	1339390	2459160	1.83
	25n37s	502190	1984890	**3.95**	1217090	2814100	**2.31**
	25n38s	492439	1695900	3.44	1213720	2505650	2.06
	25n39s	472126	1348410	2.85	1189570	2109060	1.77
	25n40s	413543	1331130	3.21	1098750	2040450	1.85
	25n41s	321581	933409	2.90	938845	1588420	1.69
	25n42s	322639	958575	2.97	925803	1599020	1.72

	25n43s	278972	873121	3.12	871812	1484410	1.70
	25n44s	278964	865993	3.10	868539	1470310	1.69
	25n45s	285644	840236	2.94	864476	1431720	1.65
30n60s1	30n32s	1726940	2705010	**1.56**	3444640	4495980	**1.30**
	30n33s	1570960	3084960	1.96	3104900	4690880	1.51
	30n34s	1564310	3105810	1.98	3072570	4714370	1.53
	30n35s	1553440	2914150	1.87	3059160	4500820	1.47
	30n36s	1439590	3140050	2.18	2834720	4632830	1.63
	30n37s	1442790	3100730	2.14	2834290	4645480	1.63
	30n38s	1406910	2719950	1.93	2721380	4616600	1.54
	30n39s	1245880	2703240	2.16	2471860	4031620	1.63
	30n40s	1220070	2856380	2.34	2413810	4158850	1.72
	30n41s	1112320	3536920	3.17	2245560	4827190	2.14
	30n42s	1057020	2894450	2.73	2181510	4129310	1.89
	30n43s	1024420	3490260	3.40	2136130	4750440	2.22
	30n44s	1018720	3468060	3.40	2131460	4725580	2.21
	30n45s	1000880	3427920	3.42	2107370	4693940	**2.22**
	30n46s	741579	2551040	**3.44**	1792810	3711400	2.07
	30n47s	601424	1894050	3.14	1643960	2936270	1.78
	30n48s	676867	2023990	2.99	1709000	3058480	1.78
	30n49s	618094	1909730	3.08	1621860	2440650	1.50
	30n50s	579461	1899150	3.27	1580250	2919640	1.84
	30n51s	536495	1554350	2.89	1487290	2551630	1.71
	30n52s	509480	1490530	2.92	1439100	2487170	1.72
	30n53s	488395	1525190	3.12	1403900	2501320	1.78
	30n54s	461897	1429280	3.09	1390210	2402650	1.72
	30n55s	451852	1385010	3.06	1365720	2337070	1.71
35n70s1	35n37s	2303890	4546800	**1.97**	4543120	7064920	**1.55**
	35n38s	1988830	4760730	2.39	4102560	7105990	1.73
	35n39s	1959050	4671280	2.38	3950110	6958220	1.76
	35n40s	2024560	4989490	2.46	3925830	7168380	1.82
	35n41s	1828930	4646310	2.54	3700310	6745980	1.82
	35n42s	1355630	4203640	3.10	3100130	6057130	1.95
	35n43s	1334760	4020600	3.01	3056500	5945800	1.94
	35n44s	1345700	3868370	2.87	3043650	5698870	1.87
	35n45s	1285650	3779130	2.93	2930560	5615540	1.91
	35n46s	1292350	4353710	3.36	2910490	6224230	2.13
	35n47s	1287280	4328590	3.36	2878820	6153800	2.13
	35n48s	1296460	4419180	3.40	2875600	6184970	2.15
	35n49s	1157360	4276280	3.69	2715260	6059120	2.23
	35n50s	1122980	4159900	3.70	2635190	5843140	2.21
	35n51s	1124300	4159840	**3.69**	2623090	5860260	**2.23**
	35n52s	1126710	4098050	3.63	2612220	5765460	2.20
	35n53s	1110770	4035840	3.63	2596160	5721960	2.20
	35n54s	1013300	3646160	3.59	2457220	5185020	2.11
	35n55s	1028980	3869480	3.76	2451880	5417780	2.20

	35n56s	878282	2701190	3.07	2203980	4100250	1.86
	35n57s	871059	2684890	3.08	2187470	4070690	1.86
	35n58s	864320	2729450	3.15	2177880	4116620	1.89
	35n59s	778674	2532260	3.25	2070960	3872120	1.86
40n80s	40n42s	3040200	5861130	**1.92**	5746450	8635710	**1.52**
	40n43s	2757510	5515040	2.00	5265600	8267220	1.57
	40n44s	2259370	5247180	2.32	4509700	7666260	1.69
	40n45s	2312470	5259120	2.27	4534620	7691000	1.69
	40n46s	2217430	5377170	2.42	4368630	7693030	1.76
	40n47s	2234890	5212970	2.33	4293130	7380650	1.71
	40n48s	2042260	4560160	2.23	3971890	6805990	1.71
	40n49s	1748800	4902450	2.80	3632200	6973330	1.91
	40n50s	1751820	4849570	2.76	3627080	6952060	1.91
	40n51s	1576460	4579070	2.90	3400090	6578410	1.93
	40n52s	1517700	4655570	**3.06**	3320560	6564890	1.97
	40n53s	1553300	4693540	3.02	3349570	6623850	1.97
	40n54s	1539230	4709850	3.05	3296340	6564840	**1.99**
	40n55s	1495000	4528770	3.02	3240000	6410610	1.97
	40n56s	1371620	4109270	2.99	2999270	5890380	1.96
	40n57s	1293840	3468810	2.68	2918090	5242950	1.79
	40n58s	1158340	3770400	3.25	2778450	5192170	1.86
	40n59s	1086300	3285220	3.02	2678200	4968650	1.85
	40n60s	1079360	3281790	3.04	2647730	4960660	1.87
	40n61s	1006680	2854310	2.83	2559860	4513290	1.76
	40n62s	1012710	3017320	2.97	2559880	4682930	1.82
	40n63s	1029440	3006020	2.92	2561050	4652700	1.81
	40n64s	1022770	2973890	2.90	2554220	4620380	1.80
	40n65s	1054540	3115250	2.95	2580780	4754150	1.84
	40n66s	1050140	3025600	2.88	2558640	4588730	1.79
	40n67s	905963	2582390	2.85	2376510	4114130	1.73
	40n68s	907011	2557640	2.81	2378110	4087680	1.71
	40n69s	902571	2622850	2.90	2381340	4174480	1.75
	40n70s	845813	2360520	2.79	2315960	3914440	1.69
	40n71s	852986	2359100	2.76	2310830	3897360	1.68
	40n72s	823026	2301820	2.79	2290400	3838380	1.67
	40n73s	803748	2246990	2.79	2257460	3766500	1.66
	40n74s	773797	2173170	2.80	2220040	3683860	1.65
	40n75s	820319	2298300	2.80	2259390	3835940	1.69

5.5.2 MM-DFMR RESULTS AND DISCUSSION

Results from the MM-DFMR formulation are presented and discussed in this section. The aim of this formulation is to minimize the total amount of non-restored working capacities over all dual failure situations in the meta-mesh design. Thus, by minimizing the number of non-restored working capacity we are maximizing the average dual failure restorability. This minimized amount is placed into the Equation (2.20) and the average dual-failure restorability of that specific network topology is obtained. After this we proceed to increase the investment in spare capacity starting from 0%, which corresponds to the investment for guarantying full single span failure restorability, and then we increased this amount by 5%, 10%, 20%, 30%, and 45%.

Figure 5.16 through Figure 5.21 illustrate the improvement of the average dual-failure restorability, R_2, as the budget (B) for allocating spare capacity in the network is increased. Data points are separated into individual curves for each member of the network family. Results for the network families of 15n30s1 and 20n40s1 took less than one minute. However, for the most richly connected networks from the rest of the families it took several minutes and sometimes hours to solve. It is important to emphasize that each network test that presented any unfeasibility issue previously described was adapted to this model as well, that is any network disconnection and/or any requirement in the increase of the number of eligible working and restoration routes. Although this MM-DFMR formulation model does not require that the test networks do not contain any degree-2 nodes cut, we adapted the model so that the results obtained here will match with our previous MM-DFMC results and therefore the price of achieving $R_2(i, j) = 1$. Otherwise, the results would not reach an unity limit.

Figure 5.16 illustrates the 15-node test network family. This network family, with no additional budget for spare capacity other than single-failure restorability, the average dual-failure restorability ranged from 0.48 up to 0.71. In other words, a networkwide average of 48% to 71% of working capacity is restorable in the event of any dual-failure scenario. Note that in almost all of the networks inside this family we can achieve an average of 95% dual failure restorability with an increase of 45% of the spare capacity cost. In addition, it is important to remember that dual failure scenarios that were inherently not survivable (e.g., they resulted in a disconnected network) were not included in those calculation.

Likewise, for the 20-node network family, the average R_2 ranged from 52% to 85%. As opposed to the previous network family, with an increase of only 20% (it was 45% on the previous) on the spare capacity we can achieve an average of 95% dual-failure restorability on the entire network family. These results are quite significant and demonstrate a high average dual-failure restorability in meta-mesh networks. In the same manner, the 25-node network family achieved an average dual failure restorability that ranged from 66% to 93% with zero percent of increase on the single-failure restorability budget. Remarkable enough, in the 25n27s test network we can ensure 93% R_2 with no more than the spare capacity investment for single failure restorability. However, we stress that such extreme cases are because the fact that we do not consider dual failure scenarios that isolate nodes within chains. As in the previous network family, we can achieve an average of 96% dual-failure restorability on the entire 25n50s network family with an increase of 20% on spare capacity. Moving to the next family, we had that in the 30n60s the average dual failure restorability ranged from 51% to 84%. However, in this network family an average of 96% dual failure restorability was only achieved with an increase of 30% of the total spare capacity needed for full single-failure restorability. In the 35n70s network family the weighted average dual failure restorability ranged from 39% to 85% with a minimum amount of spare channel capacity sufficient to yield $R_1 = 1$. This particular 35-node and 37-span network was a quite surprising result as it was the network with the lowest average R_2. Nevertheless, adding 10% more spare capacity to this network can ensure a double amount, more specifically 80%, of dual failure restorability. Ultimately, with an increase of 20% in the total spare capacity, we can ensure an average of 95% full dual failure restorability on the entire network family. Finally, in the most richly connected network family 40n80s the weighted average R_2 ranged from 44% to 89%. In addition, with 20% more spare capacity we can reach an average of 95% R_2 on the entire network family. Again, we note that dual failure scenarios that were inherently not survivable (e.g., they resulted in a disconnected network) were not included in these calculations.

Table 5.3 shows the results obtained using the MM-DFMR model for each network test inside its corresponded network family. Again, inside each test case there is an initial budget limit, which corresponds to the minimum cost of placing spare capacity on a network designed for full single-failure restorability. This is the case of 0% capacity increase, where it can be interpreted as the solver trying to assign the “leftover” spare capacity of the meta-mesh single-failure restorable design. Thus, the percentages of increase in capacity budget used in this experiment was 0%, 5%, 10%, 20%, 30%, and 45%.

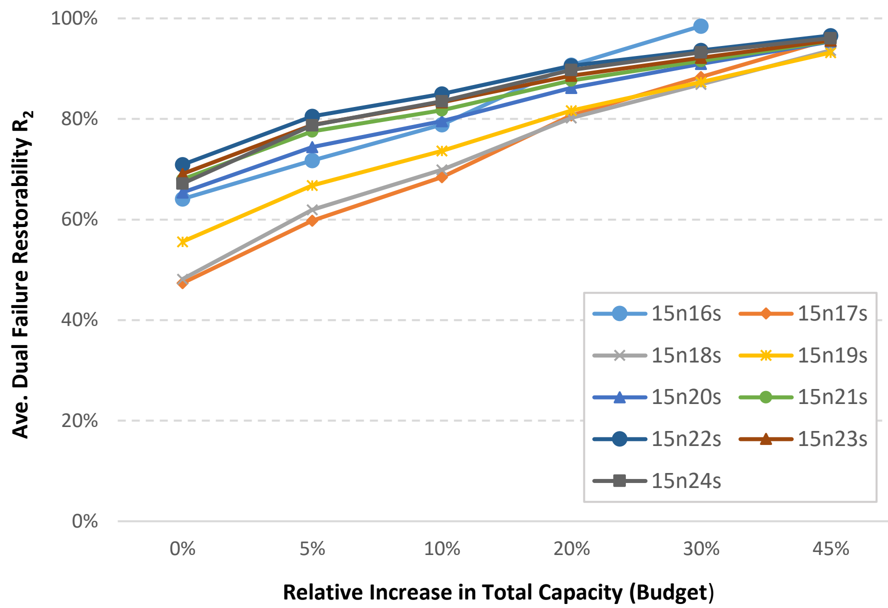


Figure 5.16 – Achievable R_2 vs. Percentage of spare capacity increase on the 15n30s1 network family

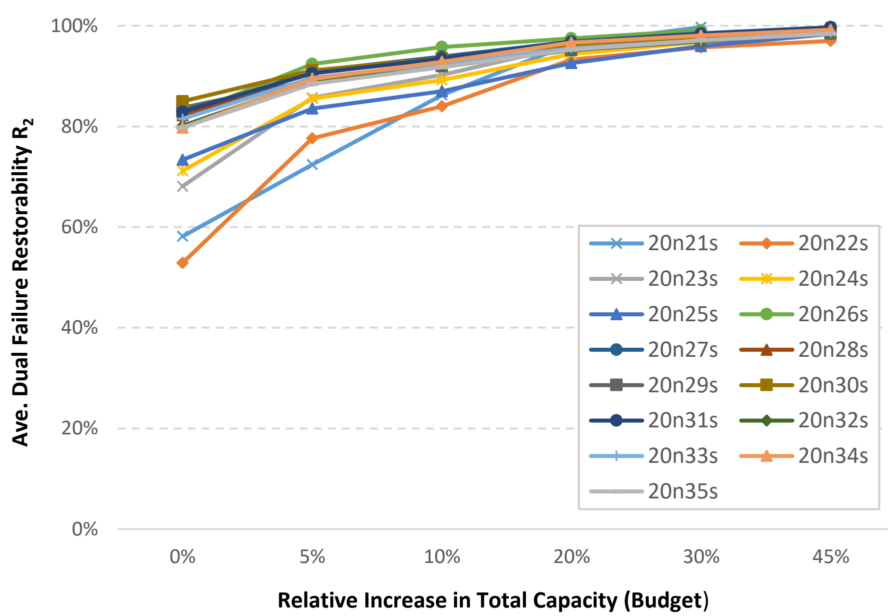


Figure 5.17 – Achievable R2 vs. Percentage of spare capacity increase on the 20n40s1 network family

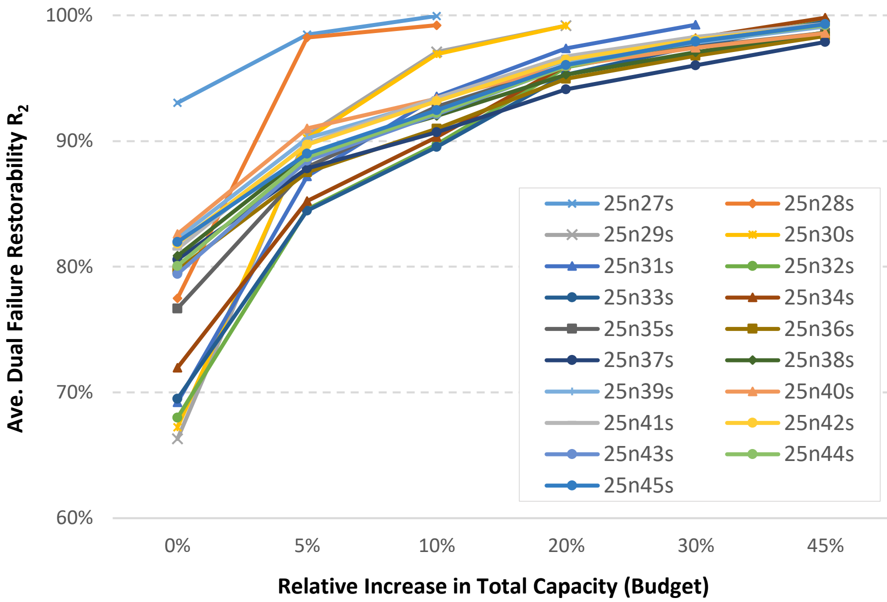


Figure 5.18 – Achievable R2 vs. Percentage of spare capacity increase on the 25n50s1 network family

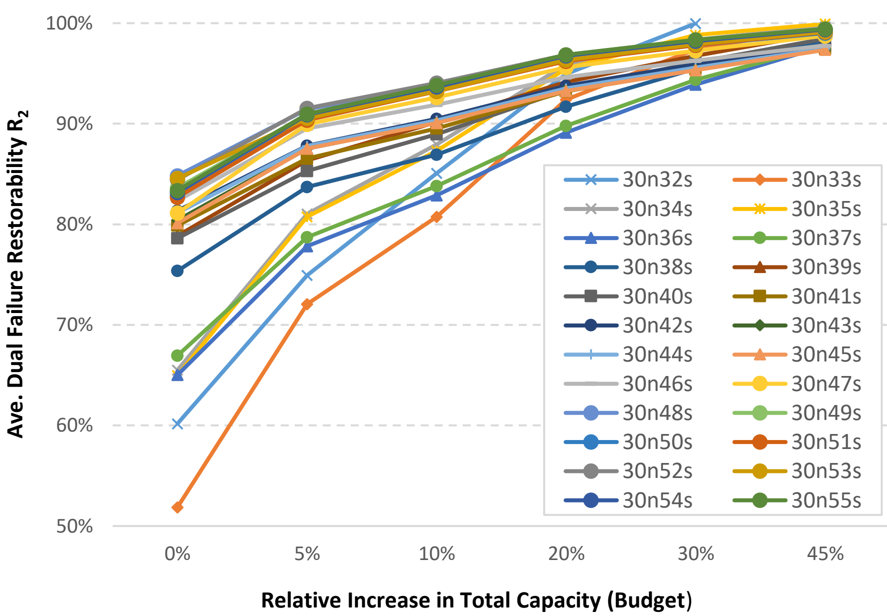


Figure 5.19 – Achievable R2 vs. Percentage of spare capacity increase on the 30n60s1 network family

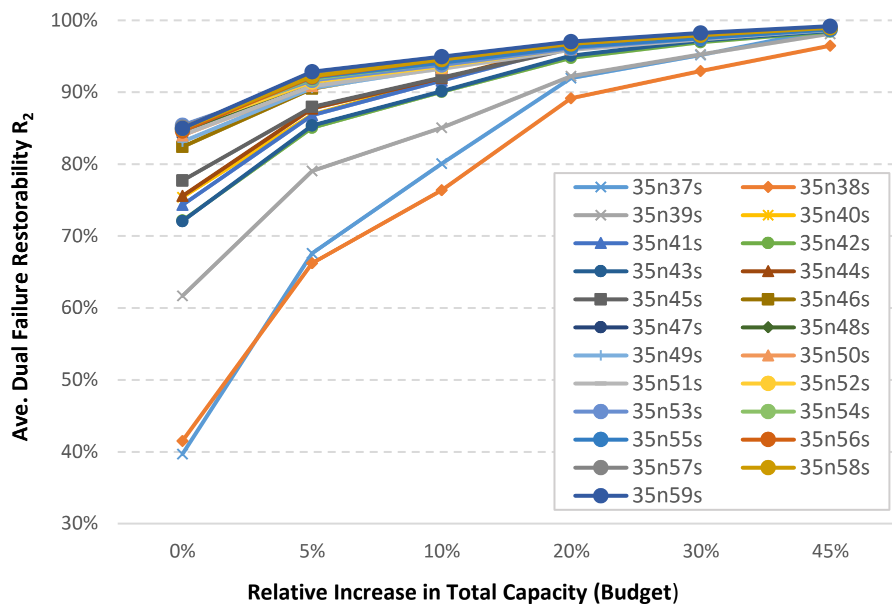


Figure 5.20 – Achievable R2 vs. Percentage of spare capacity increase on the 35n70s1 network family

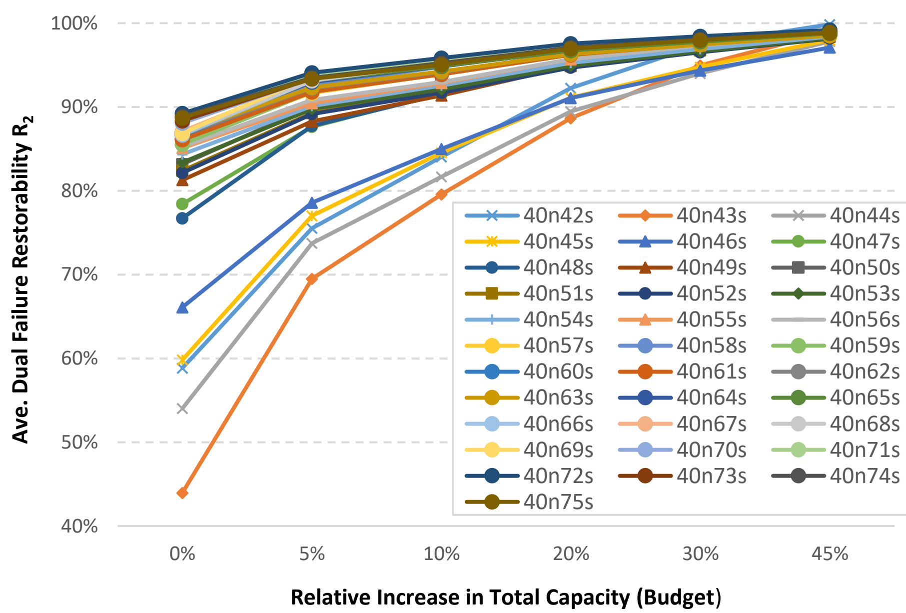


Figure 5.21 – Achievable R2 vs. Percentage of spare capacity increase on the 40n80s1 network family

Table 5.3 – MM-DFMR experiments results

Network Family	Test Network	***B*** **= 0%**		***B*** **= 5%**		***B*** **= 10%**		***B*** **= 20%**		***B*** **= 30%**		***B*** **= 45%**	
		N_2	R_2	N_2	R_2	N_2	R_2	N_2	R_2	N_2	R_2	N_2	R_2
15n30s1	15n16s	18690	0.640	14712	0.717	10996	0.789	4850	0.907	790	0.985		
	15n17s	23985	0.473	18324	0.598	14387	0.684	8792	0.807	5321	0.883	1873	0.959
	15n18s	23372	0.481	17154	0.620	13558	0.699	8937	0.802	5925	0.869	2862	0.937
	15n19s	20507	0.556	15336	0.668	12179	0.736	8480	0.816	5837	0.874	3135	0.932
	15n20s	16498	0.653	12177	0.744	9725	0.796	6581	0.862	4317	0.909	2191	0.954
	15n21s	16042	0.680	11270	0.775	9162	0.817	6200	0.876	4238	0.916	2251	0.955
	15n22s	15000	0.709	10035	0.805	7738	0.850	4870	0.906	3267	0.937	1758	0.966
	15n23s	15697	0.690	10759	0.788	8480	0.833	5776	0.886	3965	0.922	2232	0.956
	15n24s	18010	0.671	11663	0.787	9038	0.835	5619	0.898	3708	0.932	2154	0.961
20n40s1	20n21s	71666	0.581	47272	0.724	23624	0.862	6022	0.965	494	0.997		
	20n22s	70371	0.529	33381	0.776	23893	0.840	10020	0.933	6453	0.957	4491	0.970
	20n23s	48116	0.681	21520	0.857	14706	0.903	5842	0.961	2427	0.984	552	0.996
	20n24s	42463	0.712	21364	0.855	15882	0.892	8280	0.944	4718	0.968	1309	0.991
	20n25s	41231	0.734	25506	0.835	20115	0.870	11527	0.926	6278	0.959	2516	0.984
	20n26s	30225	0.821	18405	0.891	12844	0.924	7180	0.957	4234	0.975	1526	0.991
	20n27s	30822	0.822	15632	0.910	10604	0.939	5459	0.968	2609	0.985	530	0.997
	20n28s	30396	0.822	18046	0.894	13207	0.923	7768	0.955	4555	0.973	2119	0.988
	20n29s	27957	0.837	18474	0.892	13839	0.919	8316	0.951	5266	0.969	2595	0.985
	20n30s	26105	0.850	15400	0.911	11176	0.936	6489	0.963	4000	0.977	1591	0.991
	20n31s	29543	0.829	16334	0.906	11237	0.935	5688	0.967	2971	0.983	561	0.997
	20n32s	35242	0.801	19359	0.891	14282	0.919	8266	0.953	5296	0.970	2407	0.986
	20n33s	32094	0.815	17946	0.897	13114	0.924	7756	0.955	4989	0.971	2356	0.986
	20n34s	37216	0.797	19491	0.894	13107	0.929	6108	0.967	3539	0.981	1459	0.992
	20n35s	36457	0.798	20834	0.885	14873	0.918	8522	0.953	5184	0.971	2855	0.984
25n50s1	25n27s	35694	0.930	7863	0.985	261	0.999						
	25n28s	106912	0.775	8388	0.982	3728	0.992						
	25n29s	129089	0.663	36531	0.905	11150	0.971	3143	0.992				
	25n30s	129012	0.672	38539	0.902	12112	0.969	3204	0.992				
	25n31s	116900	0.692	48624	0.872	24533	0.935	9992	0.974	2841	0.993		
	25n32s	118809	0.680	57246	0.846	38078	0.897	15359	0.959	8336	0.978	1786	0.995
	25n33s	114956	0.695	58572	0.845	39440	0.895	17771	0.953	9137	0.976	1933	0.995
	25n34s	101753	0.720	53585	0.852	35166	0.903	14441	0.960	6625	0.982	684	0.998
	25n35s	85181	0.767	44080	0.879	26519	0.927	14496	0.960	9274	0.975	5096	0.986
	25n36s	77374	0.798	47690	0.875	34500	0.910	19306	0.949	12246	0.968	6278	0.984
	25n37s	73658	0.806	46189	0.878	35305	0.907	22372	0.941	15112	0.960	8041	0.979
	25n38s	74050	0.809	42632	0.890	31007	0.920	18276	0.953	11399	0.971	5272	0.986
	25n39s	70281	0.823	38745	0.902	26703	0.933	14372	0.964	8617	0.978	3567	0.991
	25n40s	69464	0.826	35937	0.910	26416	0.934	16133	0.960	10344	0.974	5665	0.986
	25n41s	68503	0.814	37695	0.898	24278	0.934	12037	0.967	6340	0.983	2488	0.993
	25n42s	68518	0.819	38857	0.897	25807	0.932	13315	0.965	7257	0.981	3036	0.992
	25n43s	81499	0.794	45776	0.884	31139	0.921	16006	0.960	8084	0.980	3106	0.992
	25n44s	81464	0.801	46105	0.887	31774	0.922	16540	0.960	8420	0.979	3173	0.992
	25n45s	75057	0.820	45778	0.890	31350	0.925	16451	0.960	8759	0.979	2923	0.993
30n60s1	30n32s	358614	0.602	225608	0.749	134212	0.851	45378	0.950	174			
	30n33s	320097	0.519	185808	0.721	127878	0.808	50439	0.924	17266	0.974	2251	0.997
	30n34s	252910	0.655	139071	0.810	88267	0.880	29956	0.959	12081	0.984	2444	0.997

