Reliable Route Design Schemes in Coding Capable Networks

Abu Hena Al Muktadir

Department of Information and Communication Engineering

The University of Electro-Communications

A thesis submitted for the degree of

Doctor of Philosophy

September 2014

APPROVED BY SUPERVISORY COMMITTEE:

Chairperson: Prof. Eiji Oki

Member: Prof. Naoto Kishi

Member: Prof. Yasushi Yamao

Member: Prof. Hiro Ito

Member: Prof. Minoru Terada

Date of the final-defense: 2014/07/30

概要

高信頼な通信がネットワークの故障に対する耐久性を確保し、貴重なデータ損失を防ぐことを保証する。最も高い信頼度クラスは，故障時に無瞬断で復旧する瞬時復旧を提供する。すなわち、遅延が無しに 100％のデータをリカバリーすることである。１＋１プロテクションと言う動的なプロテクションでは、同時にデータを二つの独立パスに送ることによりシングルリンクの故障に対する瞬時復旧することを提供する。故障時において、着ノードが予備パスに切り替えのみで瞬時復旧が実現させる。資源を有効利用する従来の復旧技術は，パスの両端の切り替えが必要でなため，瞬時復旧を達成できない。一方、1+1 プロテクションは，現用パスの 2 倍のネットワーク資源を要するという問題がある。本研究では、１＋１プロテクションの利点を保つ上で必要資源を最小化することに目指す。

現在、ネットワーク符号化が活用したシナリオは２つがあり、それらより１＋１プロテクションの瞬時復旧を維持しながら必要な資源を削減することができる。その２つシナリオはＭＳＣＤ（multiple sources and a common destination）と TS (traffic splitting)である。しかし、従来の研究はこの二つのシナリオでネットワーク符号化により最適な経路制御方法についてまだ述べていない。

本研究では MSCD と TS シナリオごとに、瞬時回復を提供する 1+1 プロテクションの経路設計法について論じる。本研究の貢献は，２つの部分に大別される。第一に，符号化可能なネットワークにおいて，瞬時回復を提供する 1+1 プロテクションに必要なネットワーク資源を最小化する経路を求める最適化問題を，整数線形計画法（ILP: integer linear programming）のモデルとして定式化する。第 2 に ILP モデルによるアプローチの問題を解決するために，大規模ネットワークに対して発見的なアルゴリズムを導入する。

提案アプローチの性能を評価するために従来の１＋１プロテクションクションと比較し、MSCD と TS シナリオの場合で要するネットワーク資源を１５%また２０％削減ができるという結果が得られた。

Abstract

Reliable communication ensures protection against network component failure, and prevents valuable data losses. There are several service grades for reliable communication. The highest grade in reliability is instantaneous recovery from any failure, i.e., recovery with negligible delay. The next service grade is to ensure protection against failure using less resources at the cost of increased recovery time.

The protection techniques can be broadly classified as either predesigned (proactive) protection or dynamic restoration techniques. Protection from a failure is achieved by providing backup paths for the working paths, where resources along the backup paths may be shared. 1+1 protection, a proactive protection technique, provides instantaneous receiver-initiated recovery from any single link failure by duplicating and sending the same source data onto two disjoint paths. Instantaneous recovery is achieved because only the destination node switches to the backup path after a failure is detected. Other resource efficient recovery techniques to deal with single link failure require switching operations at least at both ends, which restrict instantaneous recovery. However, the 1+1 protection technique demands at least double network resources. Our goal is to minimize the resources required for 1+1 protection while maintaining the advantage of instantaneous recovery.

There are two coding aware scenarios that can reduce the required network resources for 1+1 protection while maintaining instantaneous recovery. The two scenarios are named as MSCD (multiple source and a common destination) and TS (traffic splitting), respectively. In the

MSCD scenario data from different sources are encoded at the intermediate nodes (referred to as the network coding (NC) technique). In the TS scenario, at first traffic is split, and then encoding is performed with the split parts at the source node (erasure correcting code). No previous research has addressed the optimum coding aware route design issues for these two scenarios.

This thesis addresses reliable route design problems in coding capable networks. For each of the MSCD and TS scenarios, we propose a coding aware 1+1 protection route design scheme for instantaneous recovery. For both proposed schemes, our contributions consist of two parts. First, we formulate the optimization (route design) problem, where an optimum coding aware set of routes that minimizes the required network resources for 1+1 protection are determined, as an Integer Linear Programming (ILP) formulation. Second, in order to design coding aware 1+1 protection routes for all possible source destination pairs, a heuristic routing algorithm is presented where the ILP model is used iteratively.

In order to evaluate the effectiveness and applicability, the performance of our two proposed route design schemes are evaluated in terms of the total resource saving with respect to the conventional 1+1 protection technique. Numerical results observed that the proposed routing scheme with MSCD scenario achieves almost 15% of resource saving, while the routing scheme with the TS scenario achieves almost 20% of resource saving w.r.t. the conventional 1+1 protection cost (without coding) in our examined networks. The applicability of both proposed schemes depend on the network characteristics, specially on the node connectivity. In terms of resource saving the scheme with the TS scenario is preferred, but it requires high node connectivity for both the source and the destination. For the scheme with the MSCD scenario, resource saving is smaller than that of the other proposed scheme, but it requires minimum node connectivity of two between the source and the destination.

To my parents

Acknowledgements

This thesis is the summary of my doctoral study at the University of Electro-Communications, Tokyo, Japan. I am grateful to a large number of people who have helped me to accomplish this work.

First of all, I would like to express my gratefulness to my advisor Professor Eiji Oki, who gave me a chance for my study, inspires me many things, and gives me useful timely directions and encouragements for my work.

I would like to thank Professor Naoto Kishi, Professor Yasushi Yamao, Professor Hiro Ito, and Professor Minoru Terada for their valuable comments to improve my thesis and being part of my judging committee.

Finally, I want to thank my lovely parents, wife, and elder brother. Also thank for help to all of my friends in Japan.

Acronyms

$2SD$	An MSCD scenario with two sources
$3SD$	An MSCD scenario with three sources
BRITE	Boston University Representative Internet Topology generator
G_{NC}	Network coding gain
ILP	Integer linear programming
IQP	Integer quadratic programming
kSD	An MSCD scenario with k sources
MSCD	Multiple sources and a common destination
LP	Linear programming
NC	Network coding
NIC	Network interface card
SONET	Synchronize optical network
TS	Traffic splitting

ACRONYMS

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Ensuring protection against network component failure, and instantaneous recovery from failure are the demanding quality of service (QoS) requirements that the network operators must offer to the customers needing reliable communication. We envision a pre-planned instantaneous recovery protection technique from any single link failure in the network, where only the destination node is involved in the recovery operation after a failure is detected. Among the network failure events a single link failure is the most frequent type, and a number of techniques to deal with this issue have been addressed. The protection techniques can be broadly classified as either predesigned (pre-planned) protection or dynamic restoration techniques [1, 2].

Instantaneous recovery from failure means that there is no loss of data due to a failure. In other words, we can say that *instantaneous recovery* refers to exactly 100% recovery of data or services affected by any failure.

1+1 protection is a predesigned proactive protection technique [1] that provides instantaneous recovery from any single link failure in the network, as illustrated in Fig. 1.1. In this simple protection technique, copies of the same data are sent from the source node, S, to the destination node, D, via two disjoint paths, paths having no link in common. If a link failure makes one path unavailable (say path 1), the destination node detects this failure, and quickly switches over to the other path to maintain uninterrupted communication and instantaneous recovery.