	30n35s	259556	0.650	142539	0.808	94151	0.873	33496	0.955	8639	0.988	532	0.999
	30n36s	260300	0.650	165063	0.778	127261	0.829	80986	0.891	45319	0.939	16770	0.977
	30n37s	253900	0.669	163358	0.787	124299	0.838	78470	0.898	43317	0.944	16908	0.978
	30n38s	192561	0.754	127330	0.837	102183	0.869	64970	0.917	35374	0.955	13229	0.983
	30n39s	164526	0.789	106192	0.864	76398	0.902	45425	0.942	25248	0.968	7446	0.990
	30n40s	159490	0.786	109542	0.853	82396	0.890	49440	0.934	29231	0.961	11494	0.985
	30n41s	147278	0.799	98644	0.865	76665	0.895	49715	0.932	33686	0.954	18498	0.975
	30n42s	138717	0.814	90572	0.878	70685	0.905	46242	0.938	30253	0.959	16020	0.978
	30n43s	149804	0.804	95364	0.875	75285	0.902	50658	0.934	34938	0.954	20009	0.974
	30n44s	146856	0.811	95105	0.878	75161	0.903	50397	0.935	34488	0.956	19697	0.975
	30n45s	157573	0.801	98788	0.875	78732	0.901	53489	0.933	36937	0.953	21014	0.973
	30n46s	135479	0.824	80349	0.895	62183	0.919	41196	0.946	28643	0.963	16942	0.978
	30n47s	140098	0.811	74558	0.900	54886	0.926	32576	0.956	20170	0.973	9301	0.987
	30n48s	118221	0.848	66734	0.914	48291	0.938	27654	0.965	16456	0.979	7463	0.990
	30n49s	130035	0.836	72128	0.909	51854	0.935	29566	0.963	17173	0.978	7387	0.991
	30n50s	136628	0.829	74977	0.906	52154	0.935	29922	0.963	17540	0.978	7791	0.990
	30n51s	139556	0.827	77578	0.904	54504	0.932	30372	0.962	17551	0.978	7139	0.991
	30n52s	122769	0.844	66799	0.915	46781	0.941	24826	0.969	13621	0.983	5263	0.993
	30n53s	124794	0.846	76347	0.906	54482	0.933	29641	0.963	16996	0.979	6934	0.991
	30n54s	137636	0.832	74535	0.909	51665	0.937	26599	0.968	14445	0.982	5076	0.994
	30n55s	139220	0.834	75637	0.910	51582	0.938	26203	0.969	13785	0.984	4641	0.994
	35n37s	829372	0.397	445338	0.676	273920	0.801	110636	0.920	66100	0.952	16814	0.988
	35n38s	767096	0.415	443292	0.662	309946	0.764	142323	0.891	92562	0.929	46260	0.965
	35n39s	481900	0.617	263436	0.790	187805	0.851	97825	0.922	59605	0.953	23445	0.981
	35n40s	317754	0.754	169457	0.869	105274	0.918	42962	0.967	28473	0.978	15526	0.988
	35n41s	328966	0.743	169055	0.868	107492	0.916	47149	0.963	25640	0.980	11698	0.991
	35n42s	305630	0.721	163695	0.851	108969	0.901	57266	0.948	33379	0.970	17207	0.984
	35n43s	307020	0.721	160910	0.854	108277	0.901	54072	0.951	30878	0.972	16204	0.985
	35n44s	274294	0.756	137899	0.877	90207	0.920	40852	0.964	22873	0.980	9868	0.991
	35n45s	255623	0.777	138342	0.879	91165	0.921	45567	0.960	26432	0.977	13312	0.988
	35n46s	211133	0.824	114117	0.905	77251	0.936	44260	0.963	26930	0.978	12929	0.989
	35n47s	194555	0.840	108489	0.911	74404	0.939	42972	0.965	25934	0.979	11494	0.991
35n70s1	35n48s	195565	0.843	105101	0.916	72520	0.942	41743	0.967	25148	0.980	11221	0.991
	35n49s	208697	0.831	116280	0.906	82166	0.934	50114	0.959	32318	0.974	16770	0.986
	35n50s	199884	0.840	114436	0.908	83430	0.933	50058	0.960	31533	0.975	15676	0.987
	35n51s	205650	0.841	118076	0.908	86727	0.933	53044	0.959	33832	0.974	16750	0.987
	35n52s	198400	0.849	112483	0.915	83758	0.936	51420	0.961	32623	0.975	16145	0.988
	35n53s	193146	0.854	110581	0.916	82089	0.938	51272	0.961	33095	0.975	16653	0.987
	35n54s	199841	0.850	107801	0.919	78585	0.941	48672	0.963	31495	0.976	16193	0.988
	35n55s	204609	0.849	107475	0.921	79804	0.941	50053	0.963	33109	0.976	16939	0.987
	35n56s	212634	0.846	108879	0.921	75636	0.945	45387	0.967	28777	0.979	13723	0.990
	35n57s	208766	0.851	107700	0.923	74818	0.947	44333	0.968	27662	0.980	13194	0.991
	35n58s	214547	0.849	110834	0.922	77438	0.946	46613	0.967	29398	0.979	14781	0.990
	35n59s	209502	0.850	99792	0.928	70526	0.949	41039	0.971	24137	0.983	11264	0.992
	40n42s	945714	0.589	562776	0.755	366646	0.840	178176	0.922	58879	0.974	3096	0.999
	40n43s	1127330	0.440	613300	0.695	411232	0.796	228200	0.887	101145	0.950	20520	0.990
40n80s	40n44s	904547	0.540	516867	0.737	360470	0.817	207000	0.895	118143	0.940	42739	0.978
	40n45s	841339	0.598	481100	0.770	323458	0.846	186435	0.911	109679	0.948	41901	0.980
	40n46s	680158	0.661	429655	0.786	301113	0.850	178833	0.911	112925	0.944	58499	0.971
	40n47s	445427	0.784	254956	0.876	171580	0.917	87136	0.958	47859	0.977	19121	0.991

	40n48s	447295	0.767	235666	0.877	163183	0.915	87563	0.954	48703	0.975	16030	0.992
	40n49s	333635	0.813	209042	0.883	153395	0.914	91141	0.949	54692	0.969	23737	0.987
	40n50s	313551	0.832	185104	0.901	135232	0.927	81945	0.956	52005	0.972	24641	0.987
	40n51s	326308	0.824	198635	0.893	150048	0.919	93114	0.950	60111	0.968	30350	0.984
	40n52s	334196	0.822	202982	0.892	155170	0.917	99602	0.947	65270	0.965	34503	0.982
	40n53s	326249	0.834	200966	0.898	154804	0.921	99307	0.949	65770	0.967	34940	0.982
	40n54s	309034	0.844	195297	0.901	146942	0.926	93116	0.953	61097	0.969	31801	0.984
	40n55s	296652	0.851	189711	0.905	141810	0.929	85697	0.957	52450	0.974	27217	0.986
	40n56s	293861	0.853	182979	0.908	139914	0.930	85959	0.957	53090	0.973	26546	0.987
	40n57s	269347	0.862	162466	0.917	118410	0.939	67378	0.966	39357	0.980	18627	0.990
	40n58s	203255	0.866	111872	0.926	78357	0.948	46150	0.970	29324	0.981	15363	0.990
	40n59s	216886	0.856	123319	0.918	88597	0.941	54714	0.964	36567	0.976	20977	0.986
	40n60s	208371	0.862	121613	0.919	89938	0.940	54958	0.964	35951	0.976	19813	0.987
	40n61s	214650	0.861	126491	0.918	94197	0.939	58386	0.962	38618	0.975	21287	0.986
	40n62s	207672	0.867	120660	0.923	90940	0.942	56780	0.964	37669	0.976	20615	0.987
	40n63s	207713	0.869	121704	0.923	91637	0.942	58708	0.963	40432	0.975	23422	0.985
	40n64s	204877	0.870	110556	0.930	81123	0.949	50972	0.968	33990	0.978	18447	0.988
	40n65s	209111	0.870	110585	0.931	82472	0.949	52430	0.967	35497	0.978	19350	0.988
	40n66s	217594	0.867	112554	0.931	78591	0.952	47930	0.971	31088	0.981	16371	0.990
	40n67s	213572	0.871	111319	0.933	77794	0.953	46718	0.972	30070	0.982	15821	0.990
	40n68s	219382	0.869	114401	0.932	80337	0.952	48964	0.971	31807	0.981	17392	0.990
	40n69s	214376	0.869	103525	0.937	73986	0.955	43998	0.973	26704	0.984	13894	0.992
	40n70s	217773	0.881	120041	0.934	82756	0.955	48632	0.973	31288	0.983	16958	0.991
	40n71s	201543	0.891	115025	0.938	80140	0.957	47343	0.974	30221	0.984	15726	0.991
	40n72s	203255	0.892	111872	0.941	78357	0.958	46150	0.976	29324	0.984	15363	0.992
	40n73s	216886	0.884	123319	0.934	88597	0.952	54714	0.971	36567	0.980	20977	0.989
	40n74s	208371	0.888	121613	0.935	89938	0.952	54958	0.971	35951	0.981	19813	0.989
	40n75s	214650	0.887	126491	0.934	94197	0.951	58386	0.969	38618	0.980	21287	0.989

CHAPTER 6 CONCLUSION AND DISCUSSION

6.1 SUMMARY OF THESIS

The main objective of this research thesis is to provide different methods in which the basic formulation of the span-restorable meta-mesh design can be extended to design a network capable of achieving higher availability through strategies for control R_2. To achieve the objective, we developed two ILP design models that (1) provide the minimum total working and spare capacity costs to design a meta-mesh network capable of withstanding all dual-failures scenarios, except for those situations where a node is isolated; and (2) provide a maximization of the dual-failure restorability by minimizing the number of non-restored working capacities over all dual failures scenarios. Nevertheless, we found that, because of the very sparse nature of the networks studied, a significant number of dual-failure scenarios are inherently non-restorable due to they disconnect the network. This problem was overcome by not allowing a dual failure situation where a network could be disconnected. In addition, an unfeasibility arose when the solver could not found any working and/or restoration route to yield a feasible solution. This was overcome by increasing the number of eligible working routes for the problematic demand or the number of eligible restoration routes for the problematic failure scenario, as the case may be, until the infeasibility is repaired.

A brief introduction to transport networks and thesis outline was described in Chapter 1. Chapter 2 provided a background on mesh network survivability mechanisms, span restoration design model, spare capacity allocation (SCA) problem, joint capacity allocation (JCA) problem, and more importantly, the meta-mesh design model. In addition, Chapter 3 presented our research goals, our network topology models, and the computational aspects.

In Chapter 4, we investigated the prior meta-mesh network topologies of [5] and proposed a new insight capable of achieving spare capacity reduction in some network test cases. In this manner, some chains subnetwork in the majority of the meta-mesh network topologies were not bypassed because of the existence of a single-span connecting the chain anchor nodes and therefore providing of a “short” route between them. Although the experienced spare capacity reduction was not enough to yield a significant saving in cost, experiment results showed that indeed a reduction in spare capacity is achievable thanks to the allocation of a new bypass span in those chains that where not bypassed.

In Chapter 5, we developed two new meta-mesh network ILP design models that firstly provided a minimum-cost meta-mesh network design that is fully restorable in the event of dual failures, and secondly provided a meta-mesh network design with the same capacity as one with full single-failure survivability but that maximized dual-failures restorability. Both ILP models were implemented in AMPL and solved by Gurobi optimizer 6.0.5 with the default optimality gap of 0.01%. First, we have shown that designing a meta-mesh network capable of providing dual failure restorability requires a significant addition in spare capacity relative to the single-failure case. Secondly, results from the second model shows that the meta-mesh network average dual failure restorability could be substantially increased with only small additional investment in spare capacity. For example, in the 25n50s1 test network family, we observed that by providing full dual-failure restoration (of the situations for which it is possible), spare capacity requirements range from 1.2 to 3.5 times as much as required for full single-failure restoration only. However, we also showed that even a meta-mesh network designed to be only single-failure restorable would exhibit a sizeable inherent degree of dual-failure restorability (66% to 94%), and that dual-failure restorability can be substantially increased with only small additional investment in spare capacity. In addition, the investment or budget requirement to reach a dual-failure restorability is consistent with the minimum-cost meta-mesh network design as we only considered tri-connected networks where all dual-failure scenarios can be survivable with sufficient spare capacity.

6.2 MAIN CONTRIBUTIONS

There are three main contributions of this thesis. These contributions are summarized as follows:

1. Chapter 4: Analyses the previous work of the meta-mesh design model and propose a new insight capable of reducing the spare capacity cost in some test networks cases.
2. Chapter 5: Develops a new ILP design model that ensures full dual-failure restorability in the meta-mesh restoration network.

3. Chapter 5: Develops a new ILP design model that maximize the achievable level of dual failure restorability subject to none or certain extra amount of spare capacity.

6.3 OTHER CONTRIBUTIONS

Besides the contributions listed in Section 6.2, one conference paper has been submitted and we are waiting for acceptance.

1. A. Castillo, J. Doucette, "Dual-Failure Availability Analysis of Meta-Mesh Networks," *Resilient Network Design and Modeling* (RNDM 2018), to be submitted: May 2018.

REFERENCE

[1] TIA, "TIA's 2016-2020 ICT Market Review and Forecast," 2016. [Online]. Available: http://www.tiaonline.org/resources/tias-2016-2020-ict-market-review-and-forecast. [Accessed: 07-Aug-2017]

[2] ITU, "ITU's 2017 ICT Facts and Figures," 2017. [Online]. Available: https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2017.pdf. [Accessed: 07-Aug-2017].

[3] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2016-2021," 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html. [Accessed: 07-Aug-2017].

[4] Deloitte, "Economic and social benefits of expanding internet access," 2014. [Online]. Available: https://www2.deloitte.com/content/dam/Deloitte/br/Documents/technology-media-telecommunications/ValorConectividade.pdf. [Accessed: 22-Aug-2017].

[5] J. Doucette, *Advances on Design and Analysis of Mesh-restorable Networks*, University of Alberta Ph.D. Thesis, Edmonton, Alberta, Canada, 2005.

[6] P. Tomsu, C. Schmutzer, *Next Generation Optical Networks: The Convergence of IP Intelligence and Optical Technologies*, Prentice Hall, Upper Saddle River, NJ, 2002.

[7] W. D. Grover, *Mesh-based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking*, Prentice Hall PTR, Upper Saddle River, NJ, 2004.

[8] LightWave, "Altice USA to light up 1 million homes in FTTH network," 2017. [Online]. Available: http://www.lightwaveonline.com/articles/2017/08/altice-to-light-up-one-million-homes-in-ftth-network.html. [Accessed: 30-Sep-2017].

[9] Ken-ichi Sato, *Advances in Transport Network Technologies: Photonic network, ATM, and SDH,* Artech House, Norwood MA, 1996.

[10] A. K. Pradhan, K. Das, T. De, "Multicast Traffic Grooming with Survivability in WDM Mesh Networks," *International Conference on Signal Processing and Integrated Networks (SPIN 2015),* pp. 1020-1025, Noida, India, February 2015.

[11] A. Bhattacharya, M. Agarwal, S. Tabassum, M. Chatterjee, "Resource Aware Traffic Grooming with Shared Protection at Connection in WDM Mesh Networks," *International Conference on Advances in Computing, Communications and Informatics (ICACCI 2015),* pp. 110-115, Kochi, India, August 2015.

[12] H. Alazemi, S. Sebbah, M. Nurujjaman, "Fast and Efficient Network Protection Method Using Path Pre-Cross-Connected Trails," *Journal of Optical Communications and Networking,* vol. 5. 12, pp. 1343-1352, 2013.

[13] B. Mukherjee, *Optical Communication Networks,* McGraw-Hill, New York, 1997.

[14] OFC, "Reality Check on 100G Metro Networks," 2015. [Online]. Available: http://www.ofcconference.org/en-us/home/about/ofc-blog/2015/june-2015/reality-check-on-100g-metro-networks/. [Accessed: 18-Sep-2017]

[15] B. Todd, *The Use of Demand-Wise Shared Protection in Creating Topology Optimized High Availability Networks*, University of Alberta MSc. Thesis, Edmonton, Canada, 2009.

[16] T.-H. Wu, "Emerging Technologies for Fiber Network Survivability," *IEEE Communication Magazine*, vol. 33, no. 2, pp. 58-74, Feb. 1995.

[17] J. Sosnosky, "Service applications for SONET DCS distributed restoration," *IEEE Journal on Selected Areas in Communications*, Vol. 12, No. 1, pp. 59-68, January 1994.

[18] A. Kasem, *Heuristic Approaches for Survivable Network Optimization,* University of Alberta Ph.D. Thesis, Edmonton, Alberta, Canada, 2015.

[19] F. Dikbiyik, B. Mukherjee, M. Tornatore, "Adaptive Time-and Location-Aware Routing in Telecom Mesh Networks," *Network, IET,* vol. 2, no. 1, pp. 19-29, 2013.

[20] B. Todd, J. Doucette, "Use of Network Families in Survivable Network Design and Optimization," *2008 IEEE International Conference on Communications (ICC 2008),* pp. 151-157, Beijing, China, May 2008.

[21] J. Doucette, W. D. Grover, "Shared-Risk Logical Span Groups in Span-Restorable Optical Networks: Analysis and Capacity Planning Model," *Photonic Network Communication*, vol. 9, no. 1, pp. 35-53, 2005.

[22] M. A. H. Clouqueur, *Availability of Service in Mesh-Restorable Transport Networks*, University of Alberta Ph.D. Thesis, Edmonton, Alberta, Canada, 2004.

[23] W. D. Grover, J. Doucette, "Design of a Meta-Mesh of Chain Subnetworks: Enhancing the Attractiveness of Mesh-Restorable WDM Networking on Low Connectivity Graphs," *IEEE Journal on Selected Areas in Communications*, vol. 20 no. 1, January 2002.

[24] A. Nafarieh, S. C. Sivakumar, W. Phillips, W. Robertson, "Memory-Aware SLA-Based Mechanism for Shared-Mesh WDM Networks," *3rd International Congress on Ultra modern Telecommunications and Control Systems and Workshops* (ICUMT 2011), pp. 1-8, Budapest, Hungary, October 2011.

[25] L. Song, B. Mukherjee, "New Approaches for Dynamic Routing with Availability Guarantee for Differentiated Services in Survivable Mesh Networks: The Roles of Primary-

Backup Link Sharing and Multiple Backup Paths," *Global Telecommunications Conference,* pp. 1-5, San Francisco, CA, USA, November-December 2006.

[26] M. Clouqueur, W. D. Grover, "Availability Analysis of Span-Restorable Mesh Networks," *IEEE Journal on Selected Areas in Communications,* vol. 20, no. 4, pp. 810-821, May 2002.

[27] M. Herzberg, S. J. Bye, A. Utano, "Hop-limit approach for spare-capacity assignment in survivable networks," *IEEE/ACM Transactions on Networking*, vol. 3, no. 6, pp. 775-784, 1995.

[28] A. M. C. A. Koster, A. Zymolka, M. Jäger, R. Hulsermann, C. Gerlach, "Demand-wise shared protection for meshed optical networks," *Proceeding of the Workshop on the Design of Reliable Communication Networks (DRCN '03),* pp. 85-92, Banff, Alberta, Canada, 2003.

[29] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing,* Kluwer Academic Publishers, 1998.

[30] D. Stamatelakis, W. D. Grover, "Theoretical Underpinnings for the Efficiency of Restorable Networks using Preconfigured Cycles ("p-cycles")," *IEEE Transactions on Communications,* vol. 48, no. 8, pp. 1262-1265, 2000.

[31] W. D. Grover, "Self-organizing Broad-band Transport Networks," *Proceedings of the IEEE: Special Issue on Communications in the 21st Century,* vol. 85, no. 10, pp. 1582-1611, October 1997.

[32] B. D. Venables, W. Grover, M. H. MacGregor, "Two strategies for spare capacity placement (SCP) in mesh restorable networks," *Proc. IEEE ICC'93*, pp. 267-271Geneva, May 1993.

[33] H. Sakauchi, Y. Okanoue, S. Hasegawa, "Spare-channel Design Schemes for Self-healing Networks," *IEICE Trans. Comm.,* vol. E75-B, no. 7, pp. 624-633, July 1992.

[34] R. R. Iraschko, M. H. MacGregor, W. D. Grover, "Optimal Capacity Placement for Path Restoration in STM or ATM Mesh-Survivable Networks," *IEEE/ACM Transaction on Networking,* Vol. 6, no. 3, pp. 325-336, June 1998.

[35] Y. Xiong, L. G. Mason, "Restoration Strategies and Spare Capacity Requirements in Self-healing ATM Networks," *IEEE/ACM Transactions on Networking,* Volume: 7 Issue: 1, Feb. 1999, pp. 98-110.

[36] J. Doucette, W. D. Grover, "Comparison of Mesh Protection and Restoration Schemes and the Dependency on Graph Connectivity," *Proc. 3rd International Workshop on the Design of Reliable Communication Networks (DRCN 2001)*, Budapest, Hungary, October 2001.

[37] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing,* Kluwer Academic Publishers, November 1998.

[38] C. Wang, G. Shen, B. Chen, L. Peng, “Protection Path-based Hitless Spectrum Defragmentation in Eastic Optical Networks: Shared Backup Path Protection,” Optical Fiber Communication Conference, 2015.

[39] J. Doucette, M. Clouqueur, G. W. Grover, “On the Availability and Capacity Requirement of Shared Backup Path Protection Mesh Networks,” *Optical Networks Magazine,* pp. 29-44, 2003.

[40] J. L. Kennington, M. W. Lewis, “The Path Restoration Version of the Spare Capacity Allocation Problem with Modularity Restrictions: Models, Algorithms, and an Empirical Analysis,” *INFORMS Journal on Computing,* vol. 13, no. 3, pp. 181-190, 2001.

[41] N. Spring, R. Mahajan, D. Wetherall, “Measuring ISP Topologies with Rocketfuel,” *SIGCOMM*, Pittsburgh, Pennsylvania, USA 2002.

[42] M. Clouqueur, W. D. Grover, “Mesh-restorable network with complete dual failure restorability and with selectively enhanced dual-failure restorability properties,” *OptiComm 2002*.

[43] R. D. Doverspike, B. Wilson, “Comparison of Capacity Efficiency of DCS Network Restoration Routing techniques,” *JNSM*, vol. 2, no. 2, pp. 95-123, 1994.

[45] CBC News, “Perfect storm of cable cuts let to Atlantic cell outage,” 2017. [Online]. Available: http://www.cbc.ca/news/canada/nova-scotia/bell-aliant-cable-cut-outages-cell-service-blackout-1.4248111 [Accessed: 30-Aug-2017].

[46] J. Lee and R. Ben-Natan, *Integrating Service Level Agreements*, Wiley Publishing, 2002.

[47] M. Modarres, M. Kaminsky, V. Krivstov, *Reliability Engineering and Risk Analysis*, New York, NY. Marcel Dekker, 1999.

[48] M. To, P. Neusy, “Unavailability analysis of long-haul networks,” *IEEE Journal of Selected Areas in Communications*, vol. 12, pp. 100-109, Jan. 1994.

[49] D. A. Schupke, “Multiple failure survivability in WDM networks with p-cycles,” IEEE International Symposium on Circuits and Systems (ISCAS 2003), vol. 3, pp. 866-869, Bangkok, Thailand, May 2003.

[50] W. D. Grover, J. Doucette, M. Clouqueur, D. Leung, D. Stamatelakis, “New Options and Insights for Survivable Transport Networks,” *IEEE Communications Magazine*, January 2002.

[51] J. Walrand, P. Varaiya, *High-Performance Communication Networks*, Second Edition, Morgan Kaufmann Publishers, 2000.

[52] Level 3 Communication Network Map. 2017. [Online]. Available: http://www.level3.com/~/media/files/maps/en-network-services-level-3-network-map.ashx [Accessed: 30-Nov-2017].

[53] D. Tipper, "Resilient Network Design: Challenges and Future Directions," *Telecommunication Systems*, vol. 56, no. 1, pp. 5-16, May 2014.

[54] P. Cholda, A. Mykkeltveit, B. E. Helvik, O. J. Wittner, A. Jajszczyk, "A survey of resilience differentiation frameworks in communication networks," IEEE Comm. Surveys and Tutorials, vol. 9, pp. 32-55. 2007.

[55] J. M. Simmons, *Optical Network Design and Planning*, Second Edition, Springer, 2014.

[56] W. D. Grover, J. Doucette, "Advances in Optical Network Design with p-Cycles: joint optimization and pre-selection of candidate p-cycle", IEEE, July 2002.

[57] University of Kansas "ResiliNets Topology Map Viewer," 2010. [Online]. Available: http://www.ittc.ku.edu/resilinets/maps/#. [Accessed: 10-Aug-2017]

[58] "United States Map" *Google maps*. 08-Aug-2017.

[59] R. Fourer, D. M. Gay, B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, Cengage Learning, Boston, MA, USA, 2003.

[60] Gurobi Optimization Inc., Gurobi Optimizer Reference Manual, 2015, available online: http://www.gurobi.com, retrieved on 5 August 2017.

[61] M. Herzberg, S. J. Bye, "An optimal spare-capacity assignment model for survivable networks with hop limits," Proceedings of IEEE Global Communications Conference (GlobeCom 1994), pp. 1601-1607, San Francisco, CA, December 1994.

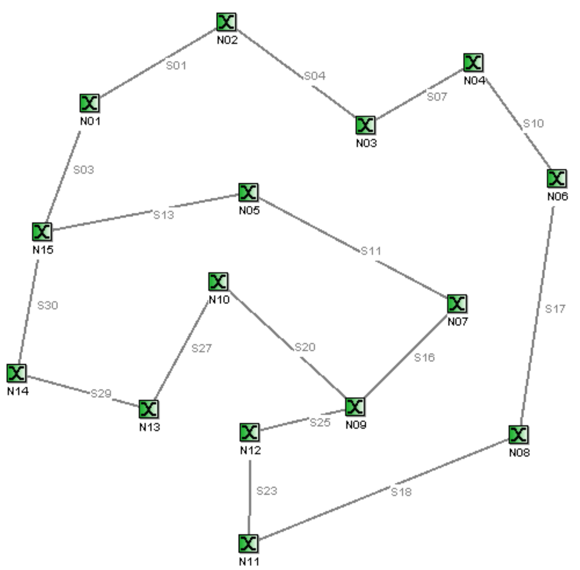
[62] J. Doucette, W. D. Grover, "Shared-Risk Logical Span Groups in Span-Restorable Optical Networks: Analysis and Capacity Planning Model," *Photonic Network Communications*, vol. 9, no. 1, pp. 35-53, 2005.

[63] F. Ramback, B. Konrad, L. Dembeck, U. Gebhard, M. Gunkel, M. Quagliotty, L. Serra, V. Lopez, "A multilayer cost model for metro/core networks," *J. Opt. Commun. Netw*, pp. 210-225 March 2013.