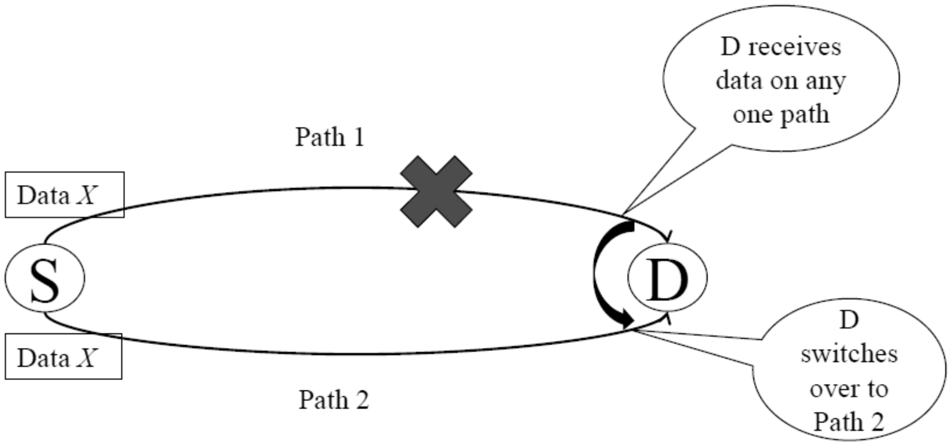


Figure 1.1: 1+1 protection technique achieves instantaneous receiver initiated recovery.

1.1 Necessity of 1+1 protection for instantaneous recovery

Protection from a failure is achieved by providing backup paths for the working paths, where resources along the backup paths may be shared. Shared backup path protection (SBPP) [3, 4, 5], and 1:1 protection are pre-planned protection techniques to deal with single link failure. Path or span restoration [6] is a dynamic restoration technique for single link failure. Both SBPP and dynamic restoration allow backup resource sharing and are resource efficient. 1:1 protection does not allow backup sharing, but the backup paths may be used for low priority data communication. Similar to the 1:1 protection technique, in SBPP a pair of working and protection paths (mutually disjoint) is pre-planned between the two end nodes. Once the two end nodes of the working path are informed of a network failure, they perform the actions of switching to the backup path from the working path. SBPP allows protection paths to share spare capacity on their common spans as long as their corresponding working paths do not share any common link or node. Upon a link failure, the dynamic restoration technique performs restoration at the end two nodes of a failed link, which diverts all the affected service flows locally onto the routes that bypass the failed link.

There are several service grades for reliable communications. The highest grade in reliability is instantaneous recovery from any failure, for which the 1+1 protection technique is appropriate. The next service grade is to ensure protection against failure using less resources at the cost of increased recovery time. In the 1:1 protection technique recovery time is higher than that of the 1+1 protection. SBPP and dynamic restoration are resource efficient techniques from the resource usage point of view. However, instantaneous recovery is not possible in the SBPP, 1:1 protection, and dynamic restoration techniques. In SBPP and 1:1 protection, switching operations at least at both ends are required for the recovery operation, while in dynamic restoration switching operations are required at two nodes corresponding to the failed path or link after finding the recovered route. The necessity of switching at least at two nodes restricts instantaneous recovery of failed data. Thus, these three techniques are not suitable to achieve our objective of instantaneous recovery.

Among the several service grades, we focus on instantaneous recovery. Only the 1+1 protection technique enables instantaneous receiver-initiated recovery from any single failure, and is perfectly suited for our objective. This feature of instantaneous recovery is achieved at the expense of at least double network resources. From a network operators' point of view, it is desirable that the employed protection technique uses less amount of network resources.

1.2 Scenarios for resource saving in 1+1 protection

There are two coding aware scenarios that can reduce the backup resources required for 1+1 protection while maintaining the instantaneous recovery advantage. The two scenarios are described in the following.