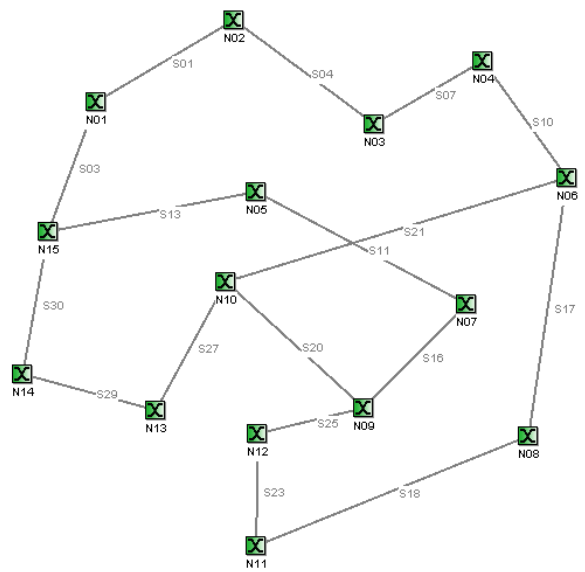
APPENDIX A
NETWORK TOPOLOGIES

A.1 15N30S1 MASTER NETWORK

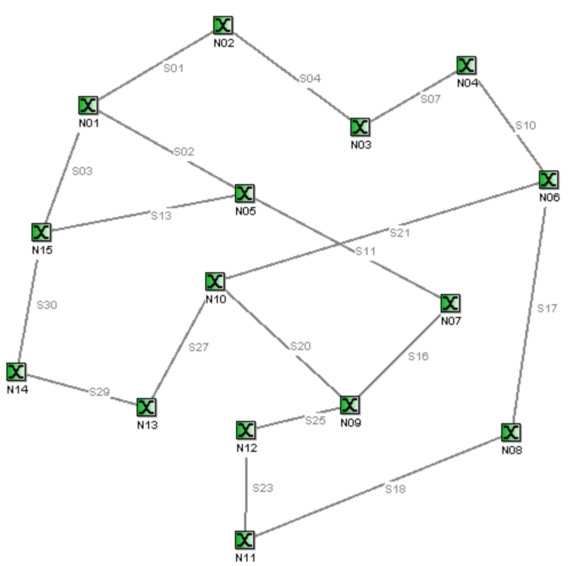
15n30s1 – 16s



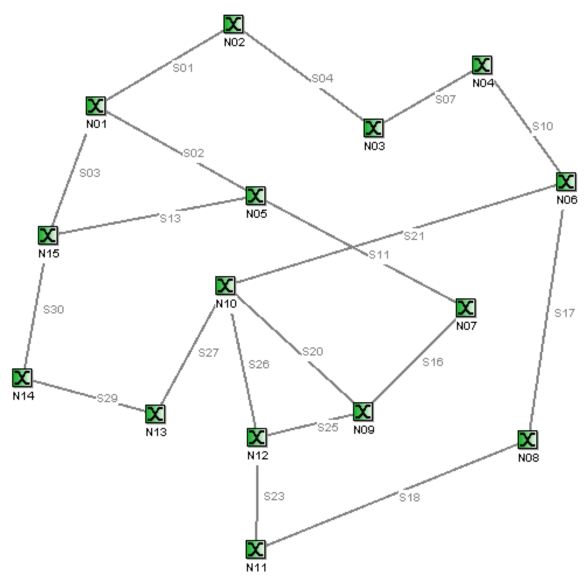
15n30s1 – 17s



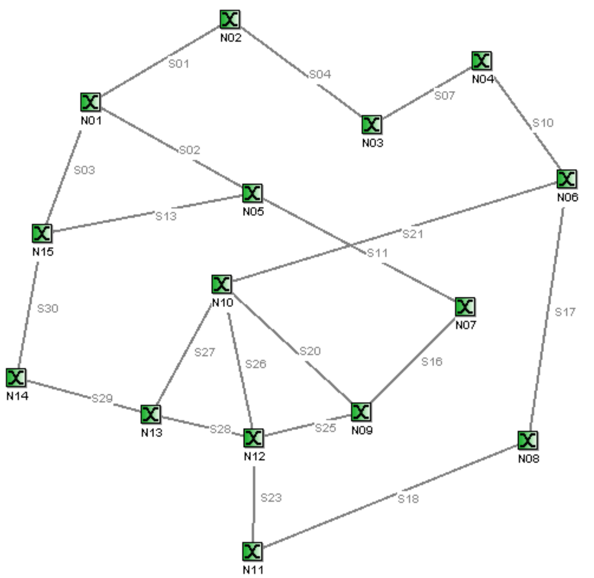
15n30s1 – 18s



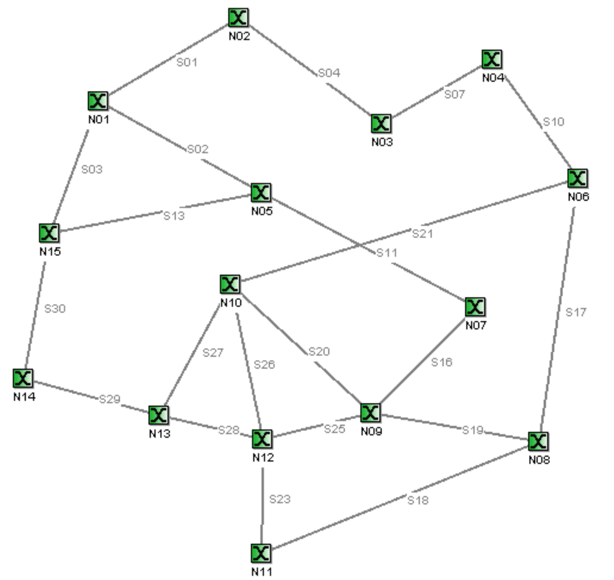
15n30s1 – 19s



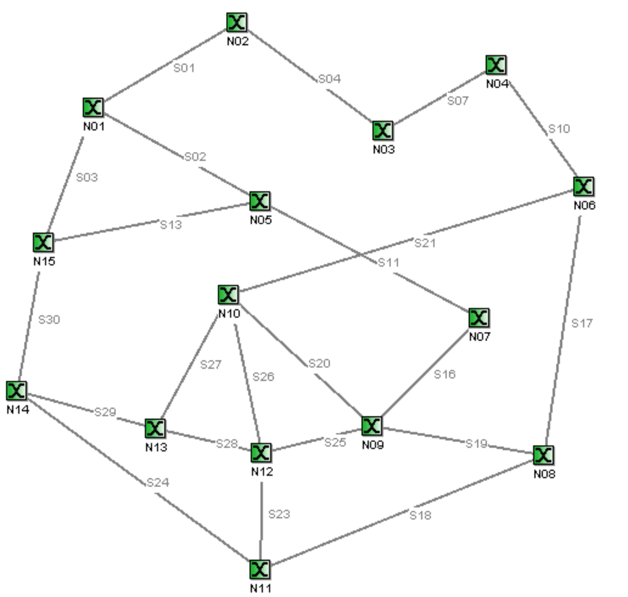
15n30s1 – 20s



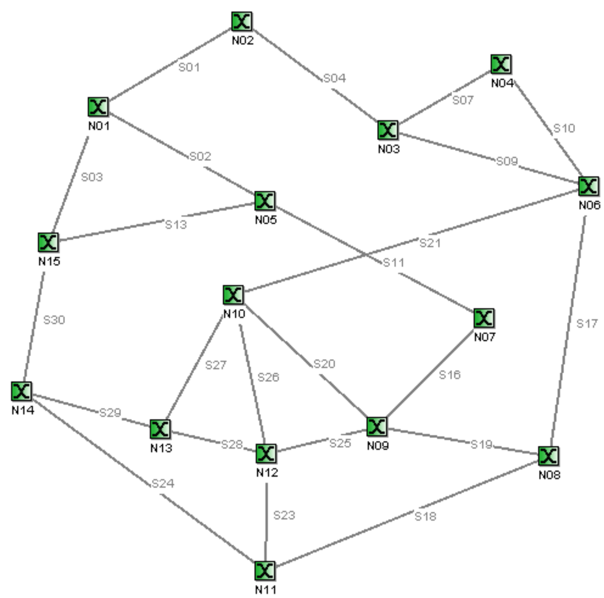
15n30s1 – 21s



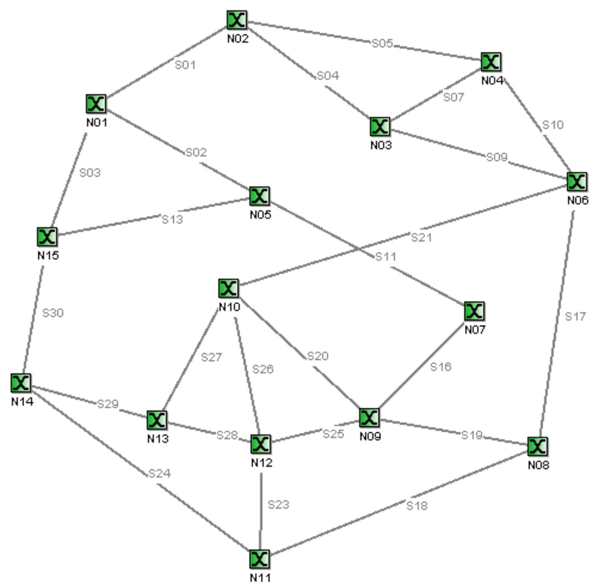
15n30s1 – 22s



15n30s1 – 23s



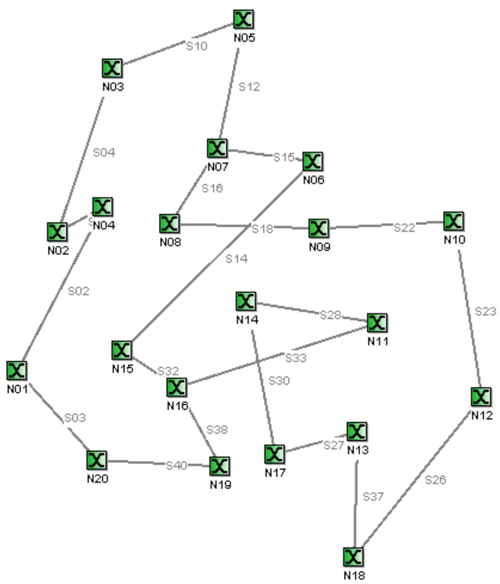
15n30s1 – 24s



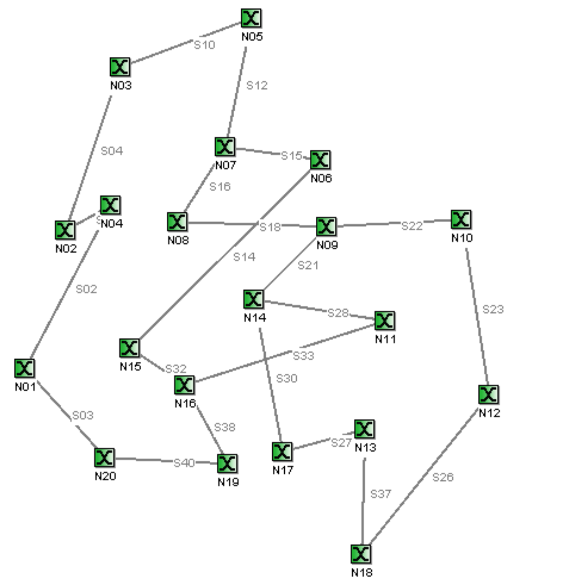
N01
N02
N03
N04
N05
N06
N07
N08
N09
N10
N11
N12
N13
N14
N15
S01
S02
S03
S04
S05
S07
S09
S10
S11
S13
S16
S17
S18
S19
S20
S21
S23
S24
S25
S26
S27
S28
S29
S30

A.2 20N40S1 MASTER NETWORK

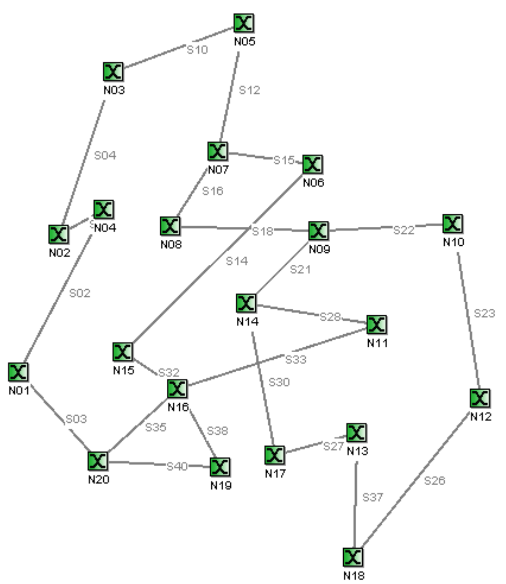
20n40s1 – 21s



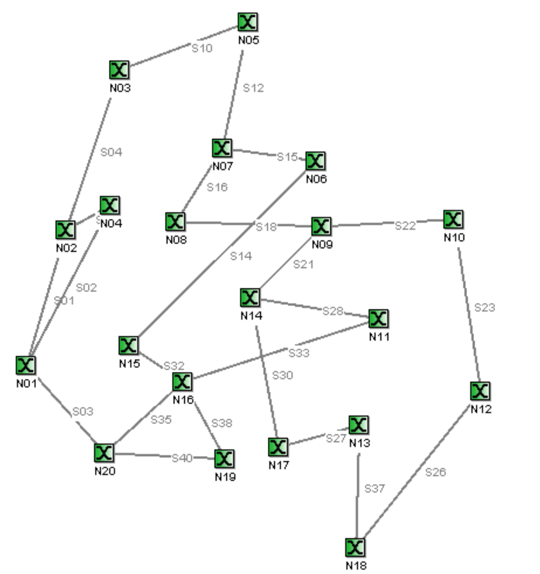
20n40s1 – 22s



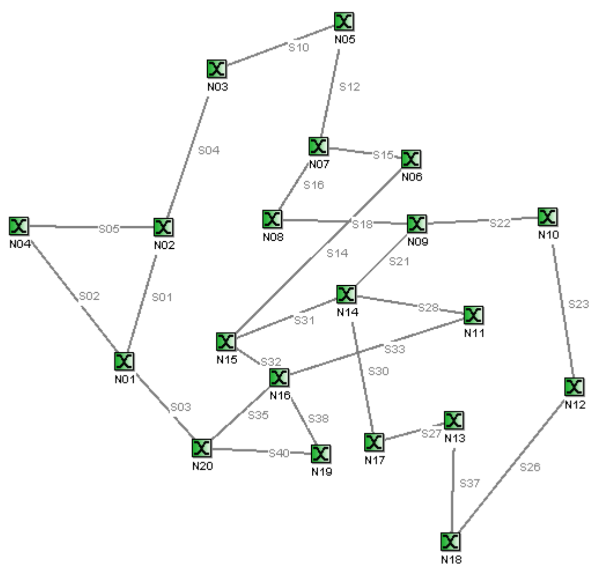
20n40s1 – 23s



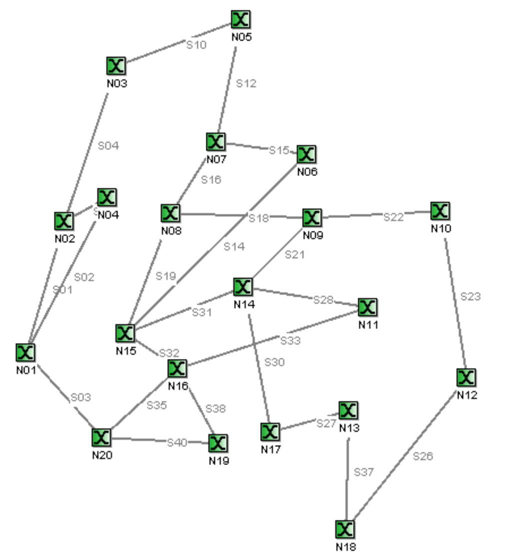
20n40s1 – 24s



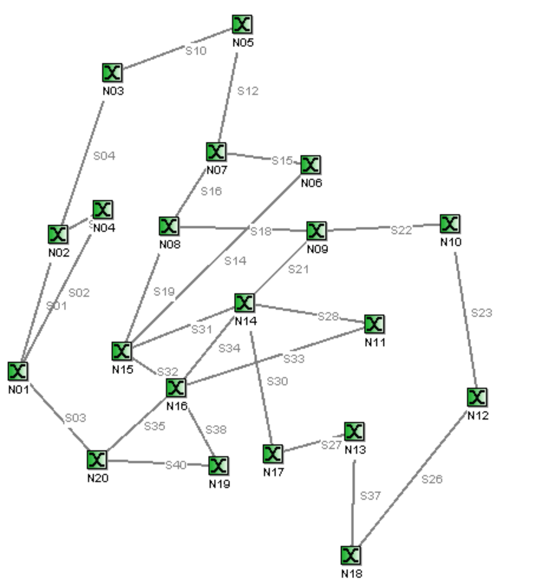
20n40s1 – 25s



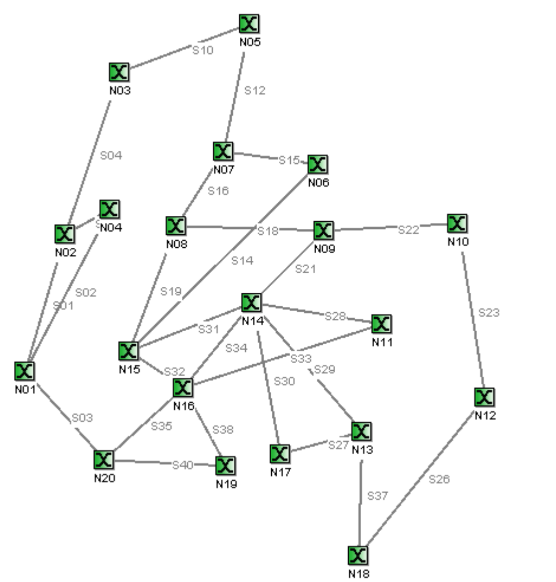
20n40s1 – 26s



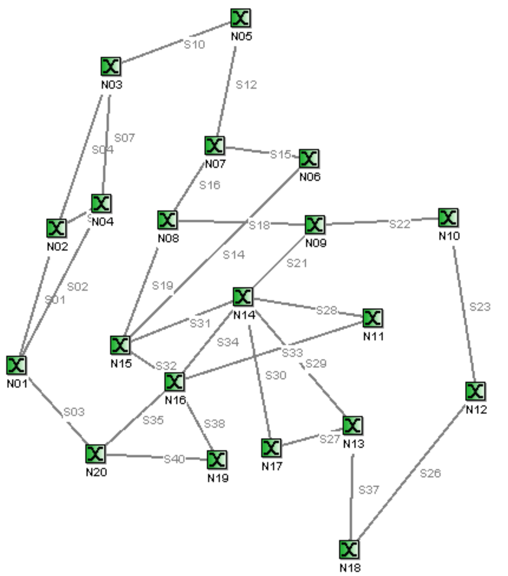
20n40s1 – 27s



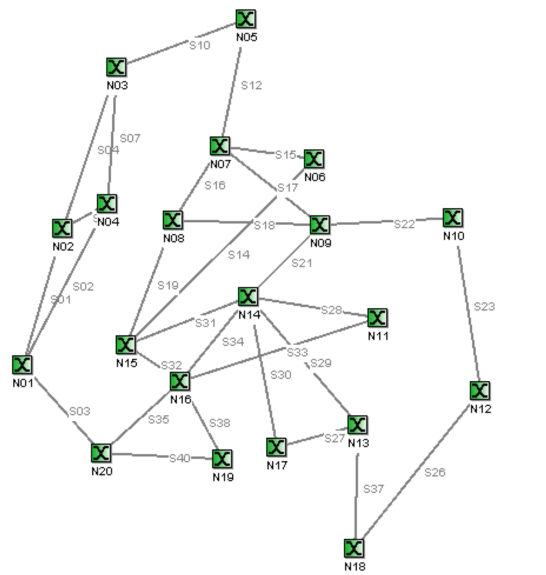
20n40s1 – 28s



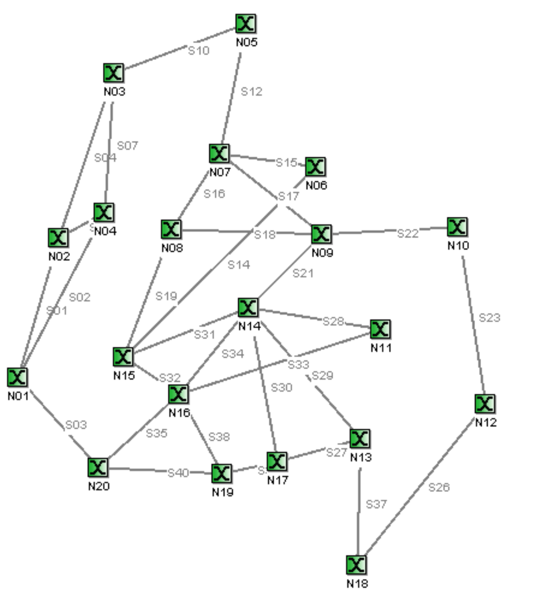
20n40s1 – 29s



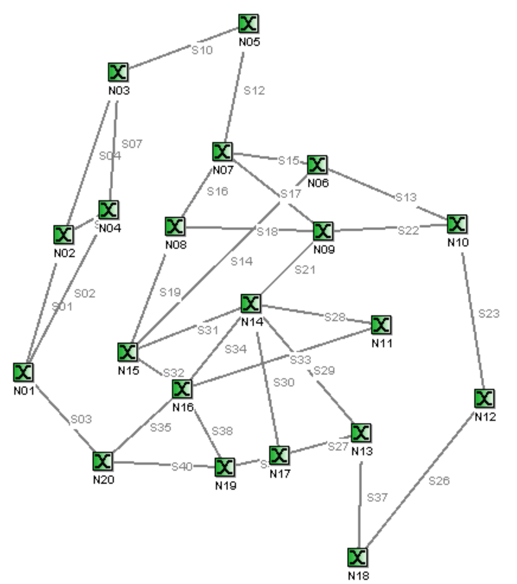
20n40s1 – 30s



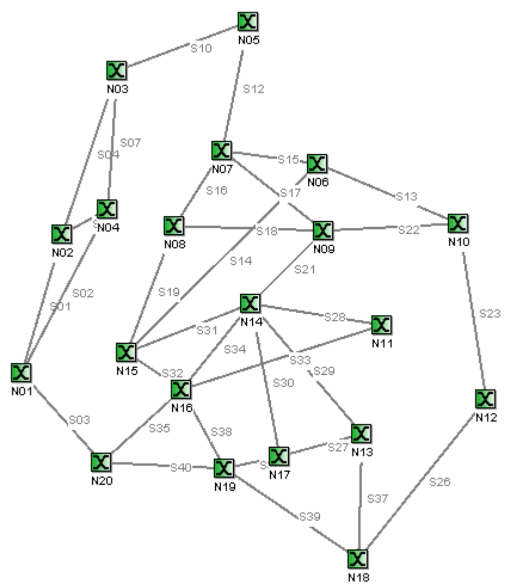
20n40s1 – 31s



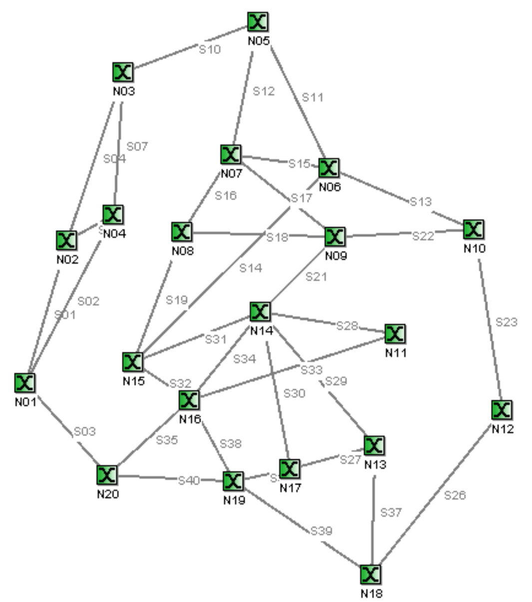
20n40s1 – 32s



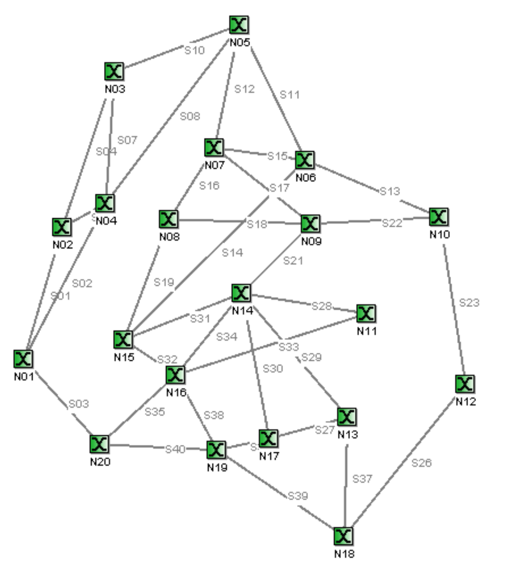
20n40s1 – 33s



20n40s1 – 34s

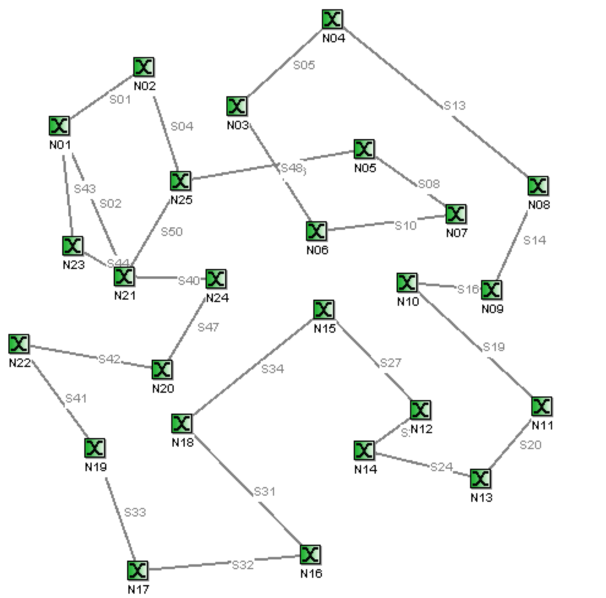


20n40s1 – 35s

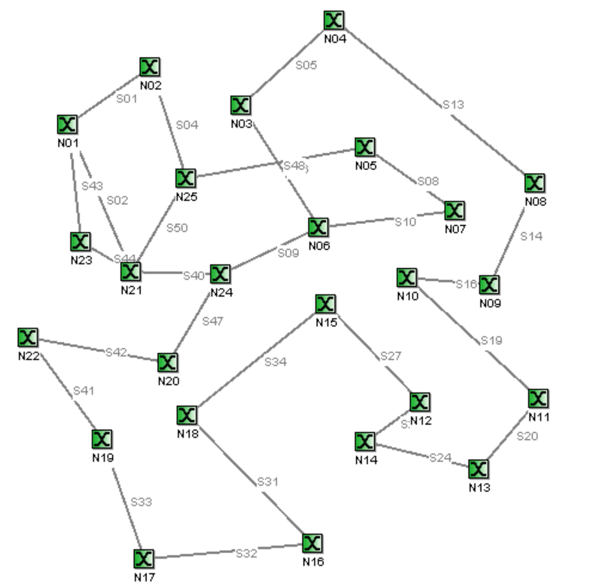


A.3 25N50S1 MASTER NETWORK

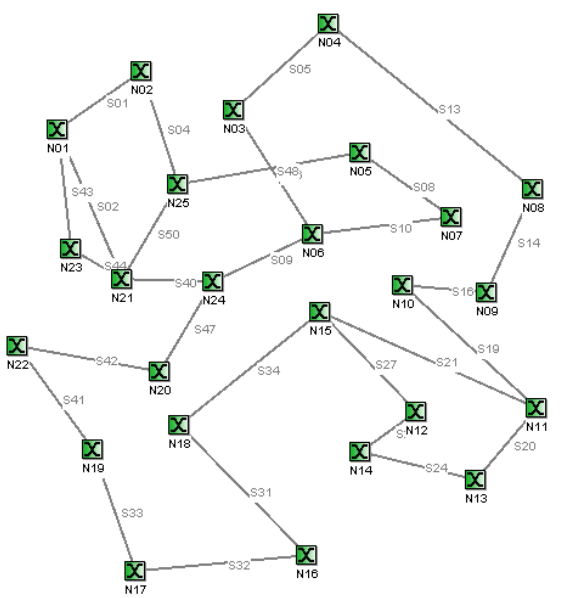
25n50s1 – 27s



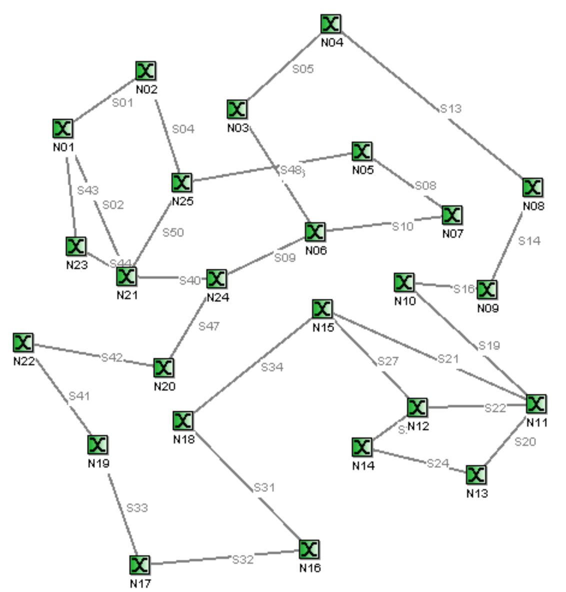
25n50s1 – 28s



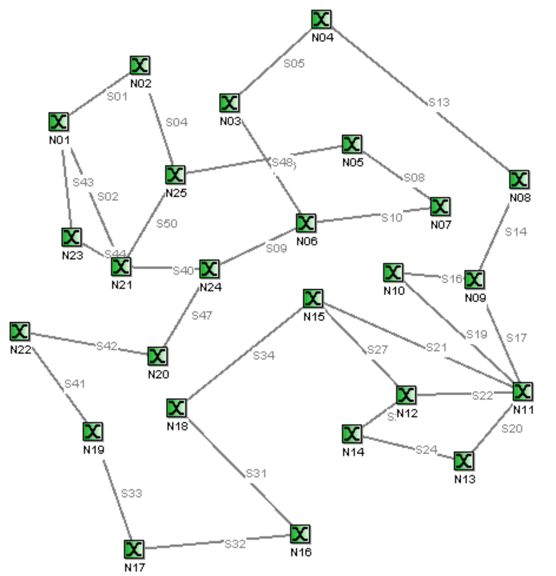
25n50s1 – 29s



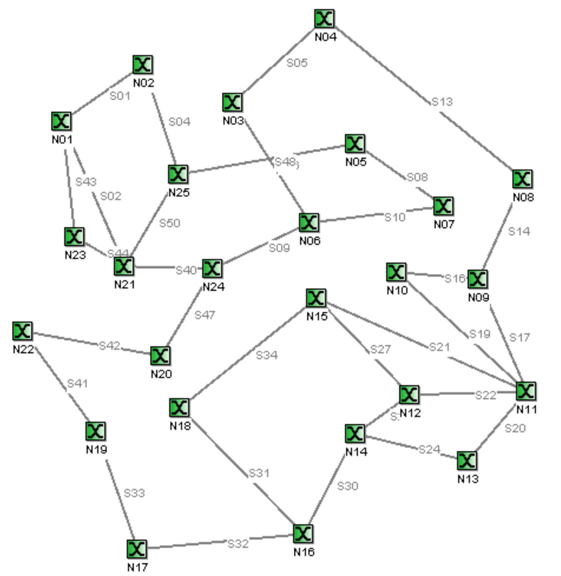
25n50s1 – 30s



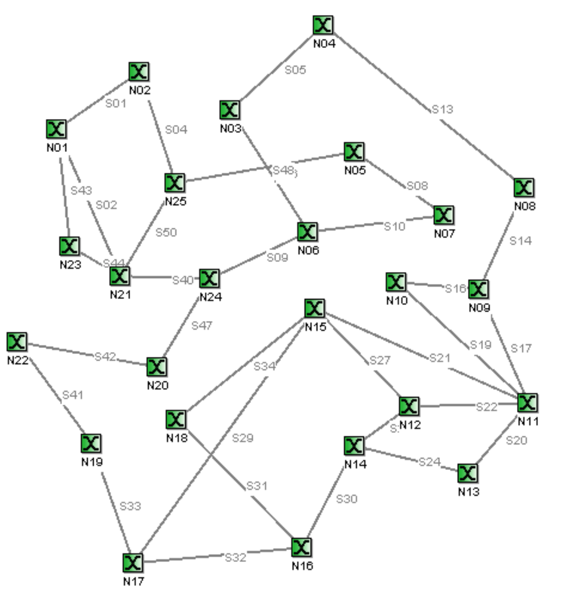
25n50s1 – 31s



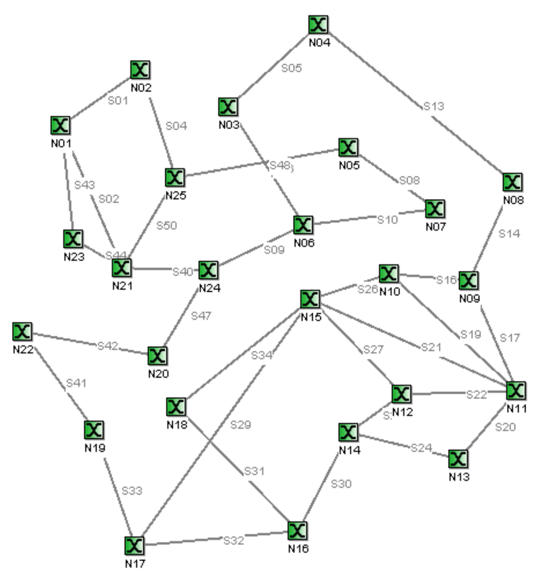
25n50s1 – 32s



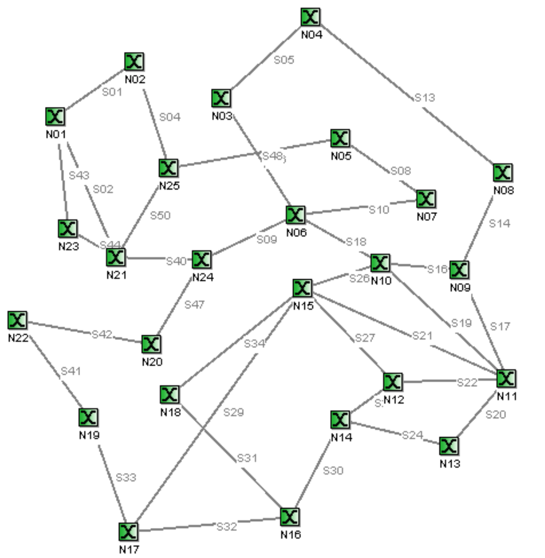
25n50s1 – 33s



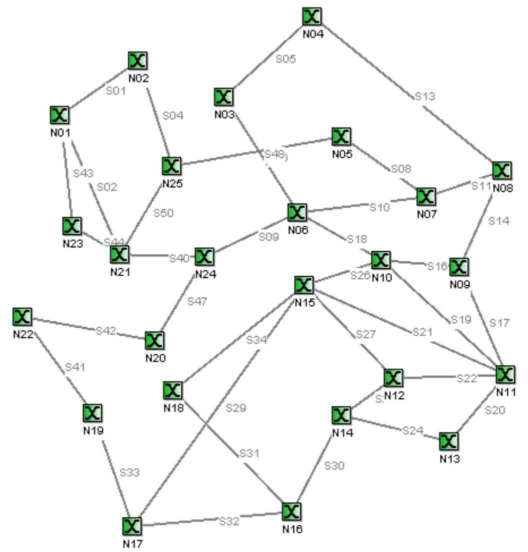
25n50s1 – 34s



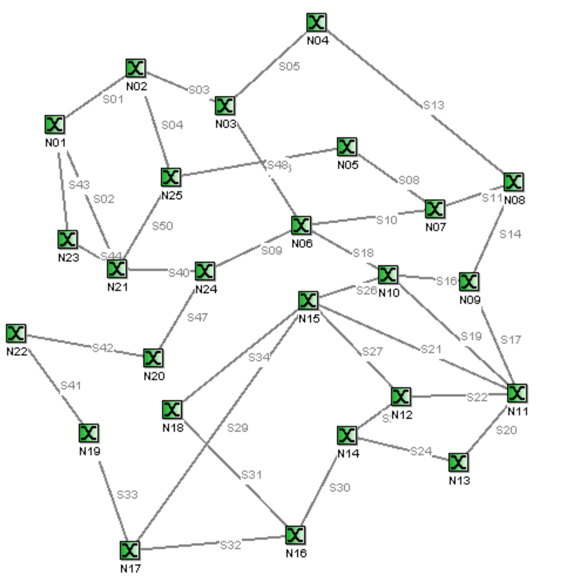
25n50s1 – 35s



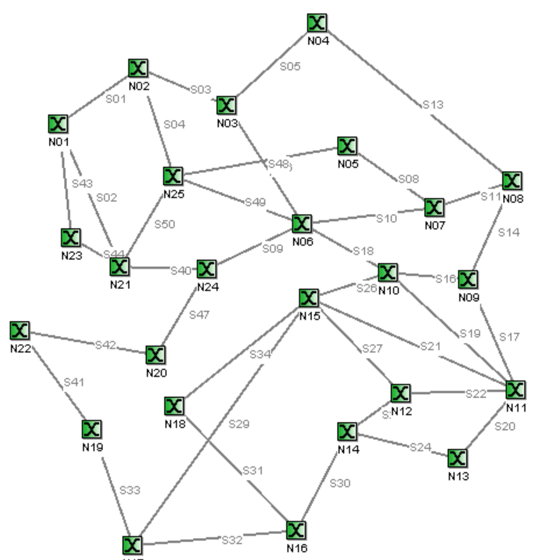
25n50s1 – 36s



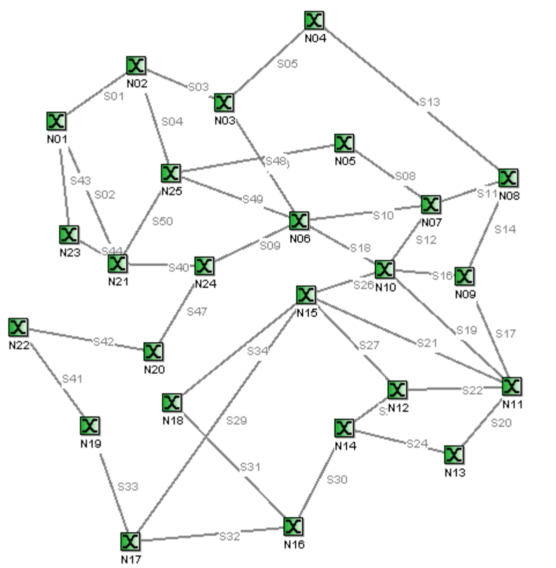
25n50s1 – 37s



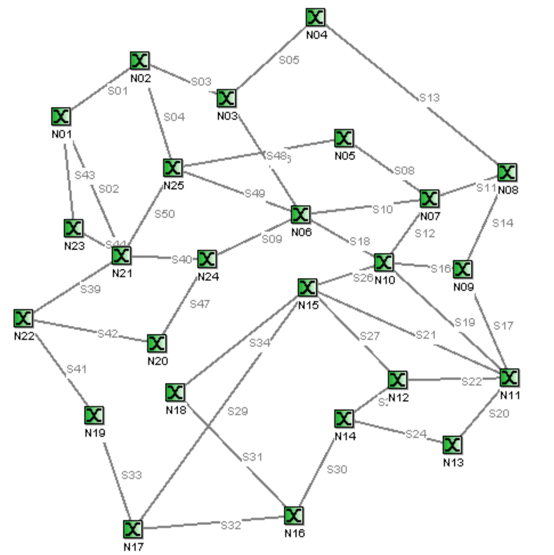
25n50s1 – 38s



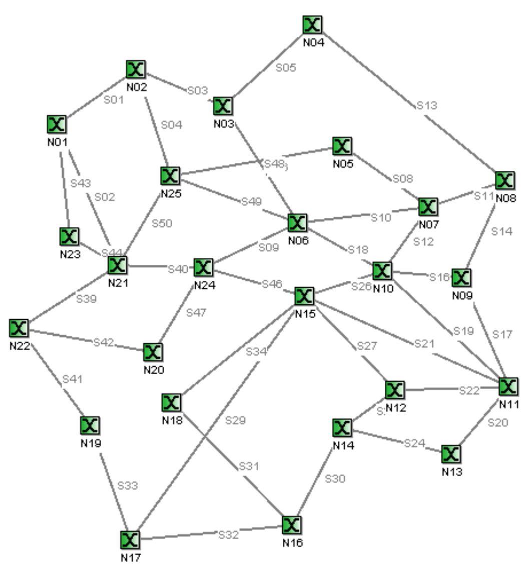
25n50s1 – 39s



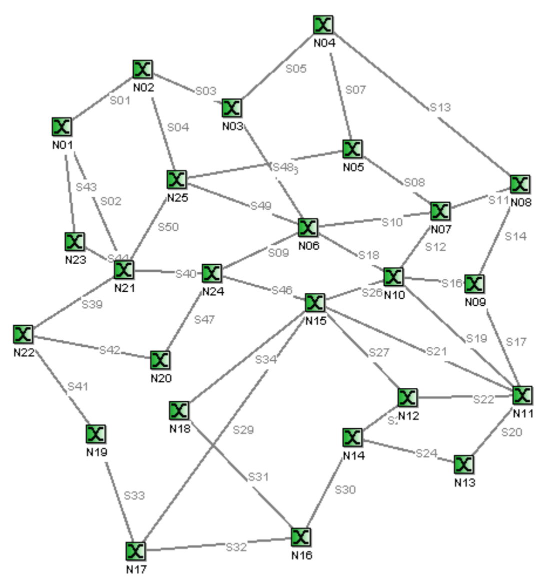
25n50s1 – 40s



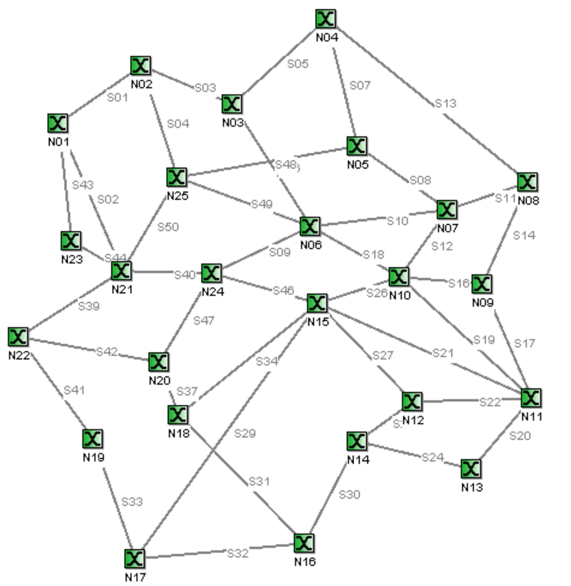
25n50s1 – 41s



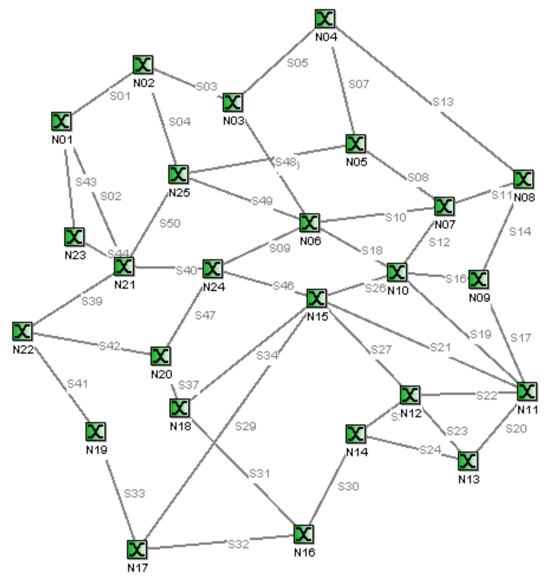
25n50s1 – 42s



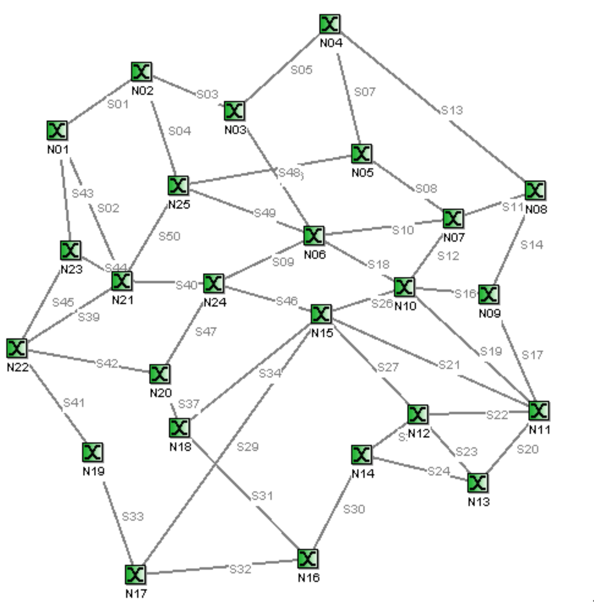
25n50s1 – 43s



25n50s1 – 44s

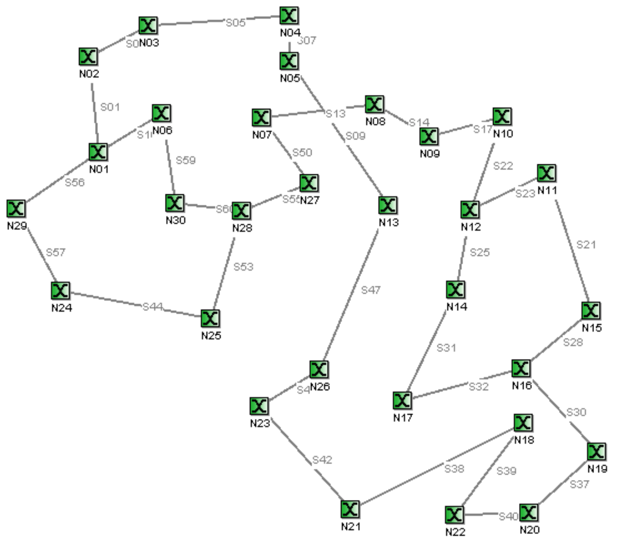


25n50s1 – 45s

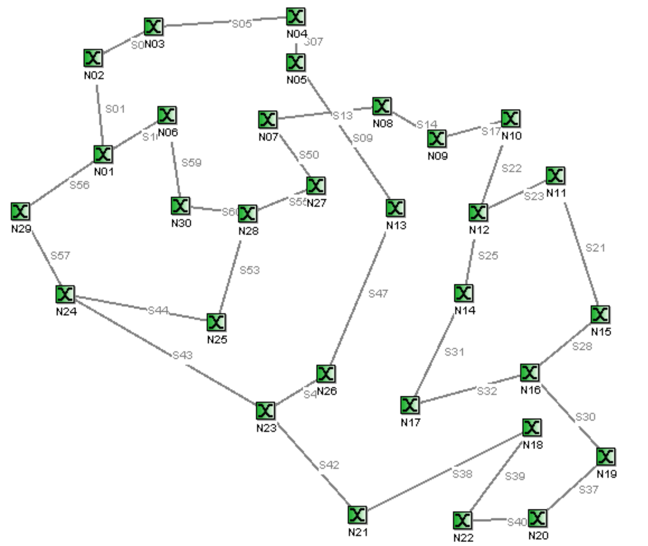


A.4 30n60s1 Master Network

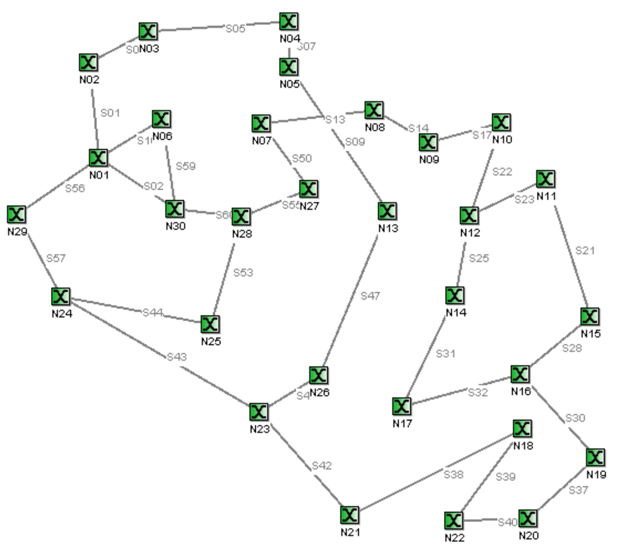
30n60s1 – 32s



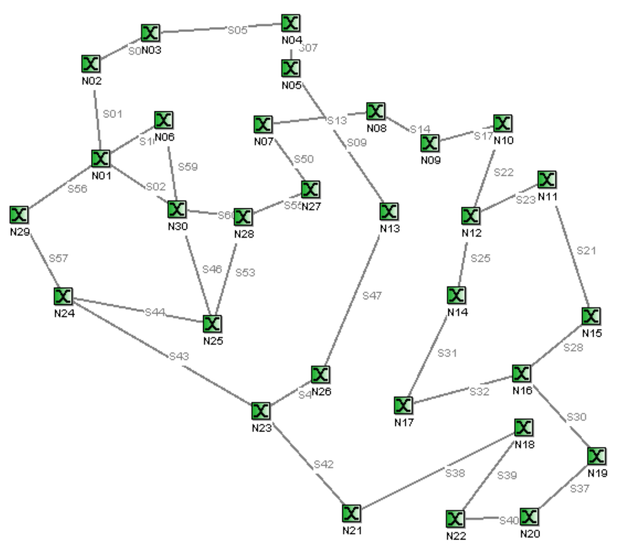
30n60s1 – 33s



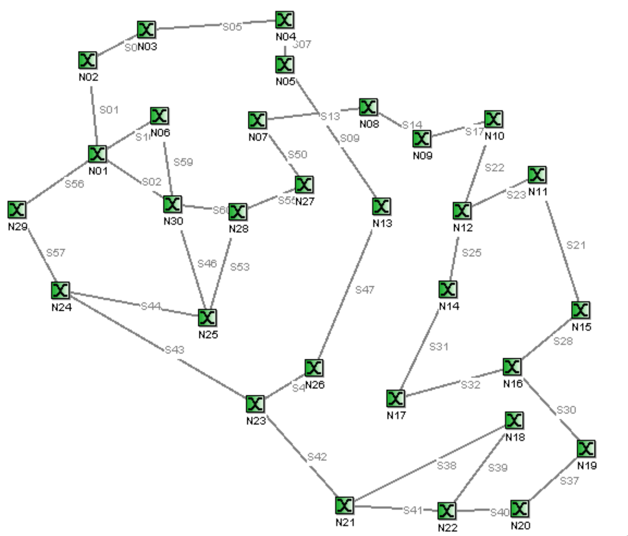
30n60s1 – 34s



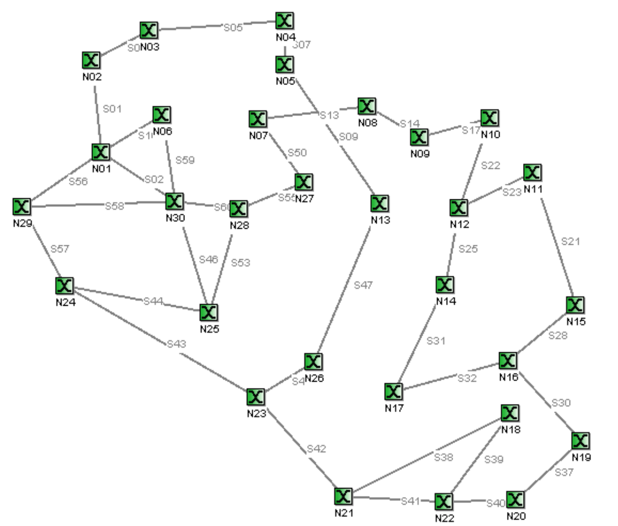
30n60s1 – 35s



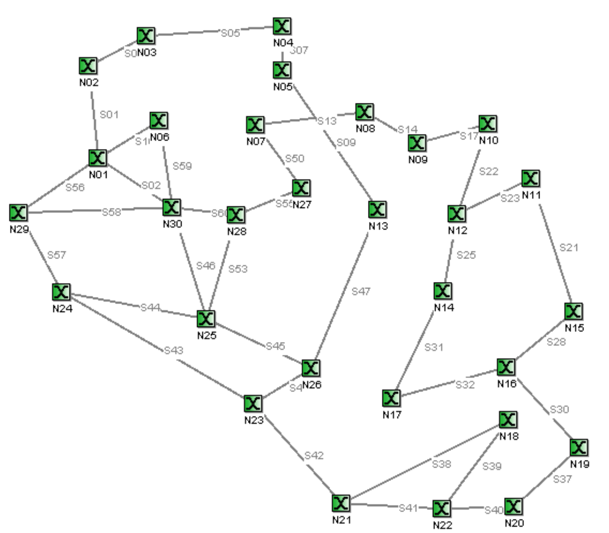
30n60s1 – 36s



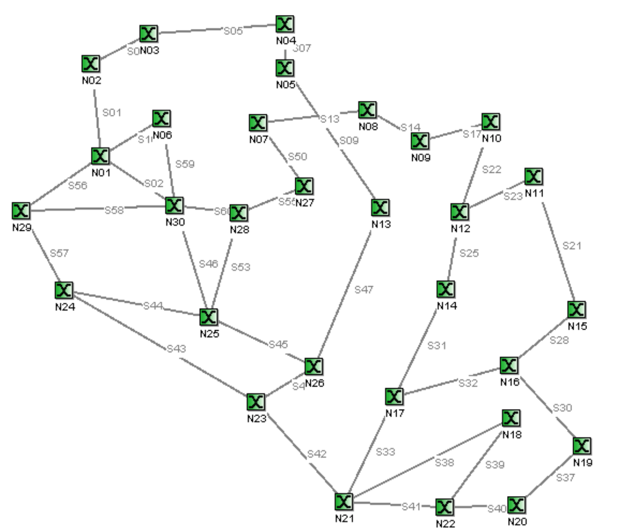
30n60s1 – 37s



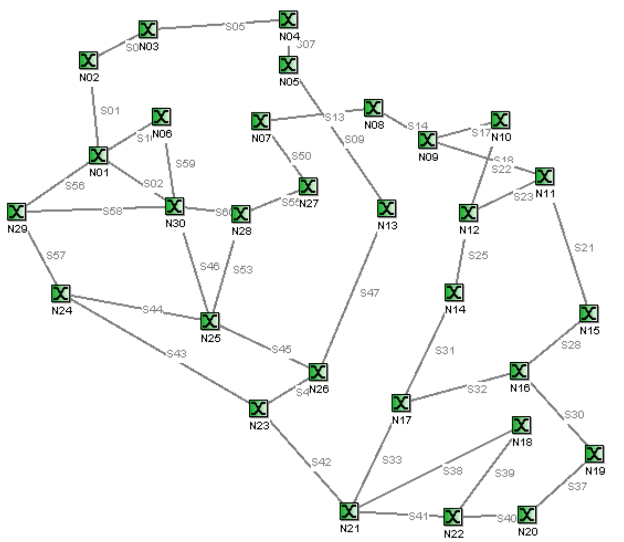
30n60s1 – 38s



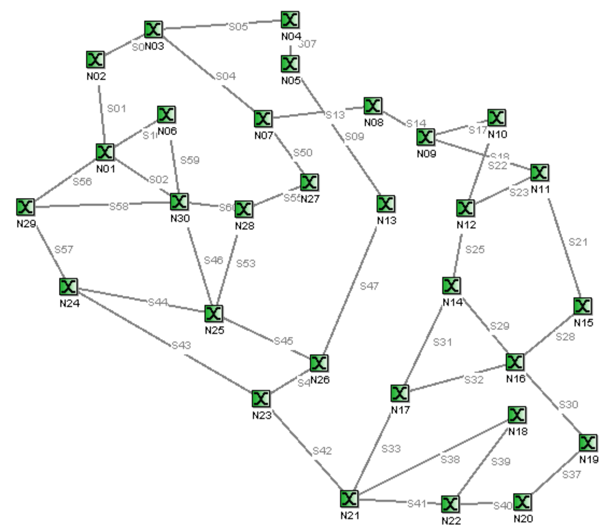
30n60s1 – 39s