1.2.1 Multiple sources and a common destination (MSCD) scenario

Adaptation of the network coding (NC) technique [12] is a way to reduce the required resources in 1+1 protection [13]. In the NC technique an intermediate

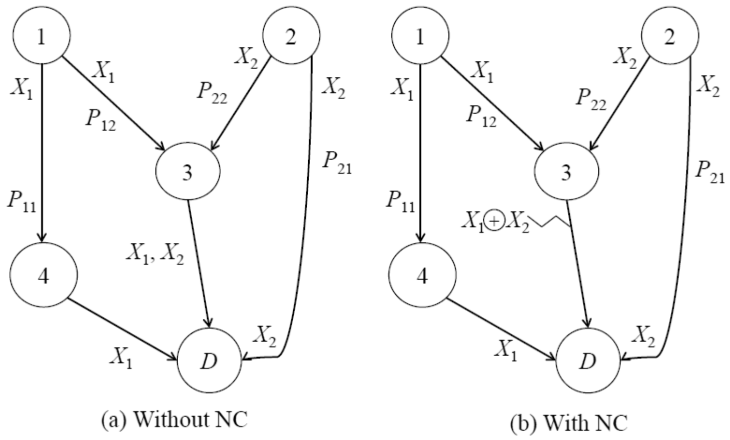


Figure 1.2: Illustration of 1+1 protection with and without NC.

node (neither a source nor a destination node in a network) receives several data streams (plain data) on its incoming links, merges them into a single data stream, and sends the combined stream (encoded data) onto its output link. The data stream on the output link from an intermediate node is a linear combination of the data streams on its input links.

Figure 1.2 illustrates how NC is applied to 1+1 protection in any scenario with two sources and a common destination ($2SD$) [13], which we denote as *MSCD scenario*[1]. In the network, source nodes 1 and 2 send their data X_1 and X_2, respectively, on two disjoint paths ($P_{11} = 1-4-D$, $P_{12} = 1-3-D$ for node 1, and $P_{21} = 2-D$, $P_{22} = 2-3-D$ for node 2) to common destination node D. Among the four paths mentioned above, link $3-D$ is shared by two paths (P_{12} and P_{22}). Node 3 is capable of applying NC. We name the node applying NC as the *NC node*. Assume that data rates of X_1 and X_2 are the same, and the cost of using a link by either X_1 or X_2 is 1. Without NC the total network resource utilization cost becomes 7 in Fig. 1.2(a), because link $3-D$ is individually used by both X_1 and X_2. In order to perform NC simple bitwise Exclusive-OR (XOR) operation is used. If NC is applied at node 3, $X_1 \oplus X_2$ becomes one single data stream and the cost for using link $3-D$ becomes 1, and total cost equals 6 in Fig. 1.2(b). One of the paths, suppose $P_{11} = 1-4-D$ is failed. Destination

[1]In general MSCD scenario consists of $k \geq 2$ sources and a common destination.

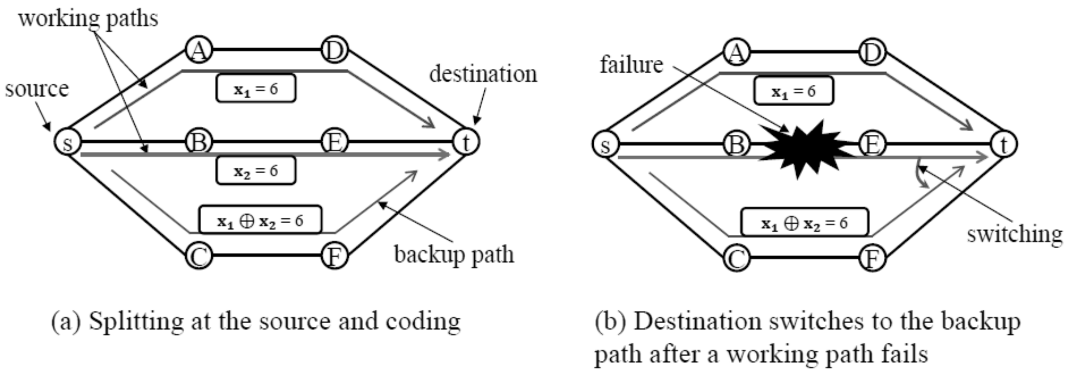


Figure 1.3: Illustration of 1+1 protection with traffic splitting.

node D recovers X_1 by performing bitwise XOR operation on the two received data, X_2 and $X_1 \oplus X_2$.