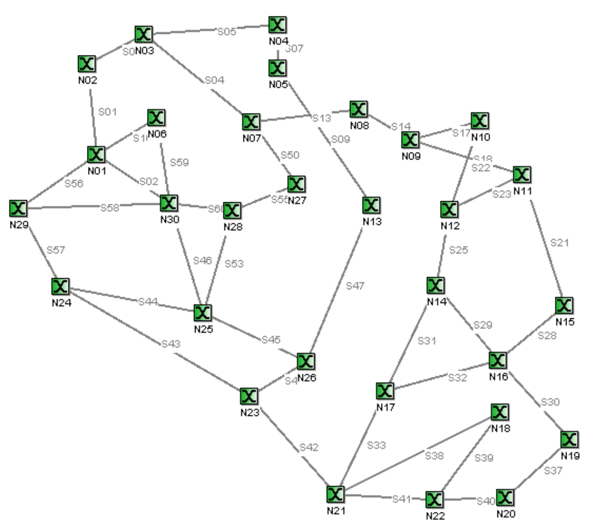
30n60s1 – 40s



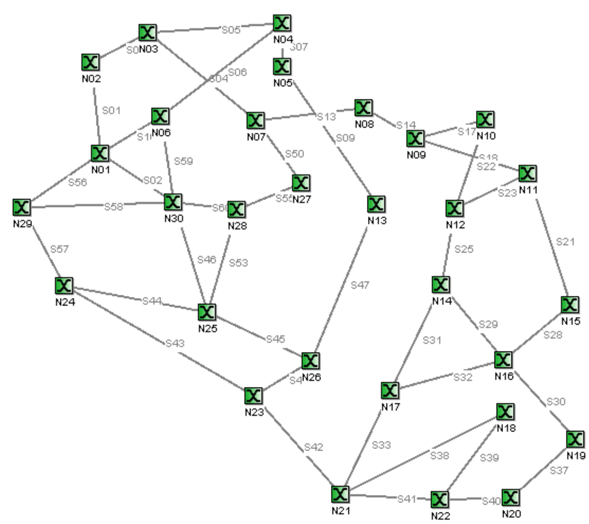
30n60s1 – 41s



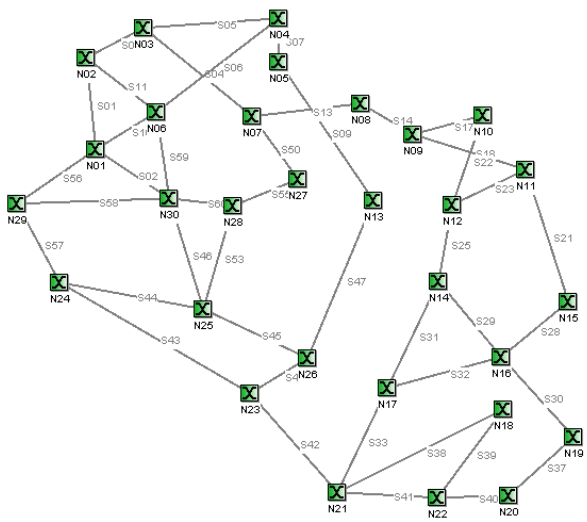
30n60s1 – 42s



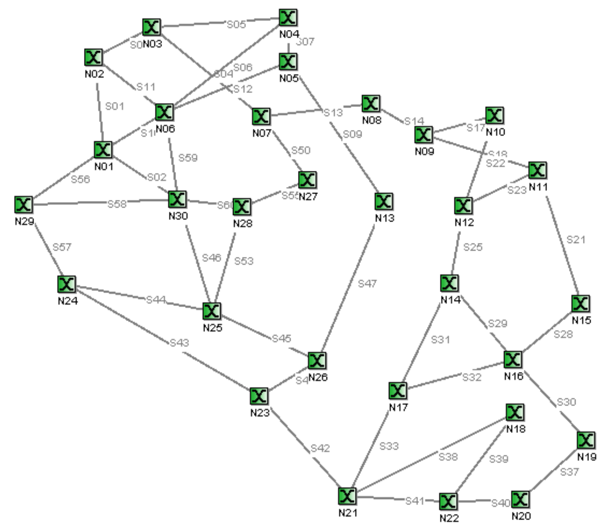
30n60s1 – 43s



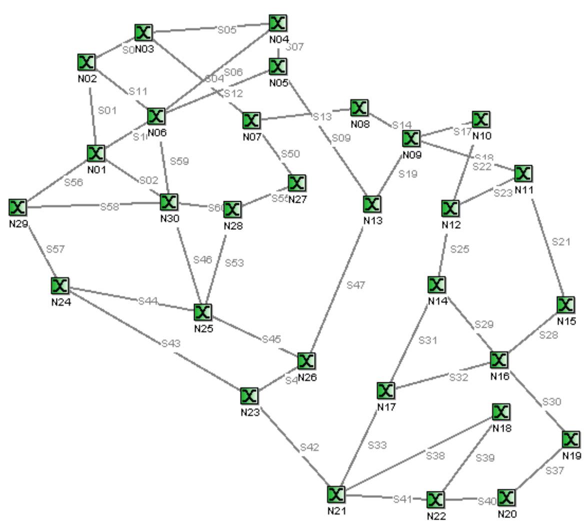
30n60s1 – 44s



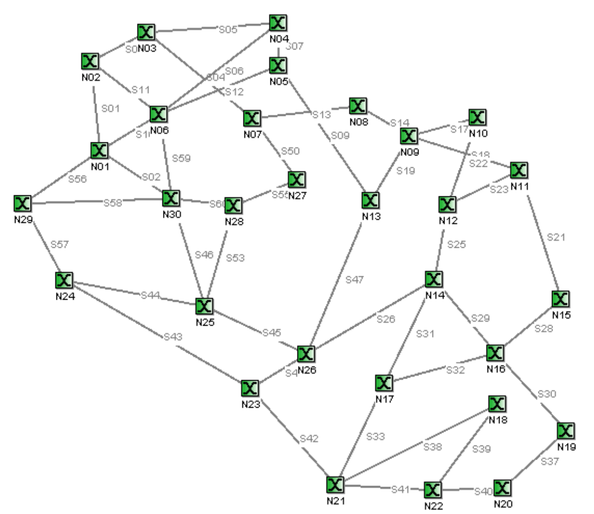
30n60s1 – 45s



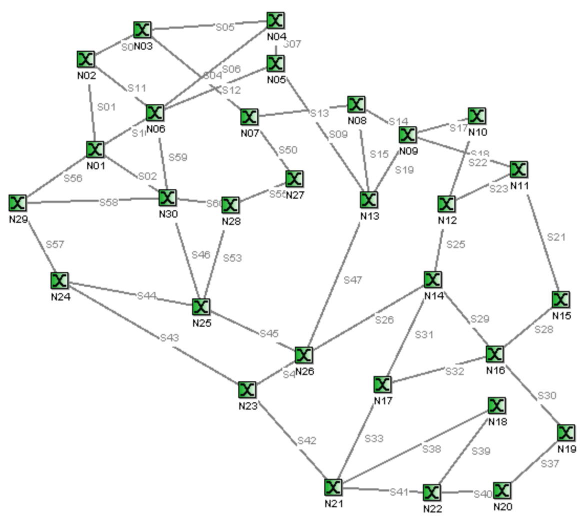
30n60s1 – 46s



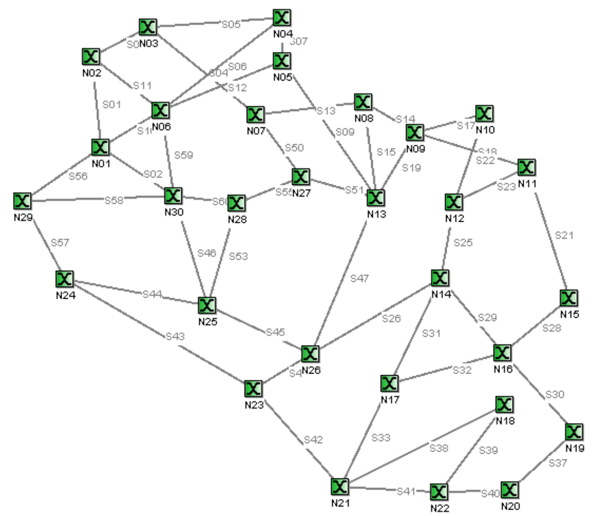
30n60s1 – 47s



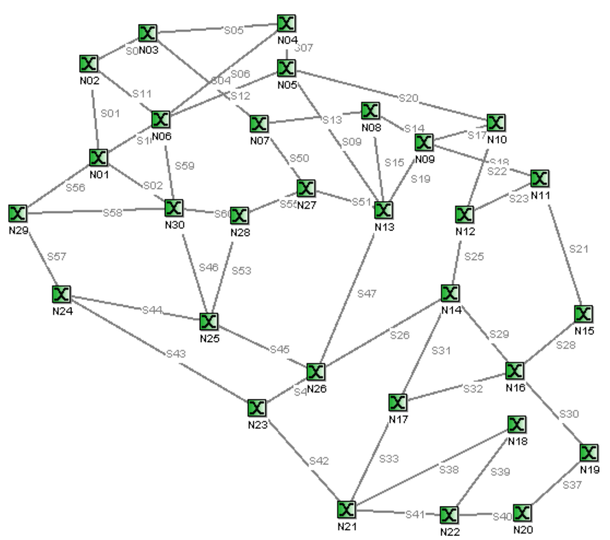
30n60s1 – 48s



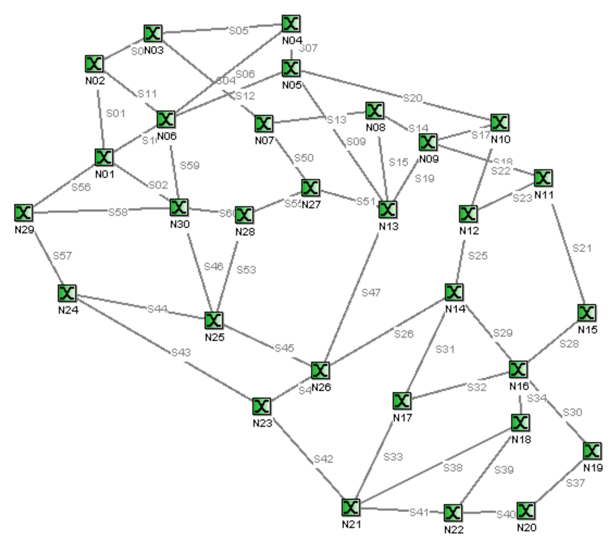
30n60s1 – 49s



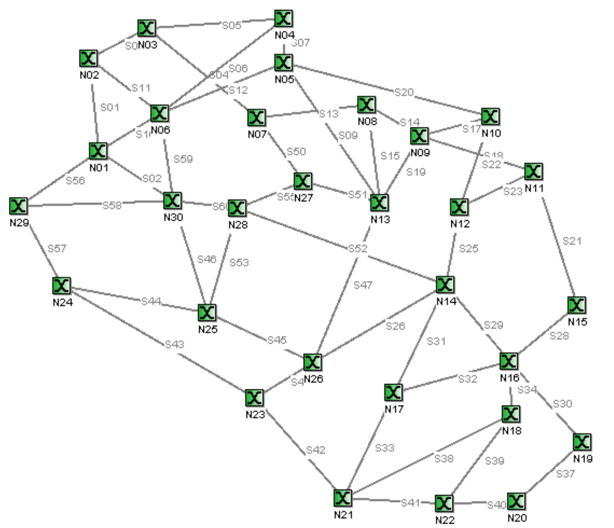
30n60s1 – 50s



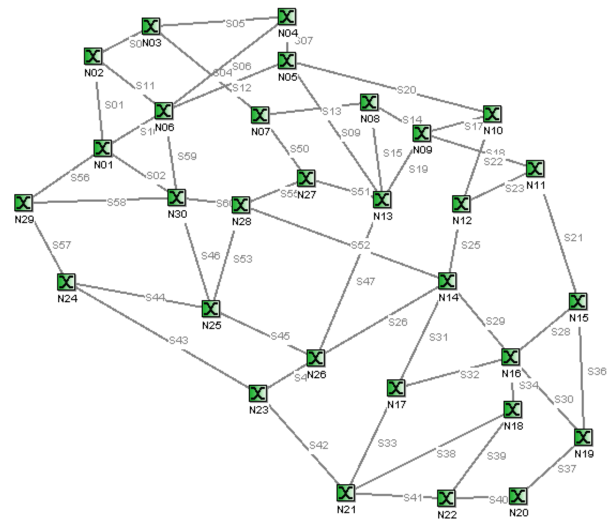
30n60s1 – 51s



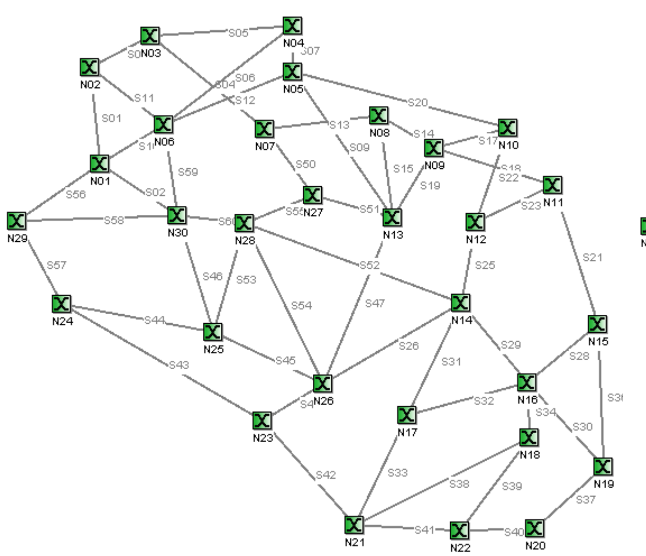
30n60s1 – 52s



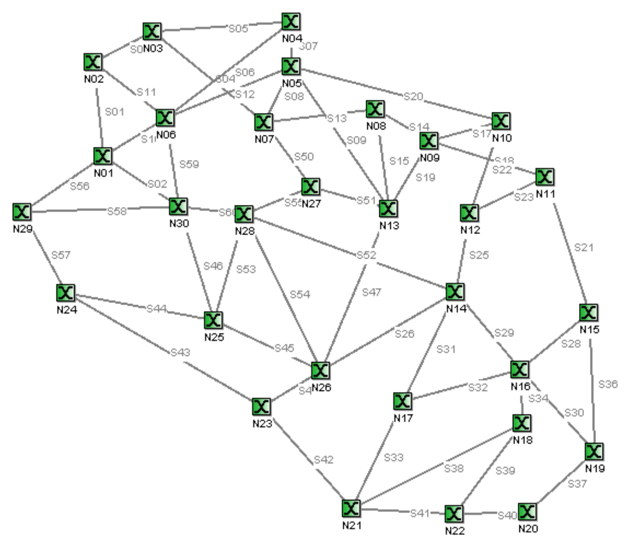
30n60s1 – 53s



30n60s1 – 54s

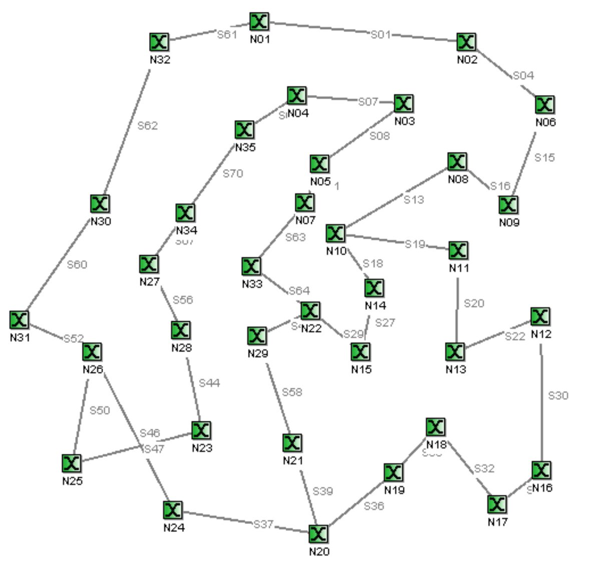


30n60s1 – 55s

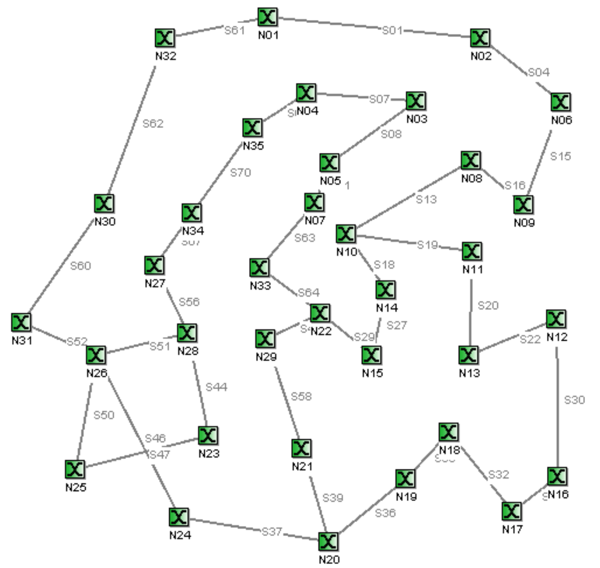


A.5 35N70S1 MASTER NETWORK

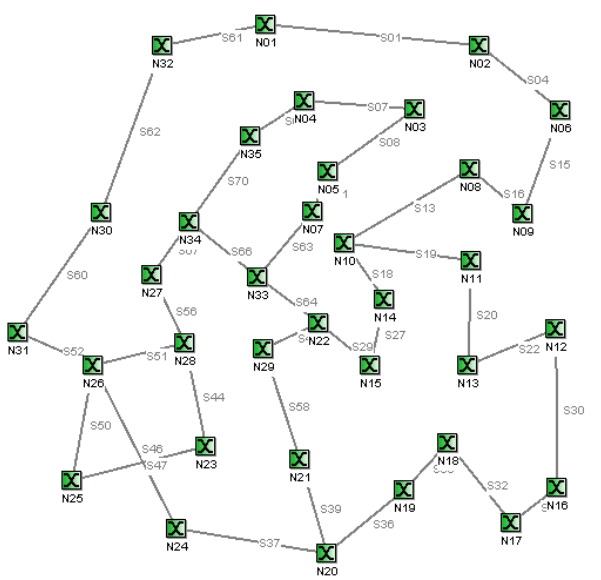
35n70s1 – 37s



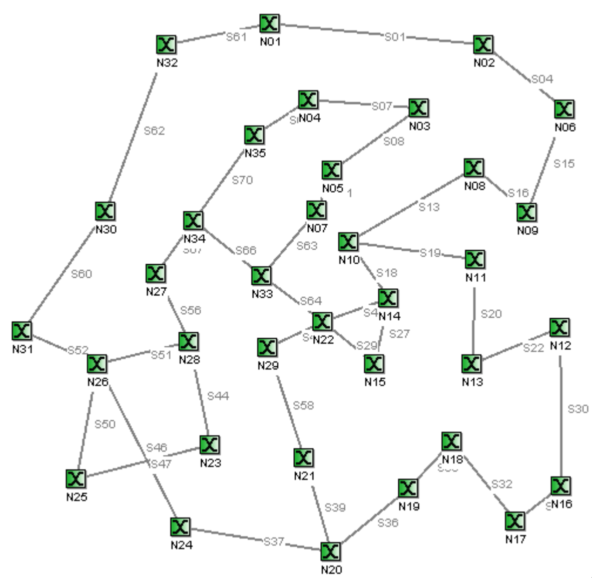
35n70s1 – 38s



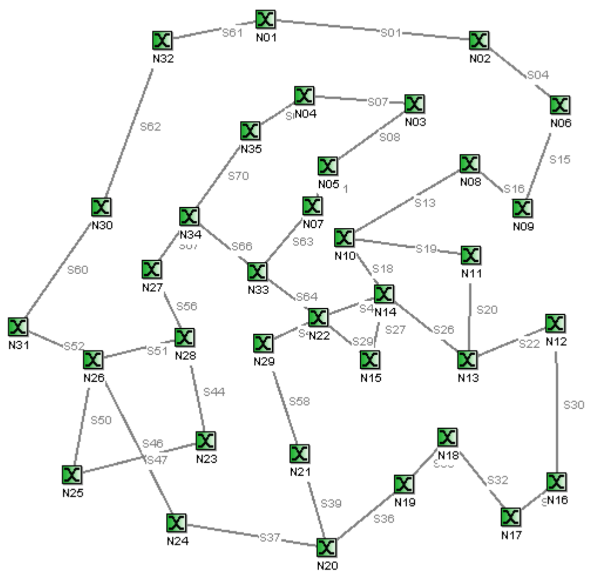
35n70s1 – 39s



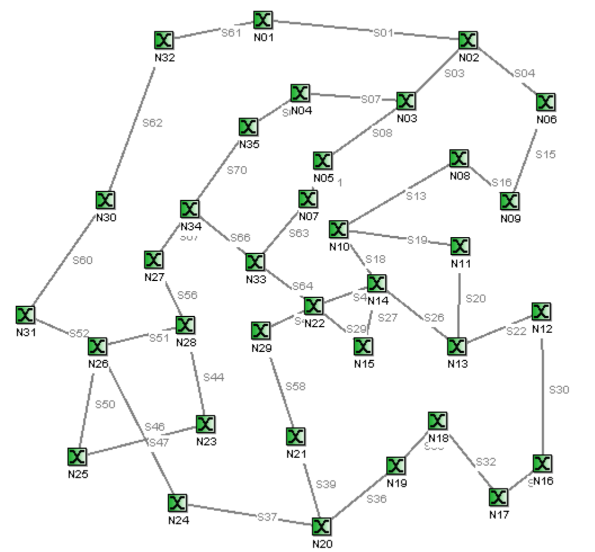
35n70s1 – 40s



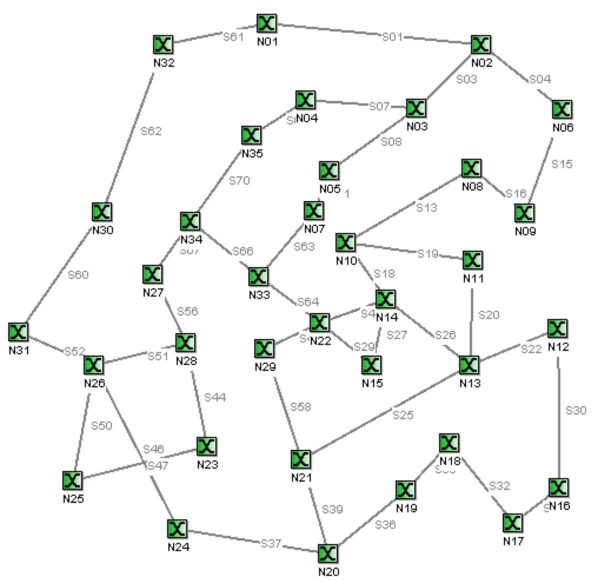
35n70s1 – 41s



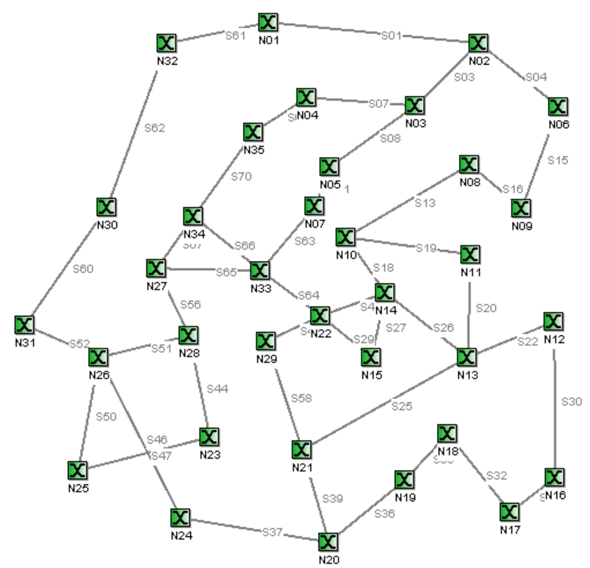
35n70s1 – 42s



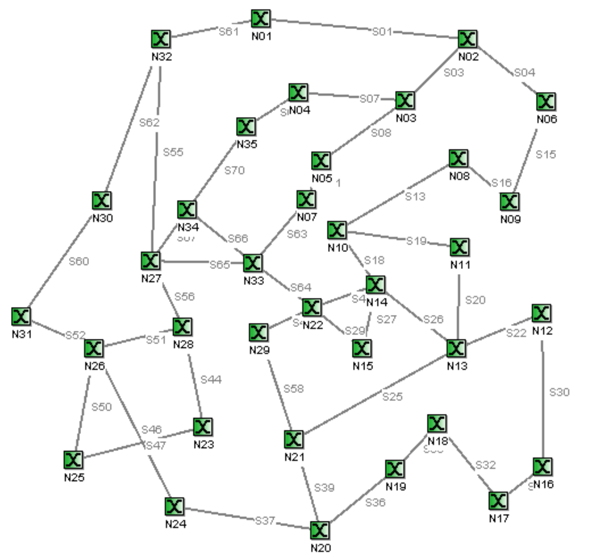
35n70s1 – 43s



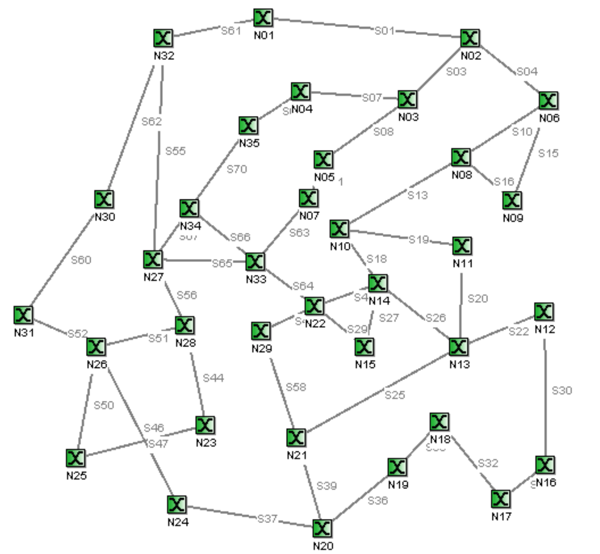
35n70s1 – 44s



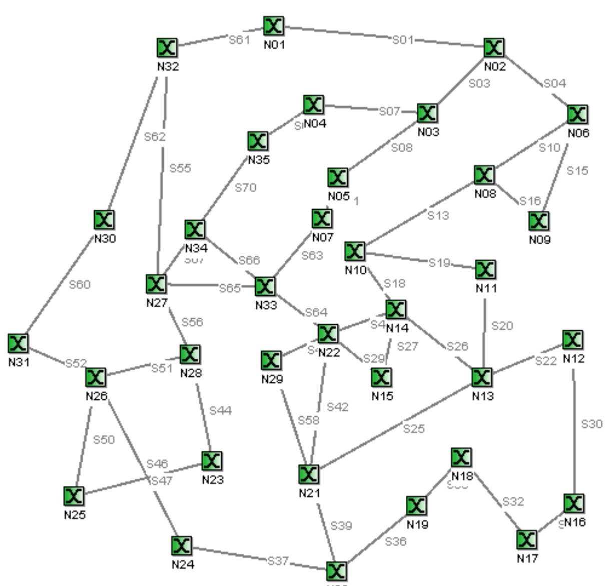
35n70s1 – 45s



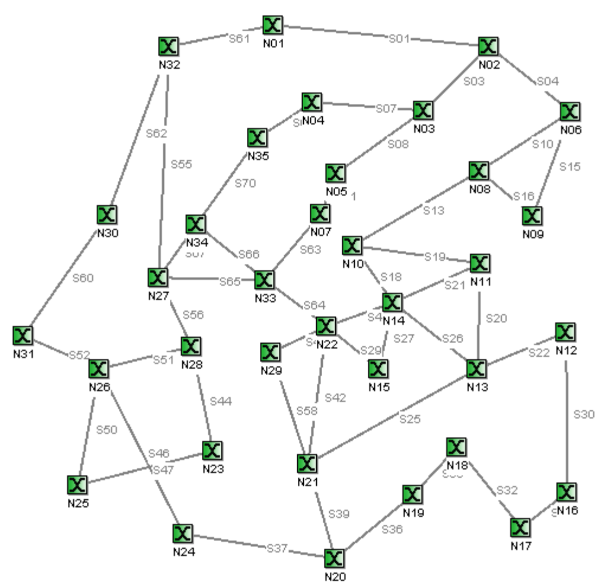
35n70s1 – 46s



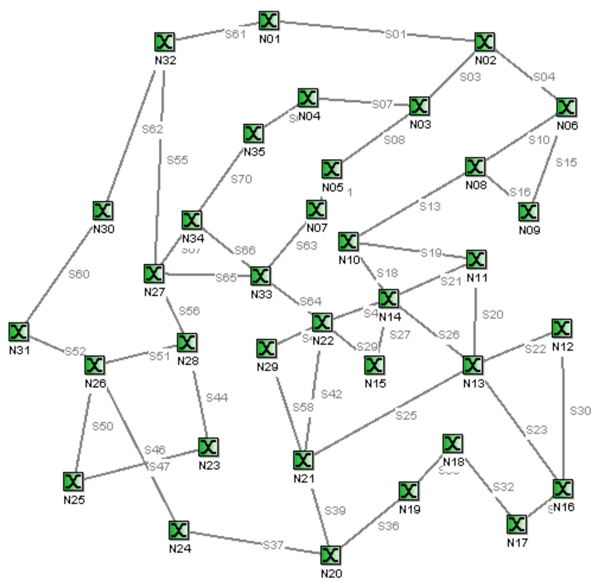
35n70s1 – 47s



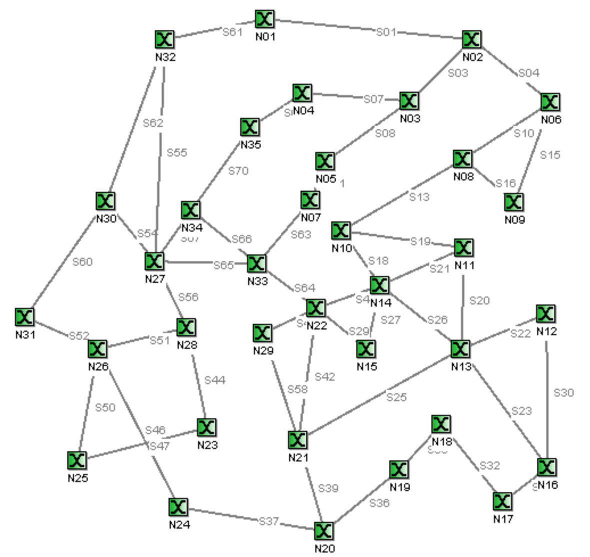
35n70s1 – 48s



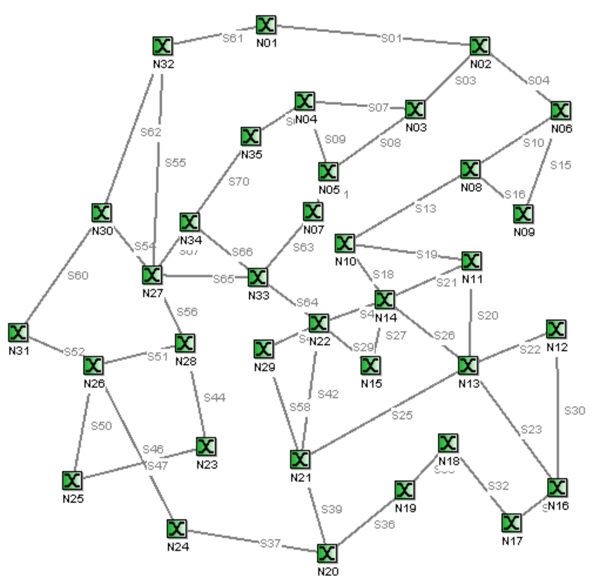
35n70s1 – 49s



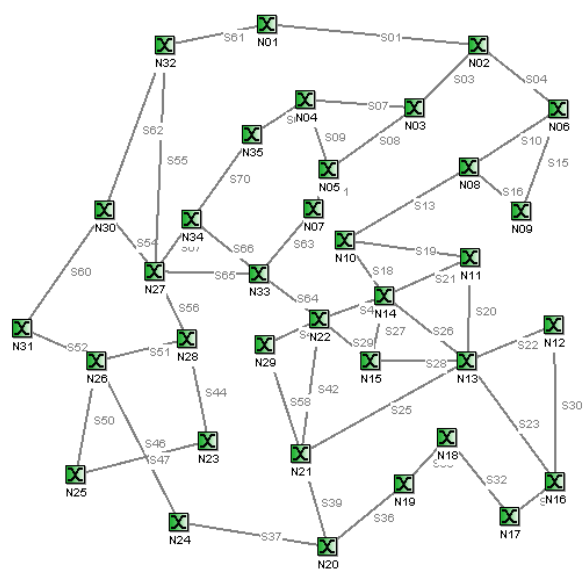
35n70s1 – 50s



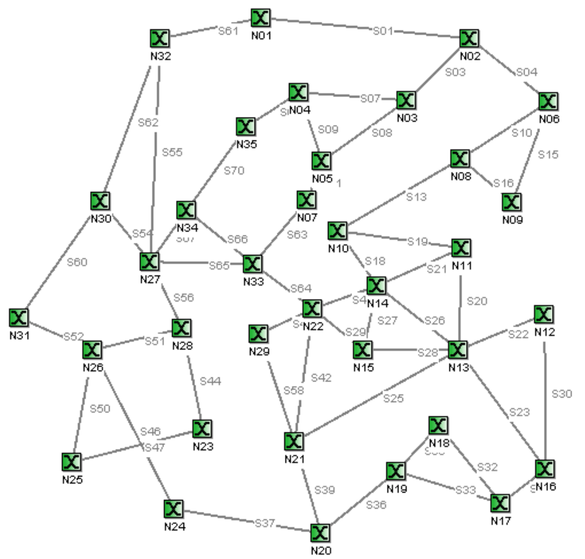
35n70s1 – 51s



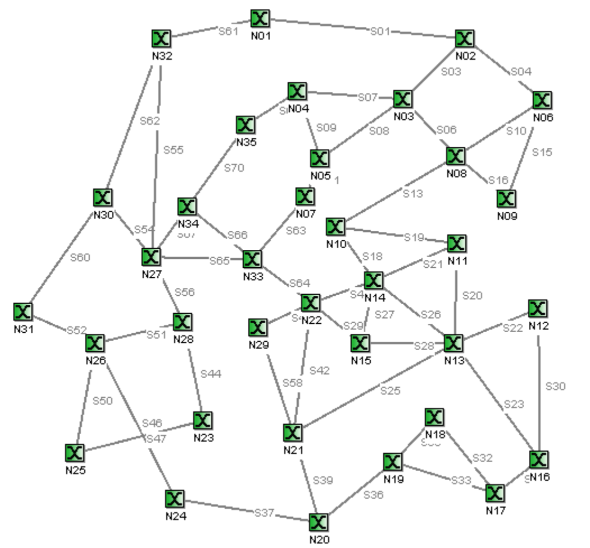
35n70s1 – 52s



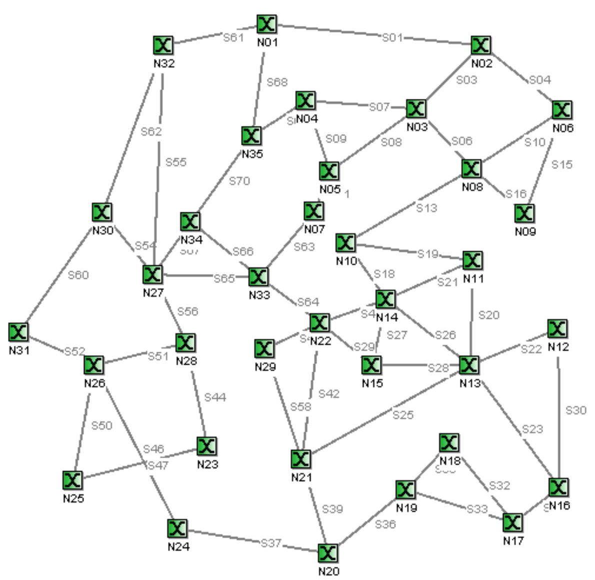
35n70s1 – 53s



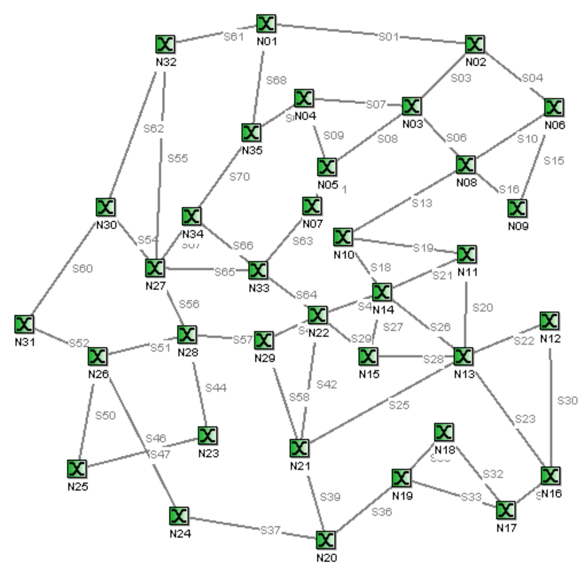
35n70s1 – 54s



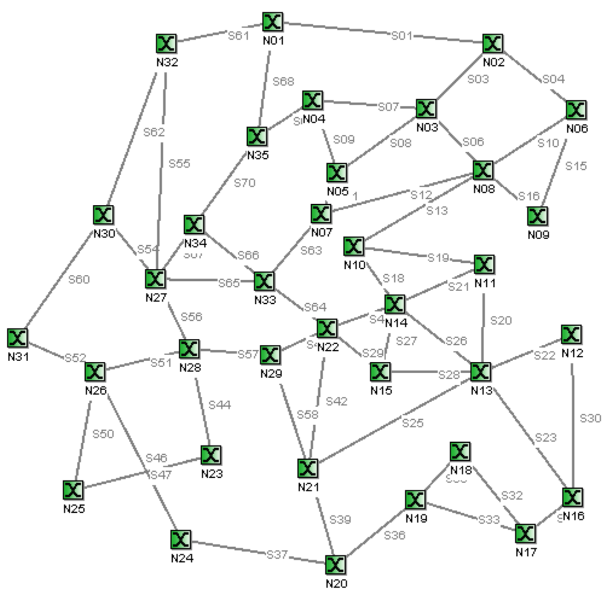
35n70s1 – 55s



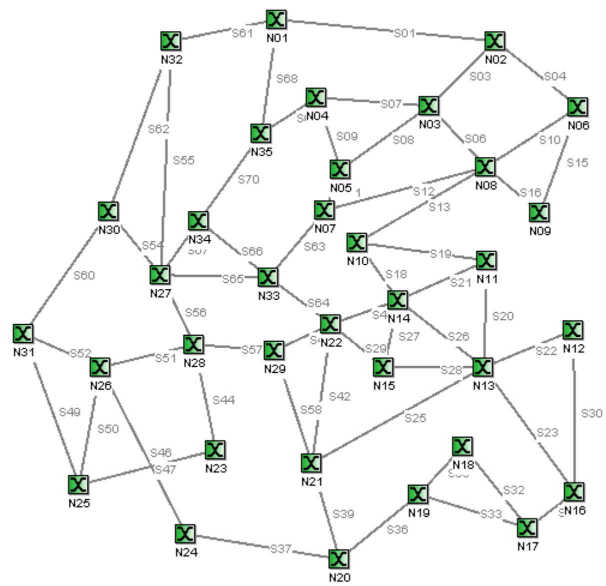
35n70s1 – 56s



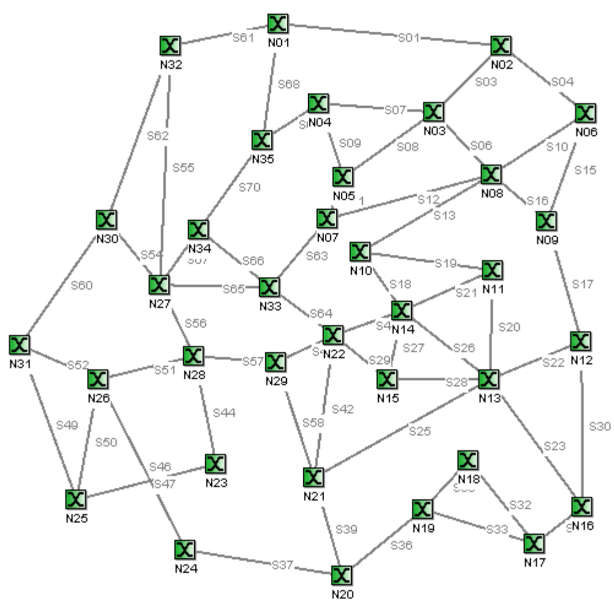
35n70s1 – 57s



35n70s1 – 58s

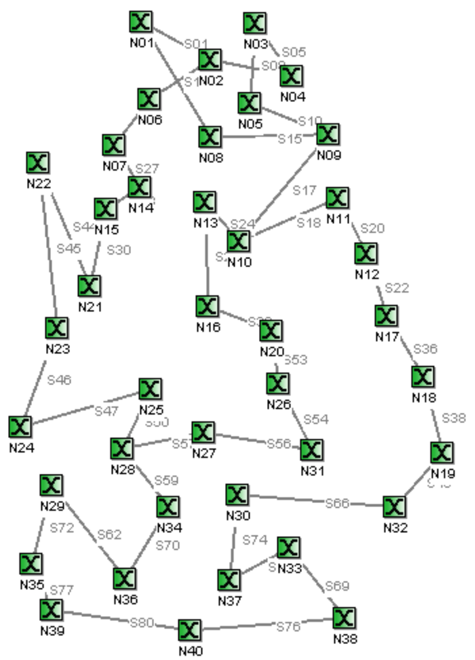


35n70s1 – 59s

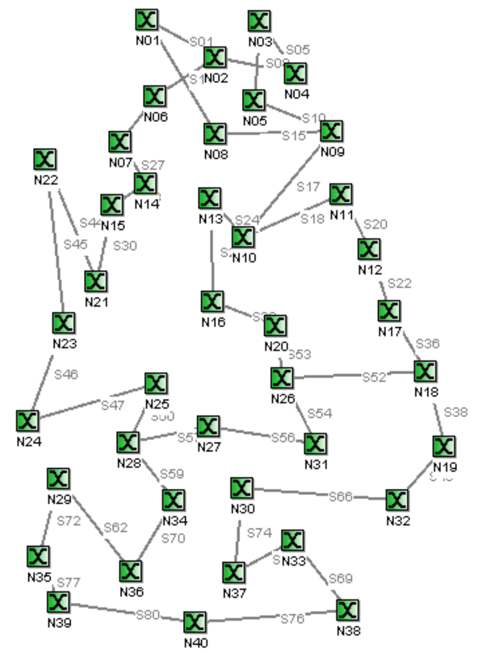


A.6 40N80S1 MASTER NETWORK

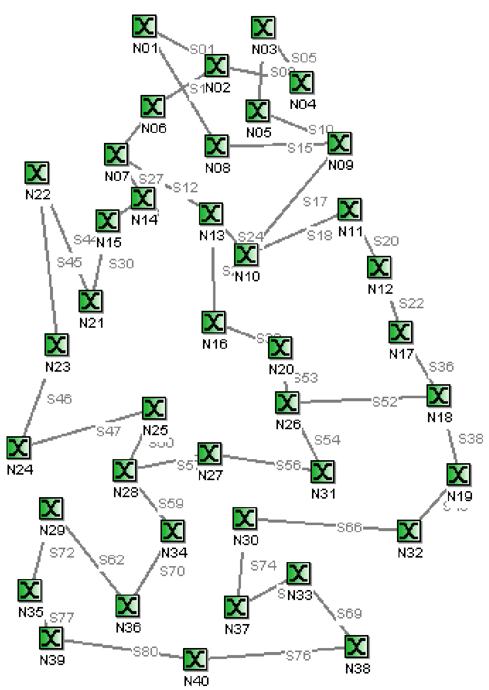
40n80s1 – 42s



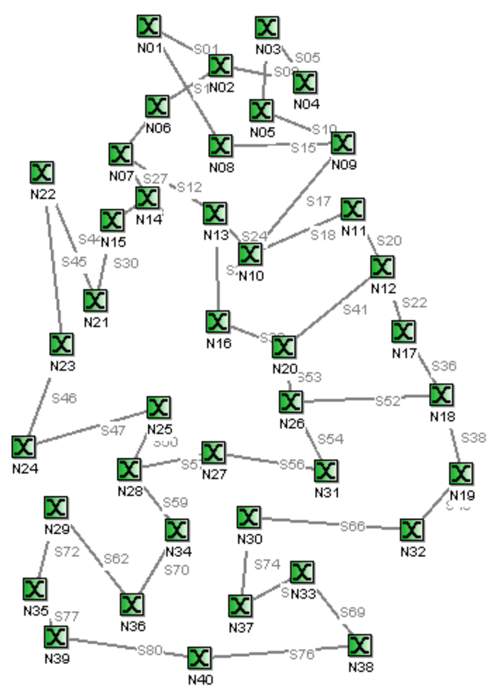
40n80s1 – 43s



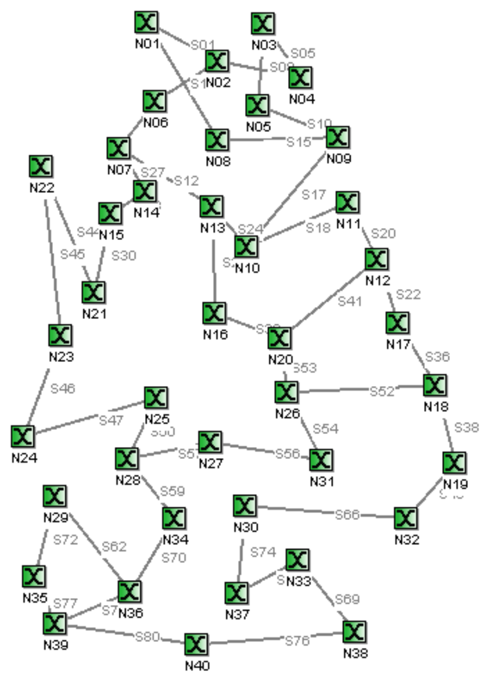
40n80s1 – 44s



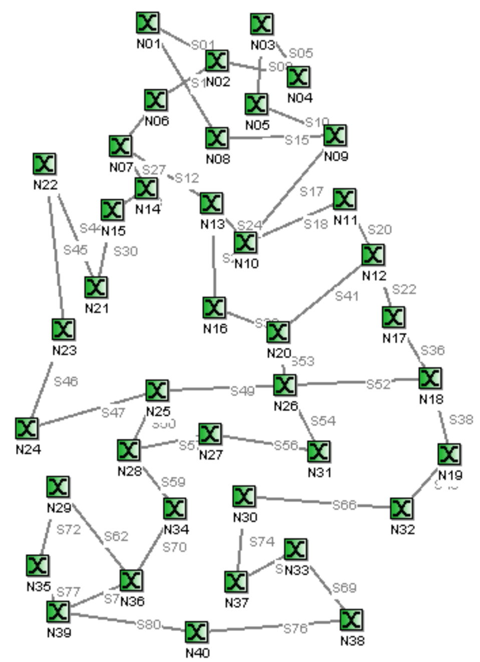
40n80s1 – 45s



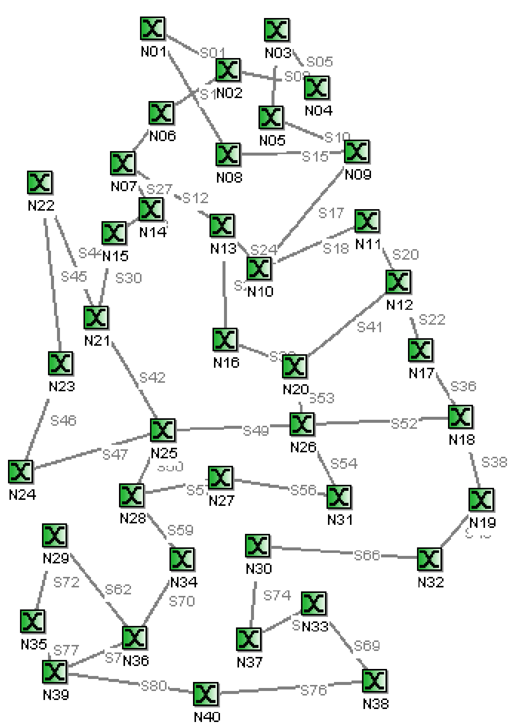
40n80s1 – 46s



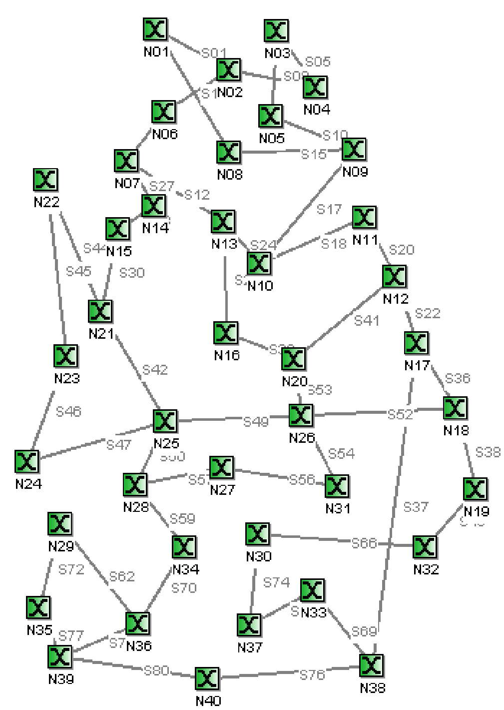
40n80s1 – 47s



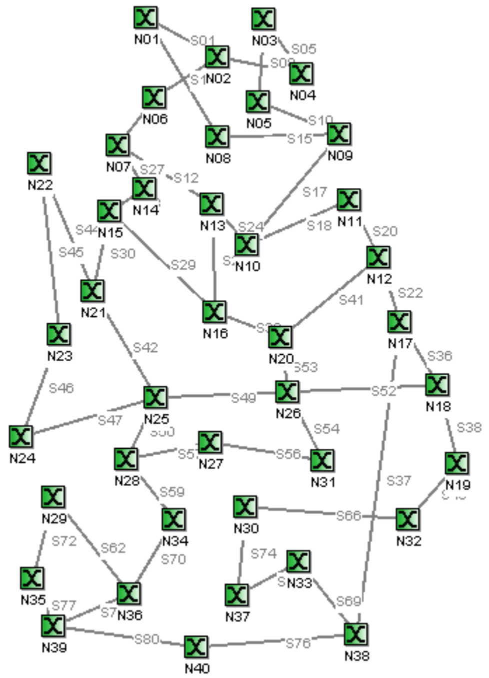
40n80s1 – 48s



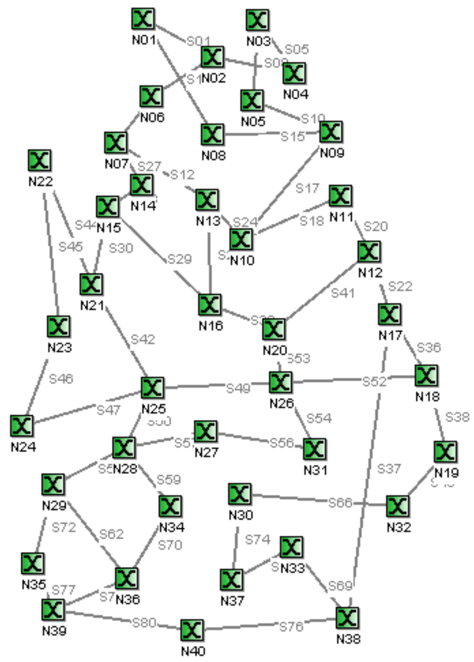
40n80s1 – 49s



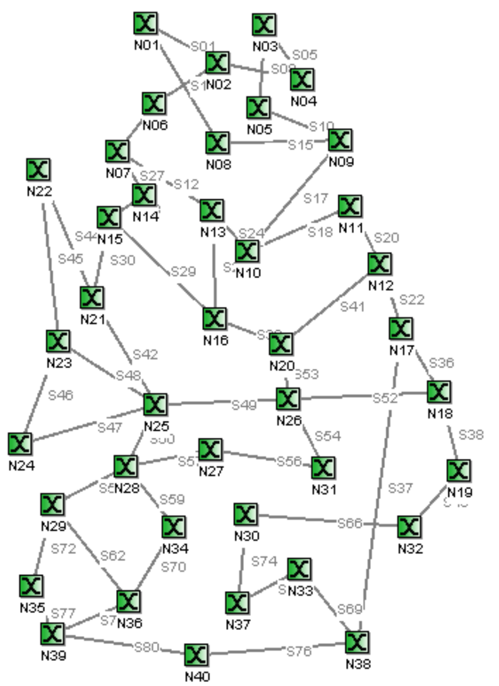
40n80s1 – 50s



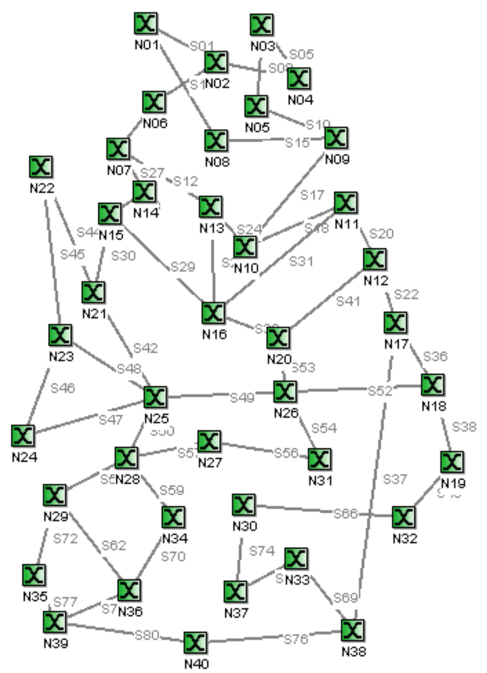
40n80s1 – 51s



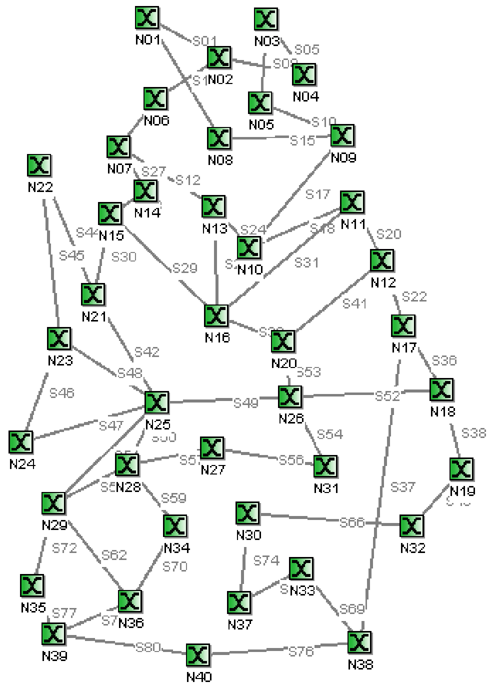
40n80s1 – 52s



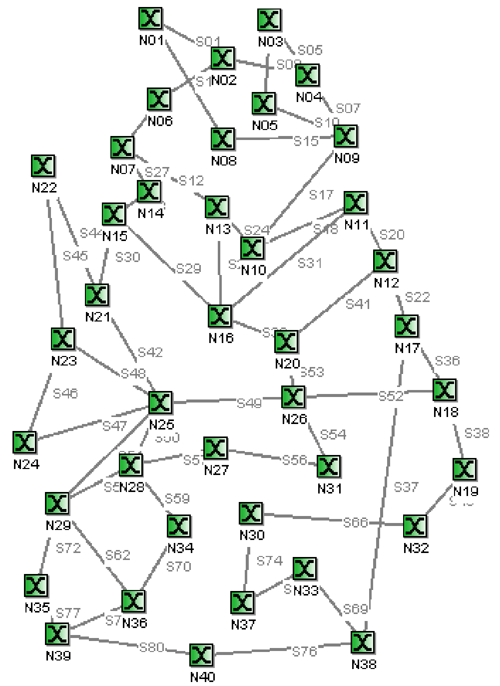
40n80s1 – 53s



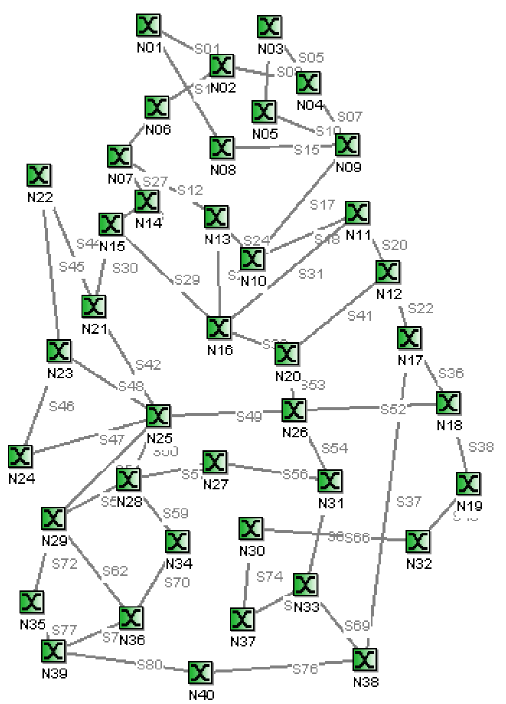
40n80s1 – 54s



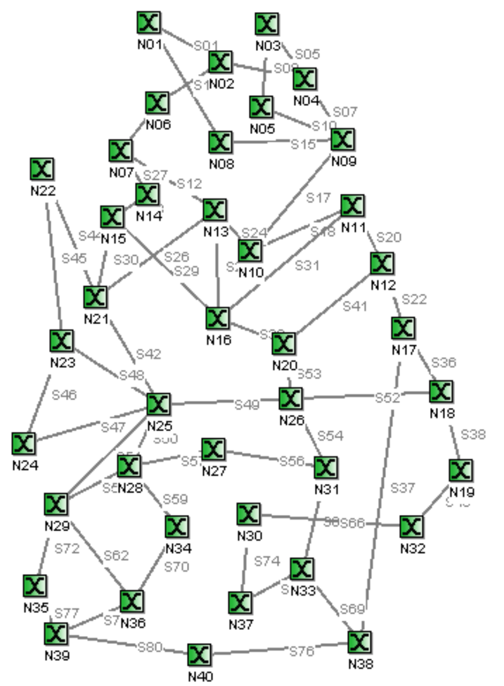
40n80s1 – 55s



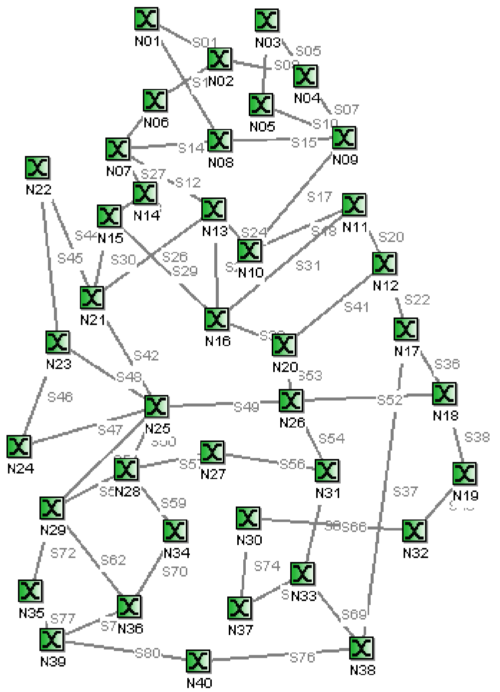
40n80s1 – 56s



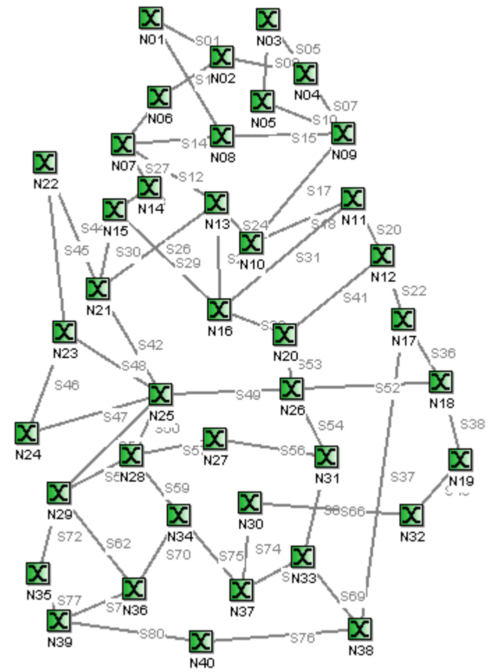
40n80s1 – 57s



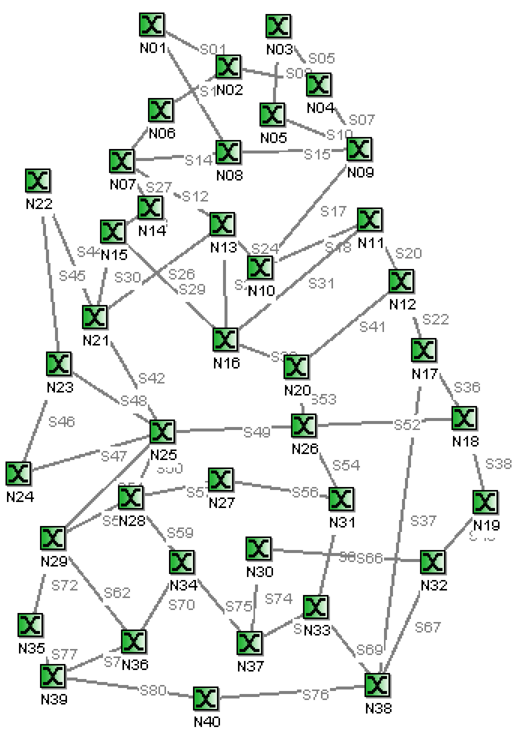
40n80s1 – 58s



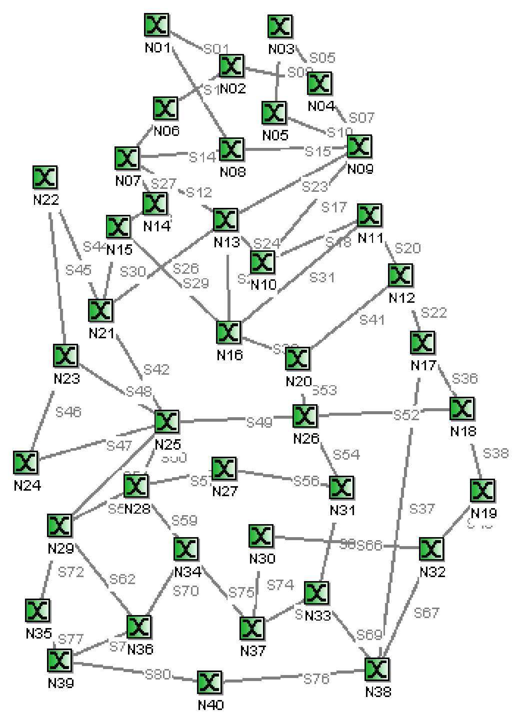
40n80s1 – 59s



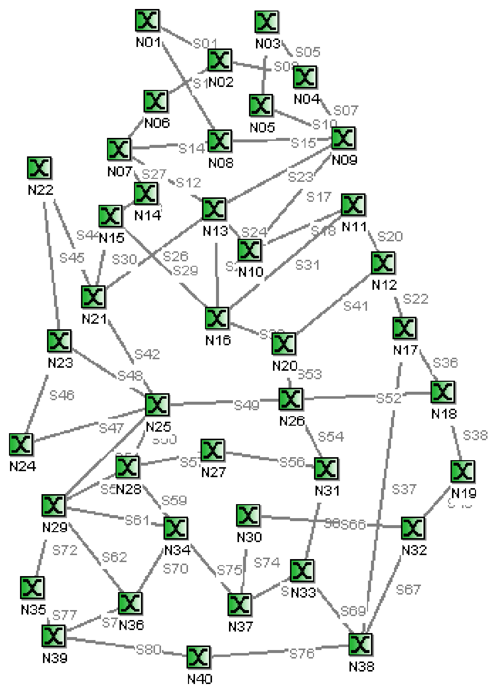
40n80s1 – 60s



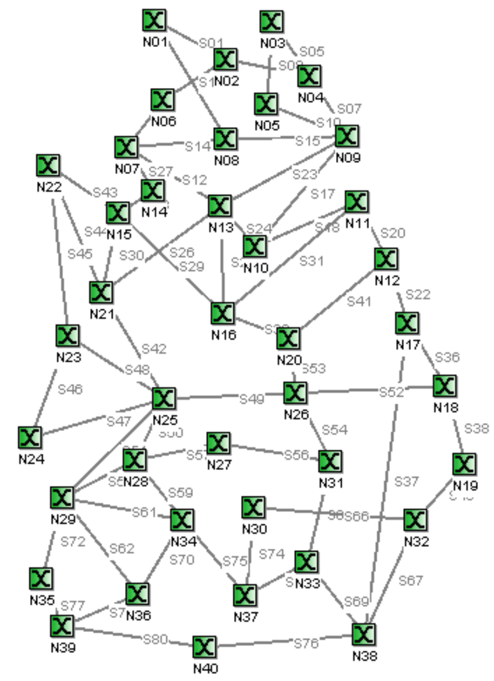
40n80s1 – 61s



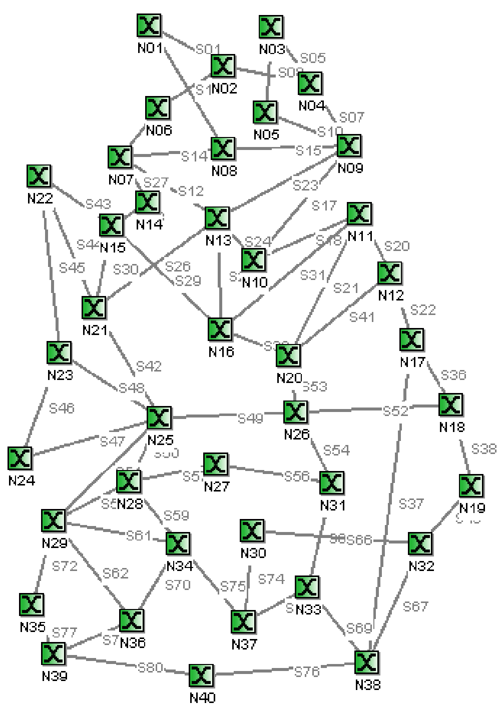
40n80s1 – 62s



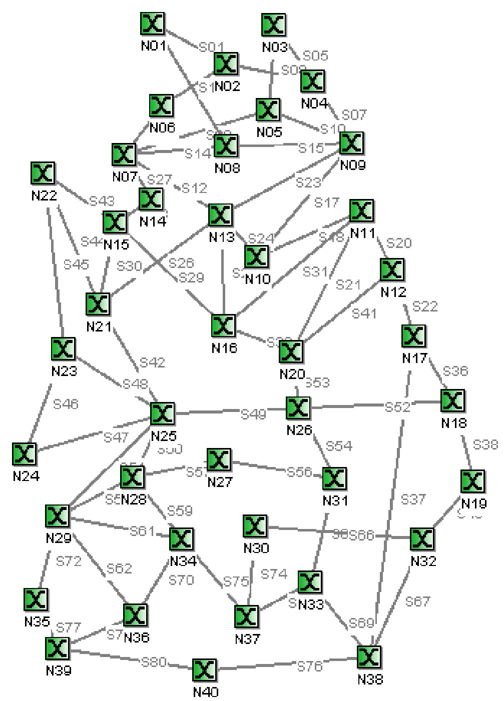
40n80s1 – 63s



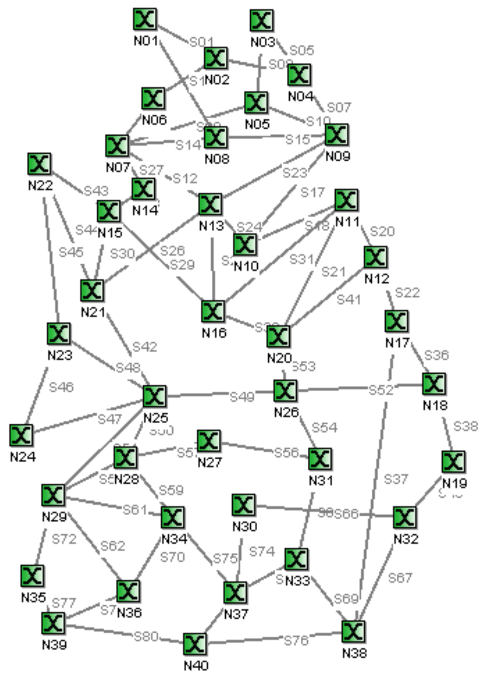
40n80s1 – 64s



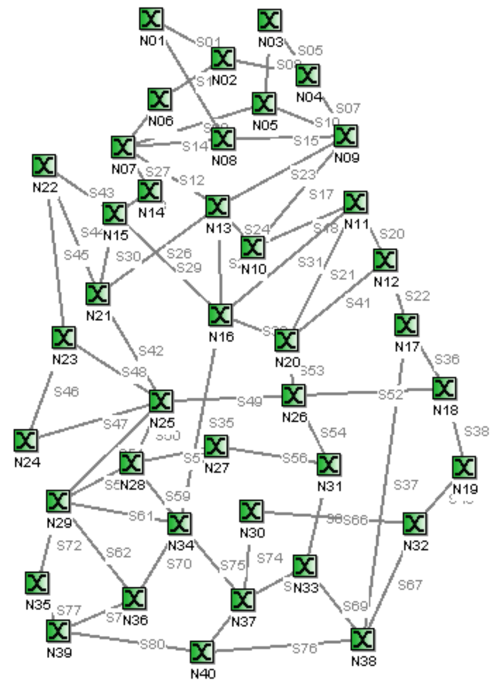
40n80s1 – 65s



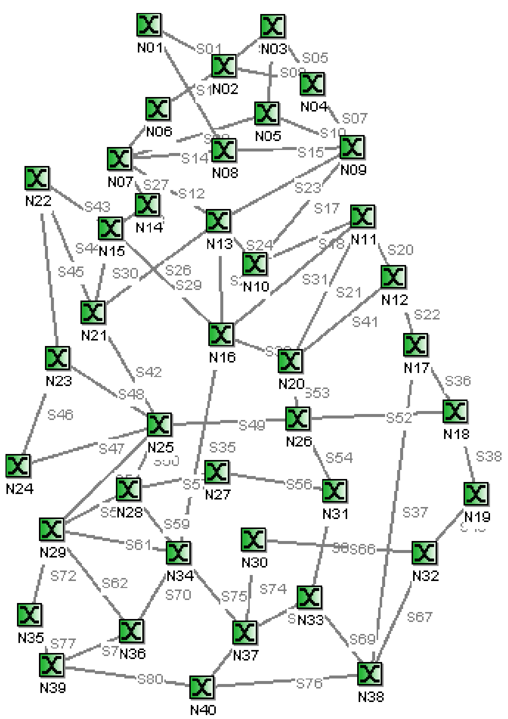
40n80s1 – 66s



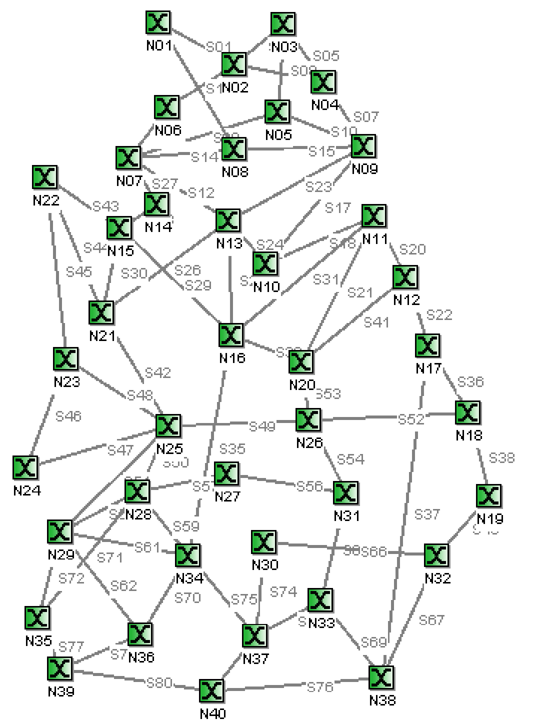
40n80s1 – 67s



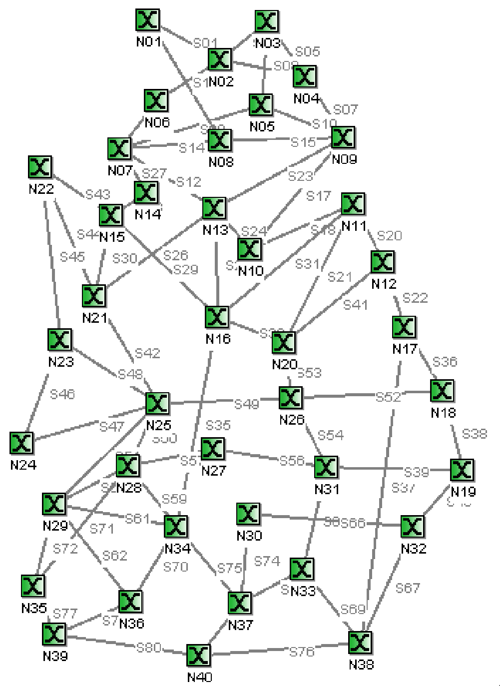
40n80s1 – 68s



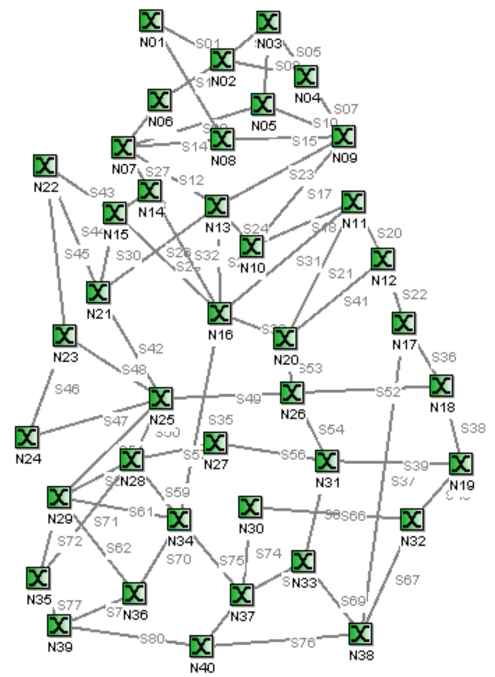
40n80s1 – 69s



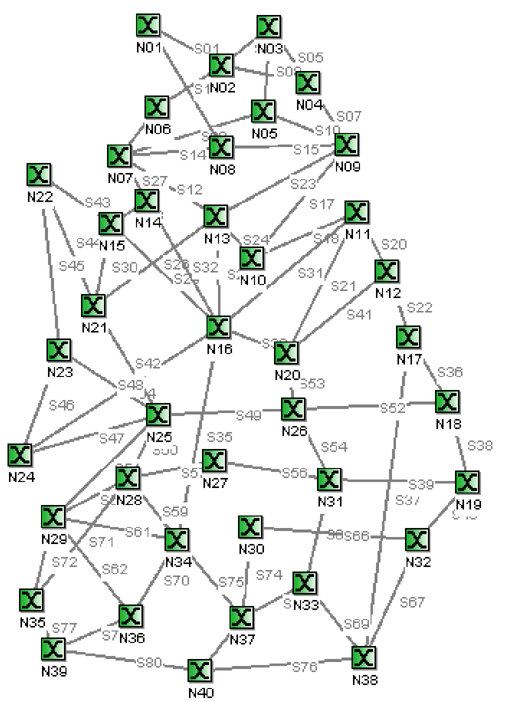
40n80s1 – 70s



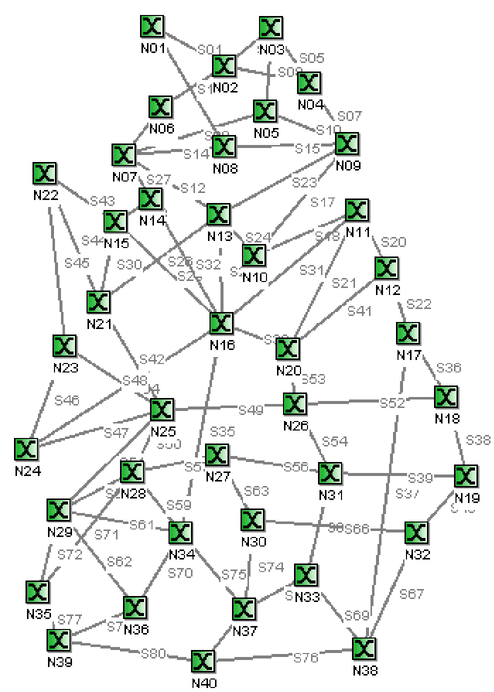
40n80s1 – 71s



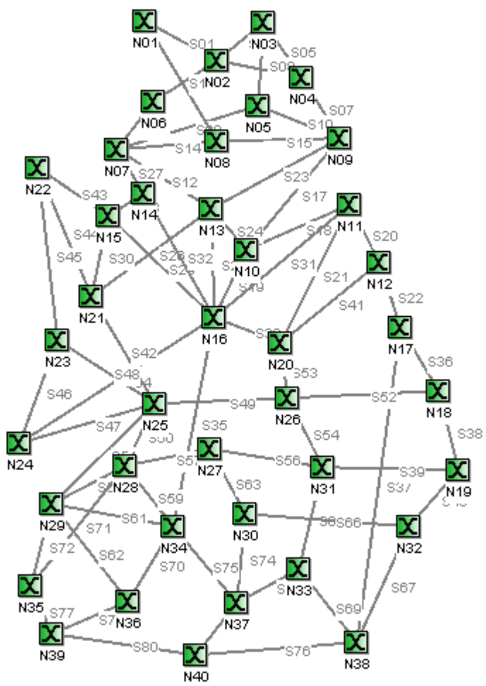
40n80s1 – 72s



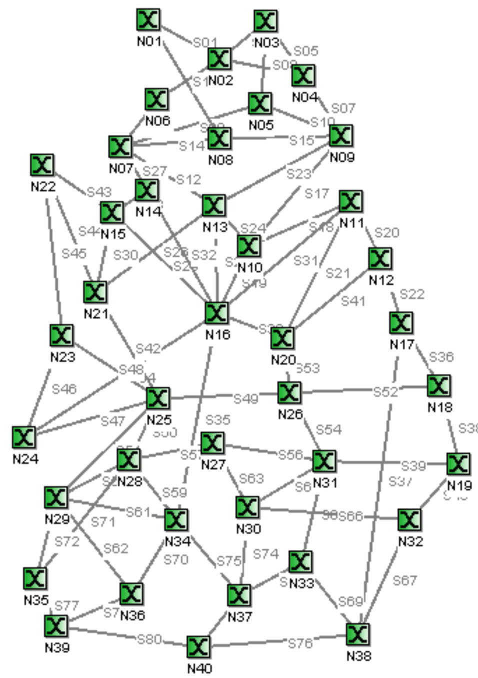
40n80s1 – 73s



40n80s1 – 74s



40n80s1 – 75s



APPENDIX B

B.1 Summary of the Total of Logical Bypass Spans Added to a Network

Network Family	Test Network	Meta-Mesh network description				
		Number of added logical bypass spans	Number of single-node chains	Number of double-node chains	Number of triple-node chains	Number of multiple-node chains
15n30s1	15n16s	2	-	1	-	1
	15n17s	-	-	-	-	-
	15n18s	-	-	-	-	-
	15n19s	-	-	-	-	-
	15n20s	-	-	-	-	-
	15n21s	-	-	-	-	-
	15n22s	-	-	-	-	-
	15n23s	1	1	-	-	-
	15n24s	-	-	-	-	-
20n40s1	20n21s	2	-	-	-	2
	20n22s	1	-	-	-	1
	20n23s	1	1	-	-	-
	20n24s	2	2	-	-	-
	20n25s	1	1	-	-	-
	20n26s	2	2	-	-	-
	20n27s	3	3	-	-	-
	20n28s	4	4	-	-	-
	20n29s	3	3	-	-	-
	20n30s	3	3	-	-	-
	20n31s	1	1	-	-	-
	20n32s	1	1	-	-	-
	20n33s	1	1	-	-	-
	20n34s	1	1	-	-	-
	20n35s	1	1	-	-	-
25n50s1	25n27s	2	1	-	-	1
	25n28s	2	1	-	-	1
	25n29s	2	1	-	1	-
	25n30s	2	1	1	-	-
	25n31s	3	2	1	-	-
	25n32s	2	2	-	-	-
	25n33s	2	2	-	-	-
	25n34s	1	1	-	-	-

	25n35s	1	1	-	-	-
	25n36s	1	1	-	-	-
	25n37s	1	1	-	-	-
	25n38s	1	1	-	-	-
	25n39s	1	1	-	-	-
	25n40s	-	-	-	-	-
	25n41s	1	1	-	-	-
	25n42s	1	1	-	-	-
	25n43s	1	1	-	-	-
	25n44s	1	1	-	-	-
	25n45s	-	-	-	-	-
30n60s1	30n32s	2	-	2	-	-
	30n33s	1	-	1	-	-
	30n34s	2	1	1	-	-
	30n35s	2	1	1	-	-
	30n36s	3	1	2	-	-
	30n37s	3	1	2	-	-
	30n38s	3	1	2	-	-
	30n39s	2	2	-	-	-
	30n40s	2	2	-	-	-
	30n41s	2	2	-	-	-
	30n42s	2	2	-	-	-
	30n43s	1	1	-	-	-
	30n44s	1	1	-	-	-
	30n45s	1	1	-	-	-
	30n46s	1	1	-	-	-
	30n47s	1	1	-	-	-
	30n48s	1	1	-	-	-
	30n49s	1	1	-	-	-
	30n50s	1	1	-	-	-
	30n51s	-	-	-	-	-
	30n52s	-	-	-	-	-
	30n53s	-	-	-	-	-
	30n54s	-	-	-	-	-
	30n55s	-	-	-	-	-
35n70s1	35n37s	-	-	-	-	-
	35n38s	1	-	1	-	-
	35n39s	2	-	1	-	1
	35n40s	3	1	1	-	1
	35n41s	3	1	1	-	1
	35n42s	2	1	1	-	-
	35n43s	2	1	1	-	-
	35n44s	2	1	1	-	-
	35n45s	2	1	1	-	-
	35n46s	3	2	1	-	-
	35n47s	4	4	-	-	-

	35n48s	4	4	-	-	-
	35n49s	5	5	-	-	-
	35n50s	5	5	-	-	-
	35n51s	5	5	-	-	-
	35n52s	4	4	-	-	-
	35n53s	5	4	1	-	-
	35n54s	5	4	1	-	-
	35n55s	5	4	1	-	-
	35n56s	4	4	-	-	-
	35n57s	4	4	-	-	-
	35n58s	3	3	-	-	-
	35n59s	1	1	-	-	-
40n80s	40n42s	2	-	-	-	2
	40n43s	1	-	1	-	-
	40n44s	1	-	1	-	-
	40n45s	1	-	1	-	-
	40n46s	2	-	1	1	-
	40n47s	2	-	1	1	-
	40n48s	3	-	1	2	-
	40n49s	3	-	1	2	-
	40n50s	3	-	1	2	-
	40n51s	2	-	-	2	-
	40n52s	2	1	-	1	-
	40n53s	2	1	-	1	-
	40n54s	2	1	-	1	-
	40n55s	2	1	-	-	1
	40n56s	2	1	-	-	1
	40n57s	2	1	-	-	1
	40n58s	2	1	-	-	1
	40n59s	2	1	1	-	-
	40n60s	2	1	1	-	-
	40n61s	2	1	1	-	-
	40n62s	2	1	1	-	-
	40n63s	2	1	1	-	-
	40n64s	2	1	1	-	-
	40n65s	1	1	-	-	-
	40n66s	1	1	-	-	-
	40n67s	1	1	-	-	-
	40n68s	1	1	-	-	-
	40n69s	1	1	-	-	-
	40n70s	1	1	-	-	-
	40n71s	1	1	-	-	-
	40n72s	-	-	-	-	-
	40n73s	-	-	-	-	-
	40n74s	-	-	-	-	-
	40n75s	-	-	-	-	-

APPENDIX C

WORKING AND RESTORATION ROUTES

C.1 MINIMUM NUMBER OF REQUIRED ELIGIBLE WORKING AND RESTORATION ROUTES

Network family	Number of spans	Working routes required	Restoration routes required
15n30s1	15n16s	5WR	10RR
	15n17s	5WR	10RR
	15n18s	5WR	10RR
	15n19s	5WR	10RR
	15n20s	5WR	10RR
	15n21s	5WR	10RR
	15n22s	5WR	20RR
	15n23s	5WR	20RR
	15n24s	5WR	20RR
20n40s1	20n21s	5WR	10RR
	20n22s	5WR	10RR
	20n23s	5WR	10RR
	20n24s	5WR	10RR
	20n25s	5WR	10RR
	20n26s	15WR	15RR
	20n27s	5WR	10RR
	20n28s	5WR	10RR
	20n29s	15WR	15RR
	20n30s	5WR	10RR
	20n31s	15WR	15RR
	20n32s	15WR	15RR
	20n33s	20WR	20RR
	20n34s	20WR	20RR
	20n35s	20WR	20RR
25n50s1	25n27s	5WR	10RR
	25n28s	5WR	10RR
	25n29s	15WR	15RR
	25n30s	15WR	15RR
	25n31s	15WR	15RR
	25n32s	30WR	30RR
	25n33s	30WR	30RR
	25n34s	30WR	30RR
	25n35s	30WR	30RR
	25n36s	30WR	30RR
	25n37s	20WR	20RR
	25n38s	15WR	15RR

	25n39s	15WR	15RR
	25n40s	15WR	15RR
	25n41s	15WR	15RR
	25n42s	15WR	15RR
	25n43s	15WR	15RR
	25n44s	15WR	15RR
	25n45s	5WR	10RR
30n60s1	30n32s	5WR	10RR
	30n33s	5WR	10RR
	30n34s	5WR	10RR
	30n35s	15WR	15RR
	30n36s	30WR	30RR
	30n37s	30WR	30RR
	30n38s	20WR	20RR
	30n39s	15WR	15RR
	30n40s	15WR	15RR
	30n41s	15WR	15RR
	30n42s	30WR	30RR
	30n43s	30WR	30RR
	30n44s	45WR	45RR
	30n45s	60WR	60RR
	30n46s	60WR	60RR
	30n47s	60WR	60RR
	30n48s	5WR	10RR
	30n49s	5WR	10RR
	30n50s	5WR	10RR
	30n51s	15WR	15RR
	30n52s	10WR	10RR
	30n53s	10WR	10RR
	30n54s	15WR	15RR
	30n55s	15WR	15RR
35n70s1	35n37s	5WR	10RR
	35n38s	5WR	10RR
	35n39s	5WR	10RR
	35n40s	5WR	10RR
	35n41s	15WR	15RR
	35n42s	15WR	15RR
	35n43s	30WR	30RR
	35n44s	40WR	40RR
	35n45s	50WR	50RR
	35n46s	50WR	50RR
	35n47s	50WR	50RR
	35n48s	60WR	60RR
	35n49s	80WR	80RR
	35n50s	120WR	120RR
	35n51s	120WR	120RR
	35n52s	150WR	150RR
	35n53s	240WR	240RR
	35n54s	240WR	240RR
	35n55s	80WR	80RR

	35n56s	30WR	30RR
	35n57s	30WR	30RR
	35n58s	30WR	30RR
	35n59s	30WR	30RR
40n80s1	40n42s	5WR	10RR
	40n43s	5WR	10RR
	40n44s	5WR	10RR
	40n45s	15WR	15RR
	40n46s	15WR	15RR
	40n47s	5WR	10RR
	40n48s	15WR	15RR
	40n49s	5WR	10RR
	40n50s	15WR	15RR
	40n51s	15WR	15RR
	40n52s	15WR	15RR
	40n53s	20WR	20RR
	40n54s	20WR	20RR
	40n55s	20WR	20RR
	40n56s	15WR	15RR
	40n57s	15WR	15RR
	40n58s	15WR	15RR
	40n59s	20WR	20RR
	40n60s	15WR	15RR
	40n61s	15WR	15RR
	40n62s	15WR	15RR
	40n63s	15WR	15RR
	40n64s	15WR	15RR
	40n65s	10WR	10RR
	40n66s	5WR	10RR
	40n67s	5WR	10RR
	40n68s	5WR	10RR
	40n69s	5WR	10RR
	40n70s	15WR	15RR
	40n71s	15WR	15RR
	40n72s	15WR	15RR
	40n73s	15WR	15RR
	40n74s	15WR	15RR
	40n75s	30WR	10RR

APPENDIX D
AMPL MODELS

D.1 MM-DFMC

```
# ************************
# MM-DFMC.mod
# Joint Modular Span-Restorable Meta-Mesh Model under Dual-Failure Scenario
# March 2017 by Andres Castillo
# ************************

# ************************
# SETS
# ************************

set SPANS;
# set of all spans

set DIRECT_SPANS;
# set of spans that are not bypassed (i.e. not a part of any bypassed chain)

set BYPASS_SPANS;
# set of spans that act as bypasses for chains

set CHAIN_SPANS := {SPANS diff (DIRECT_SPANS union BYPASS_SPANS)};
# set of chain spans

set REST_ROUTES{i in SPANS};
# set of all restoration paths for each span failure i

set DEMANDS;
# set of all demand pairs or node pairs

set WORK_ROUTES{r in DEMANDS};
# set of all working routes for each demand pair r

# ************************
# PARAMETERS
# ************************

param Bypass{i in CHAIN_SPANS} symbolic;
param Cost{j in SPANS} default 1;
# cost of a unit of capacity on span j

param DemandUnits{r in DEMANDS} default 0;
# number of demand units between node pair r

param DeltaRestRoutes{i in SPANS, j in SPANS, p in REST_ROUTES[i]} default 0;
# equal to 1 if pth restoration route for failure of span i uses span j and 0 otherwise

param ZetaWorkRoutes{j in SPANS, r in DEMANDS, q in WORK_ROUTES[r]} default 0;
# equal to 1 if qth working route for demand between node pair r uses span k and 0 otherwise

param MaxFlow := sum {r in DEMANDS} DemandUnits[r];
# Used for upper bounds on flow and capacity variables.

# ************************
# VARIABLES
# ************************

var gwrkflow{r in DEMANDS, q in WORK_ROUTES[r]} >=0, <=10000;
# working capacity required by qth working route for demand between node pair r
```

```
var flowrest{i in SPANS, p in REST_ROUTES[i]} >=0, <=10000;
# restoration flow through pth restoration route for failure of span i

var flowrest_dual{i in SPANS, j in SPANS, p in REST_ROUTES[i]: i<>j} >= 0 integer, <=10000;
# restoration flow through pth restoration route for failure of span i when a span j has failed
simultaneously

var totalDwork >=0, <= (( sum{j in SPANS} Cost[j] ) * MaxFlow);
var totalCwork >=0, <= (( sum{j in SPANS} Cost[j] ) * MaxFlow);
var totalBwork >=0, <= (( sum{j in SPANS} Cost[j] ) * MaxFlow);
var totalspare >=0, <= (( sum{j in SPANS} Cost[j] ) * MaxFlow);

var work{j in SPANS} >=0, <=100000 integer;
# number of working links placed on span j

var spare{j in SPANS} >=0, <=100000 integer;
# number of spare links place on span j

# ***********************
# OBJECTIVE FUNCTION
# ***********************

minimize TotalCost: totalDwork + totalCwork + totalBwork + totalspare;
# minimize totcost: sum{j in SPANS} Cost[j] * (spare[j] + work[j]);

# ***********************
# CONSTRAINTS
# ***********************

subject to calculate_totalDwork:
totalDwork = sum{j in DIRECT_SPANS} work[j] * Cost[j];

subject to calculate_totalCwork:
totalCwork = sum{j in CHAIN_SPANS} work[j] * Cost[j];

subject to calculate_totalBwork:
totalBwork = sum{j in BYPASS_SPANS} work[j] * Cost[j];

subject to calculate_totalspare:
totalspare = sum{j in SPANS} spare[j] * Cost[j];


# Restoration of a Single-Failure with Chain-Wise Dual-Failure Scenario
subject to restn1{i in DIRECT_SPANS}:
       sum{p in REST_ROUTES[i]} flowrest[i,p] = work[i];

subject to restn12{i in CHAIN_SPANS, k in BYPASS_SPANS: k=Bypass[i]}:
       sum{p in REST_ROUTES[i]: DeltaRestRoutes[i,k,p] = 0} flowrest[i,p] = work[i];

subject to restn13{i in CHAIN_SPANS, k in BYPASS_SPANS: k=Bypass[i]}:
       sum{p in REST_ROUTES[k]: DeltaRestRoutes[k,i,p] = 0} flowrest[k,p] = work[k];

subject to sparasst1{i in DIRECT_SPANS, j in SPANS: i<>j}:
       spare[j] >=    sum{p in REST_ROUTES[i]} (DeltaRestRoutes[i,j,p] * flowrest[i,p]);

subject to sparasst12{i in CHAIN_SPANS, j in SPANS, k in BYPASS_SPANS: i<>j<>k and k=Bypass[i]}:
       spare[j] >=    sum{p in REST_ROUTES[i]} DeltaRestRoutes[i,j,p] * flowrest[i,p] +
                      sum{p in REST_ROUTES[k]} DeltaRestRoutes[k,j,p] * flowrest[k,p];


# Restoration of a Dual-Failure with Chain-Wise Logical Failure Scenarios
subject to restn2a{i in DIRECT_SPANS, j in DIRECT_SPANS: i<>j}:
       sum{p in REST_ROUTES[i]} flowrest_dual[i,j,p] = work[i];

subject to restn22a{i in CHAIN_SPANS, j in DIRECT_SPANS, k in BYPASS_SPANS: i<>j<>k and
k=Bypass[i]}:
       sum{p in REST_ROUTES[i]: DeltaRestRoutes[i,k,p] = 0} flowrest_dual[i,j,p] = work[i];

subject to restn22b{i in CHAIN_SPANS, j in DIRECT_SPANS, k in BYPASS_SPANS: i<>j<>k and
k=Bypass[i]}:
       sum{p in REST_ROUTES[j]: DeltaRestRoutes[j,k,p] = 0} flowrest_dual[j,i,p] = work[j];
```

```
subject to restn22c{i in CHAIN_SPANS, j in DIRECT_SPANS, k in BYPASS_SPANS: i<>j<>k and
k=Bypass[i]}:
        sum{p in REST_ROUTES[k]: DeltaRestRoutes[k,i,p] = 0} flowrest_dual[k,j,p] = work[k];

subject to restn23a{i in CHAIN_SPANS, j in CHAIN_SPANS, k in BYPASS_SPANS, l in BYPASS_SPANS:
i<>j and k<>l and k=Bypass[i] and l=Bypass[j]}:
        sum{p in REST_ROUTES[i]: (DeltaRestRoutes[i,k,p] = 0 and DeltaRestRoutes[i,l,p] = 0)}
        flowrest_dual[i,j,p] = work[i];

subject to restn23c{i in CHAIN_SPANS, j in CHAIN_SPANS, k in BYPASS_SPANS, l in BYPASS_SPANS:
i<>j and k<>l and k=Bypass[i] and l=Bypass[j]}:
        sum{p in REST_ROUTES[k]: (DeltaRestRoutes[k,i,p] = 0 and DeltaRestRoutes[k,l,p] = 0)}
        flowrest_dual[k,j,p] = work[k];

subject to restn23d{i in CHAIN_SPANS, j in CHAIN_SPANS, k in BYPASS_SPANS, l in BYPASS_SPANS:
i<>j and k<>l and k=Bypass[i] and l=Bypass[j]}:
        sum{p in REST_ROUTES[l]: (DeltaRestRoutes[l,j,p] = 0 and DeltaRestRoutes[l,k,p] = 0)}
        flowrest_dual[l,i,p] = work[l];

subject to sparasst2{i in DIRECT_SPANS, j in DIRECT_SPANS, k in SPANS: i<>j<>k}:
        spare[k] >=
        sum{p in REST_ROUTES[i]} (flowrest_dual[i,j,p] * DeltaRestRoutes[i,k,p]) +
        sum{p in REST_ROUTES[j]} (flowrest_dual[j,i,p] * DeltaRestRoutes[j,k,p]);

subject to sparasst22{i in CHAIN_SPANS, j in DIRECT_SPANS, l in SPANS, k in BYPASS_SPANS:
i<>j<>l<>k and k=Bypass[i]}:
        spare[l] >=
        sum{p in REST_ROUTES[i]} (flowrest_dual[i,j,p] * DeltaRestRoutes[i,l,p]) +
        sum{p in REST_ROUTES[j]} (flowrest_dual[j,i,p] * DeltaRestRoutes[j,l,p]) +
        sum{p in REST_ROUTES[k]} (flowrest_dual[k,j,p] * DeltaRestRoutes[k,l,p]);

subject to sparasst23{i in CHAIN_SPANS, j in CHAIN_SPANS, w in SPANS, k in BYPASS_SPANS, l in
BYPASS_SPANS: i<>j<>k<>l<>w and k=Bypass[i] and l=Bypass[j]}:
        spare[w] >=
        sum{p in REST_ROUTES[i]} (flowrest_dual[i,j,p] * DeltaRestRoutes[i,w,p]) +
        sum{p in REST_ROUTES[j]} (flowrest_dual[j,i,p] * DeltaRestRoutes[j,w,p]) +
        sum{p in REST_ROUTES[k]} (flowrest_dual[k,j,p] * DeltaRestRoutes[k,w,p]) +
        sum{p in REST_ROUTES[l]} (flowrest_dual[l,i,p] * DeltaRestRoutes[l,w,p]);

subject to limit1a{i in DIRECT_SPANS, j in DIRECT_SPANS: i<>j}:
        sum{p in REST_ROUTES[i]: DeltaRestRoutes[i,j,p] = 1} flowrest_dual[i,j,p] = 0;

subject to limit2a{i in CHAIN_SPANS, j in DIRECT_SPANS, k in BYPASS_SPANS: i<>j<>k and
k=Bypass[i]}:
        sum{p in REST_ROUTES[i]: DeltaRestRoutes[i,j,p] = 1} flowrest_dual[i,j,p] = 0;

subject to limit2b{i in CHAIN_SPANS, j in DIRECT_SPANS, k in BYPASS_SPANS: i<>j<>k and
k=Bypass[i]}:
        sum{p in REST_ROUTES[j]: DeltaRestRoutes[j,i,p] = 1} flowrest_dual[j,i,p] = 0;

subject to limit2c{i in CHAIN_SPANS, j in DIRECT_SPANS, k in BYPASS_SPANS: i<>j<>k and
k=Bypass[i]}:
        sum{p in REST_ROUTES[k]: DeltaRestRoutes[k,j,p] = 1} flowrest_dual[k,j,p] = 0;

subject to limit3a{i in CHAIN_SPANS, j in CHAIN_SPANS, k in BYPASS_SPANS, l in BYPASS_SPANS: i<>j
and k<>l and k=Bypass[i] and l=Bypass[j]}:
        sum{p in REST_ROUTES[i]: DeltaRestRoutes[i,j,p] = 1} flowrest_dual[i,j,p] = 0;

subject to limit3c{i in CHAIN_SPANS, j in CHAIN_SPANS, k in BYPASS_SPANS, l in BYPASS_SPANS: i<>j
and k<>l and k=Bypass[i] and l=Bypass[j]}:
        sum{p in REST_ROUTES[k]: DeltaRestRoutes[k,j,p] = 1} flowrest_dual[k,j,p] = 0;

subject to limit3d{i in CHAIN_SPANS, j in CHAIN_SPANS, k in BYPASS_SPANS, l in BYPASS_SPANS: i<>j
and k<>l and k=Bypass[i] and l=Bypass[j]}:
        sum{p in REST_ROUTES[l]: DeltaRestRoutes[l,i,p] = 1} flowrest_dual[l,i,p] = 0;

subject to demmet{r in DEMANDS}:
        sum{q in WORK_ROUTES[r]} gwrkflow[r,q] = DemandUnits[r];

subject to workasst{j in SPANS}:
        work[j] = sum{r in DEMANDS, q in WORK_ROUTES[r]} ZetaWorkRoutes[j,r,q] * gwrkflow[r,q];
```

D.2 MM-DFMR

```
# ***********************
# MM-DFMR.mod
# Meta-Mesh Maximum Dual Failure Restorability
# May 2017 by Andres Castillo
# ***********************

# ***********************
# SETS
# ***********************

set SPANS;
# set of all spans

set DIRECT_SPANS;
# set of spans that are not bypassed (i.e. not a part of any bypassed chain)

set BYPASS_SPANS;
# set of spans that act as bypasses for chains

set CHAIN_SPANS := {SPANS diff (DIRECT_SPANS union BYPASS_SPANS)};
# set of chain spans

set REST_ROUTES{i in SPANS};
# set of all restoration paths for each span failure i

set DEMANDS;
# set of all demand pairs or node pairs

set WORK_ROUTES{r in DEMANDS};
# set of all working routes for each demand pair r

# ***********************
# PARAMETERS
# ***********************

param Bypass{i in CHAIN_SPANS} symbolic;
param Cost{j in SPANS} default 1;
# cost of a unit of capacity on span j

param DemandUnits{r in DEMANDS} default 0;
# number of demand units between node pair r

param DeltaRestRoutes{i in SPANS, j in SPANS, p in REST_ROUTES[i]} default 0;
# equal to 1 if pth restoration route for failure of span i uses span j and 0 otherwise

param ZetaWorkRoutes{j in SPANS, r in DEMANDS, q in WORK_ROUTES[r]} default 0;
# equal to 1 if qth working route for demand between node pair r uses span k and 0 otherwise.

param MaxFlow := sum {r in DEMANDS} DemandUnits[r];
# Used for upper bounds on flow and capacity variables.

param Budget;
# budget limit for dual failure restoration

# ***********************
# VARIABLES
# ***********************

var gwrkflow{r in DEMANDS, q in WORK_ROUTES[r]} >=0, <=10000;
# working capacity required by qth working route for demand between node pair r

var flowrest{i in SPANS, p in REST_ROUTES[i]} >=0, <=10000;
# restoration flow through pth restoration route for failure of span i
```

```
var flowrest_dual{i in SPANS, j in SPANS, p in REST_ROUTES[i]: i<>j} >= 0 integer, <=10000;
# restoration flow through pth restoration route for failure of span i when a span j has failed
simultaneously

var totalDwork >=0, <= (( sum{j in SPANS} Cost[j] ) * MaxFlow);
var totalCwork >=0, <= (( sum{j in SPANS} Cost[j] ) * MaxFlow);
var totalBwork >=0, <= (( sum{j in SPANS} Cost[j] ) * MaxFlow);
var totalspare >=0, <= (( sum{j in SPANS} Cost[j] ) * MaxFlow);

var non_restored{i in SPANS, j in SPANS: i<>j} >=0 integer;
# number of non-restored working capacities under dual failure (i, j)

var work{j in SPANS} >=0, <=100000 integer;
# number of working links placed on span j

var spare{j in SPANS} >=0, <=100000 integer;
# number of spare links place on span j

# ***********************
# OBJECTIVE FUNCTION
# ***********************

minimize tot_non_restored:
sum{i in SPANS, j in SPANS: i<>j} non_restored[i,j];
# minimize number of non-restored working capacities under all dual failure scenarios

# ***********************
# CONSTRAINTS
# ***********************

subject to calculate_totalDwork:
totalDwork = sum{j in DIRECT_SPANS} work[j] * Cost[j];

subject to calculate_totalCwork:
totalCwork = sum{j in CHAIN_SPANS} work[j] * Cost[j];

subject to calculate_totalBwork:
totalBwork = sum{j in BYPASS_SPANS} work[j] * Cost[j];

subject to calculate_totalspare:
totalspare = sum{j in SPANS} spare[j] * Cost[j];

subject to NWC2{i in DIRECT_SPANS, j in DIRECT_SPANS: i<>j}:
	non_restored[i,j] = work[i] + work[j] - sum{p in REST_ROUTES[i]} flowrest_dual[i,j,p] -
	sum{p in REST_ROUTES[j]} flowrest_dual[j,i,p];

subject to NWC2_1{i in CHAIN_SPANS, j in DIRECT_SPANS, k in BYPASS_SPANS: k=Bypass[i]}:
	non_restored[i,j] = work[i] + work[j] + work[k] -
	sum{p in REST_ROUTES[i]: DeltaRestRoutes[i,k,p] = 0} flowrest_dual[i,j,p] -
	sum{p in REST_ROUTES[j]: DeltaRestRoutes[j,k,p] = 0} flowrest_dual[j,i,p] -
	sum{p in REST_ROUTES[k]: DeltaRestRoutes[k,i,p] = 0} flowrest_dual[k,j,p];

subject to NWC2_2{i in CHAIN_SPANS, j in CHAIN_SPANS, k in BYPASS_SPANS, l in BYPASS_SPANS: i<>j
and k<>l and k=Bypass[i] and l=Bypass[j]}:
	non_restored[i,j] = work[i] + work[j] + work[k] + work[l] -
	sum{p in REST_ROUTES[i]: (DeltaRestRoutes[i,k,p] = 0 and DeltaRestRoutes[i,l,p] = 0)}
	flowrest_dual[i,j,p] -
	sum{p in REST_ROUTES[j]: (DeltaRestRoutes[j,l,p] = 0 and DeltaRestRoutes[j,k,p] = 0)}
	flowrest_dual[j,i,p] -
	sum{p in REST_ROUTES[k]: (DeltaRestRoutes[k,i,p] = 0 and DeltaRestRoutes[k,l,p] = 0)}
	flowrest_dual[k,j,p] -
	sum{p in REST_ROUTES[l]: (DeltaRestRoutes[l,j,p] = 0 and DeltaRestRoutes[l,k,p] = 0)}
	flowrest_dual[l,i,p];

# Restoration of a Single-Failure with Chain-Wise Dual-Failure Scenario
subject to restn1{i in DIRECT_SPANS}:
	sum{p in REST_ROUTES[i]} flowrest[i,p] = work[i];

subject to restn12{i in CHAIN_SPANS, k in BYPASS_SPANS: k=Bypass[i]}:
	sum{p in REST_ROUTES[i]: DeltaRestRoutes[i,k,p] = 0} flowrest[i,p] = work[i];
```

```
subject to restn13{i in CHAIN_SPANS, k in BYPASS_SPANS: k=Bypass[i]}:
        sum{p in REST_ROUTES[k]: DeltaRestRoutes[k,i,p] = 0} flowrest[k,p] = work[k];

subject to sparasst1{i in DIRECT_SPANS, j in SPANS: i<>j}:
        spare[j] >= sum{p in REST_ROUTES[i]} (DeltaRestRoutes[i,j,p] * flowrest[i,p]);

subject to sparasst12{i in CHAIN_SPANS, j in SPANS, k in BYPASS_SPANS: i<>j<>k and k=Bypass[i]}:
        spare[j] >= sum{p in REST_ROUTES[i]} DeltaRestRoutes[i,j,p] * flowrest[i,p] +
        sum{p in REST_ROUTES[k]} DeltaRestRoutes[k,j,p] * flowrest[k,p];

# Restoration of a Dual-Failure with Chain-Wise Triple-Logical Failure Scenario
subject to restn2a{i in DIRECT_SPANS, j in DIRECT_SPANS: i<>j}:
        sum{p in REST_ROUTES[i]} flowrest_dual[i,j,p] <= work[i];

subject to restn22a{i in CHAIN_SPANS, j in DIRECT_SPANS, k in BYPASS_SPANS: k=Bypass[i]}:
        sum{p in REST_ROUTES[i]: DeltaRestRoutes[i,k,p] = 0} flowrest_dual[i,j,p] <= work[i];

subject to restn22b{i in CHAIN_SPANS, j in DIRECT_SPANS, k in BYPASS_SPANS: k=Bypass[i]}:
        sum{p in REST_ROUTES[j]: DeltaRestRoutes[j,k,p] = 0} flowrest_dual[j,i,p] <= work[j];

subject to restn22c{i in CHAIN_SPANS, j in DIRECT_SPANS, k in BYPASS_SPANS: k=Bypass[i]}:
        sum{p in REST_ROUTES[k]: DeltaRestRoutes[k,i,p] = 0} flowrest_dual[k,j,p] <= work[k];

subject to restn23a{i in CHAIN_SPANS, j in CHAIN_SPANS, k in BYPASS_SPANS, l in BYPASS_SPANS:
i<>j and k<>l and k=Bypass[i] and l=Bypass[j]}:
        sum{p in REST_ROUTES[i]: (DeltaRestRoutes[i,k,p] = 0 and DeltaRestRoutes[i,l,p] = 0)}
        flowrest_dual[i,j,p] <= work[i];

subject to restn23c{i in CHAIN_SPANS, j in CHAIN_SPANS, k in BYPASS_SPANS, l in BYPASS_SPANS:
i<>j and k<>l and k=Bypass[i] and l=Bypass[j]}:
        sum{p in REST_ROUTES[k]: (DeltaRestRoutes[k,i,p] = 0 and DeltaRestRoutes[k,l,p] = 0)}
        flowrest_dual[k,j,p] <= work[k];

subject to restn23d{i in CHAIN_SPANS, j in CHAIN_SPANS, k in BYPASS_SPANS, l in BYPASS_SPANS:
i<>j and k<>l and k=Bypass[i] and l=Bypass[j]}:
        sum{p in REST_ROUTES[l]: (DeltaRestRoutes[l,j,p] = 0 and DeltaRestRoutes[l,k,p] = 0)}
        flowrest_dual[l,i,p] <= work[l];

subject to sparasst2{i in DIRECT_SPANS, j in DIRECT_SPANS, k in SPANS: i<>j<>k}:
        spare[k] >=
        sum{p in REST_ROUTES[i]} (flowrest_dual[i,j,p] * DeltaRestRoutes[i,k,p]) +
        sum{p in REST_ROUTES[j]} (flowrest_dual[j,i,p] * DeltaRestRoutes[j,k,p]);

subject to sparasst22{i in CHAIN_SPANS, j in DIRECT_SPANS, l in SPANS, k in BYPASS_SPANS:
i<>j<>l<>k and k=Bypass[i]}:
        spare[l] >=
        sum{p in REST_ROUTES[i]} (flowrest_dual[i,j,p] * DeltaRestRoutes[i,l,p]) +
        sum{p in REST_ROUTES[j]} (flowrest_dual[j,i,p] * DeltaRestRoutes[j,l,p]) +
        sum{p in REST_ROUTES[k]} (flowrest_dual[k,j,p] * DeltaRestRoutes[k,l,p]);

subject to sparasst23{i in CHAIN_SPANS, j in CHAIN_SPANS, w in SPANS, k in BYPASS_SPANS, l in
BYPASS_SPANS: i<>j<>k<>l<>w and k=Bypass[i] and l=Bypass[j]}:
        spare[w] >= sum{p in REST_ROUTES[i]} (flowrest_dual[i,j,p] * DeltaRestRoutes[i,w,p]) +
        sum{p in REST_ROUTES[j]} (flowrest_dual[j,i,p] * DeltaRestRoutes[j,w,p]) +
        sum{p in REST_ROUTES[k]} (flowrest_dual[k,j,p] * DeltaRestRoutes[k,w,p]) +
        sum{p in REST_ROUTES[l]} (flowrest_dual[l,i,p] * DeltaRestRoutes[l,w,p]);


subject to limit1a{i in DIRECT_SPANS, j in DIRECT_SPANS: i<>j}:
        sum{p in REST_ROUTES[i]: DeltaRestRoutes[i,j,p] = 1} flowrest_dual[i,j,p] = 0;

subject to limit2a{i in CHAIN_SPANS, j in DIRECT_SPANS, k in BYPASS_SPANS: k=Bypass[i]}:
        sum{p in REST_ROUTES[i]: DeltaRestRoutes[i,j,p] = 1} flowrest_dual[i,j,p] = 0;

subject to limit2b{i in CHAIN_SPANS, j in DIRECT_SPANS, k in BYPASS_SPANS: k=Bypass[i]}:
        sum{p in REST_ROUTES[j]: DeltaRestRoutes[j,i,p] = 1} flowrest_dual[j,i,p] = 0;

subject to limit2c{i in CHAIN_SPANS, j in DIRECT_SPANS, k in BYPASS_SPANS: k=Bypass[i]}:
        sum{p in REST_ROUTES[k]: DeltaRestRoutes[k,j,p] = 1} flowrest_dual[k,j,p] = 0;
```

```
subject to limit3a{i in CHAIN_SPANS, j in CHAIN_SPANS, k in BYPASS_SPANS, l in BYPASS_SPANS: i<>j
and k<>l and k=Bypass[i] and l=Bypass[j]}:
        sum{p in REST_ROUTES[i]: DeltaRestRoutes[i,j,p] = 1} flowrest_dual[i,j,p] = 0;

subject to limit3c{i in CHAIN_SPANS, j in CHAIN_SPANS, k in BYPASS_SPANS, l in BYPASS_SPANS: i<>j
and k<>l and k=Bypass[i] and l=Bypass[j]}:
        sum{p in REST_ROUTES[k]: DeltaRestRoutes[k,j,p] = 1} flowrest_dual[k,j,p] = 0;

subject to limit3d{i in CHAIN_SPANS, j in CHAIN_SPANS, k in BYPASS_SPANS, l in BYPASS_SPANS: i<>j
and k<>l and k=Bypass[i] and l=Bypass[j]}:
        sum{p in REST_ROUTES[l]: DeltaRestRoutes[l,i,p] = 1} flowrest_dual[l,i,p] = 0;

subject to demmet{r in DEMANDS}:
        sum{q in WORK_ROUTES[r]} gwrkflow[r,q] = DemandUnits[r];

subject to workasst{j in SPANS}:
        work[j] = sum{r in DEMANDS, q in WORK_ROUTES[r]} ZetaWorkRoutes[j,r,q] * gwrkflow[r,q];

subject to Restriction: totalDwork + totalCwork + totalBwork + totalspare <= Budget;
```