1.2.2 Traffic splitting (TS) scenario

Another way to reduce resource requirement in 1+1 protection is to use traffic splitting and coding for recovery of failed data as illustrated in *traffic splitting (TS)* scenario of Fig. 1.3 [20]. In Fig. 1.3(a), a single unicast flow with a capacity requirement of 12 is split into two sub-flows of equal size (6), and they are sent to the destination through two disjoint paths. The two split parts are coded into a single part of equal size (6) at the source and sent through another disjoint path to the destination. In this way backup path is shared by both the working paths. Due to sharing total capacity requirements yields 18, where without splitting i.e., with conventional 1+1 protection the capacity requirement is 24. If a failure occurs, instantaneous recovery is performed at the destination by performing one decoding operation on the received data. Since coding is performed at the source, the coding technique is referred to as erasure correcting code [54]. For encoding and decoding bitwise XOR operation is used.

Table 1.1 below compares the features of MSCD and TS scenarios.

Table 1.1: Comparison of MSCD and TS scenarios

Features	MSCD	TS
Requirement	Two or more sources and a common destination	A source destination pair only
Minimum node connectivity	Two	$\geq$ Three
Coding is performed at	Intermediate node(s)	Source node
Traffic splitting	No	Yes
Content is performed by using data belonging to	Different sources	Same source
Decoding is performed on	All received data	All received data

1.3 Related previous works

For the last few years, NC based protection techniques have been studied due to its ability to reduce required backup resources for protection. Provisioning of protection techniques against component failure demand high network resources and the NC technique reduces resource utilization. The topic of the application of NC for protecting network failure has been addressed in many research contributions [13, 14, 15, 16, 17, 18, 19, 20].

A.E. Kamal *et al.* introduced a cost efficient NC based $1 + N$ protection scheme in [14, 15] for any single link failure on N working paths, and extended this scheme for protection against multiple link failures in [16]. In these three papers, Integer Linear Programming (ILP) formulations were presented to evaluate the cost of using $1 + N$ protection. The cost obtained by those ILP formulations were compared to that of the 1+1 protection *without NC*. In [17] an NC based protection technique, known as network protection codes, was adopted for self-healing protection against multiple link failures at the cost of reduced capacity. There are other research works that addressed NC based protection techniques from network failure [18, 19, 20]. [18] suggests the use of NC against multiple link failures in multi-domain networks, where combination of 1+1 protection and dual homing [21] serve the purpose of protection. 1+1 protection demands high network resources but the node degree requirement is only two. $1+N$ protection is more resource efficient than 1+1 protection, but the node degree requirement in this case is N. In [19] a new protection mechanism was adopted which creates a balance between the capacity and node degree requirements. Rouayheb *et al.* suggested an algorithm for designing proper network code to protect against any single link failure [20]. Link capacity is considered as an important factor in [20]. The issues of the reliability of two-path protection, and how to achieve the maximum reliability are addressed in [22].

The NC technique is also used in optical networks [23, 24] and wireless networks [25, 26, 27, 28, 29, 30, 31, 32, 33, 34]. A heuristic algorithm to find two link disjoint multicast trees for protection using the NC technique in all optical networks was introduced in [23]. Manley *et al.* also formulated an ILP formulation to determine an optimal network code, for protecting a single-unit transmission against a single link failure, and introduced path disjoint constraints in [23]. Lun *et al.* [25] formulated a linear optimization problem for finding minimum cost sub-graphs for single and multiple multicast connections considering the cost and delay associated with the network resource uses. The work [25] was also extended for multicast wireless networks. To minimize the bandwidth requirement and protect against failure in IPTV multicast scenario using the NC technique, a mixed integer programming (MIP) formulation was presented in [26]. In [27] the possible NC scenarios were systematically analyzed and the generalized coding conditions to ensure decoding ability at the destinations were developed. [28] is one of the first works to illustrate the application of NC in wireless networks. [29] suggests that routing in wireless networks should be aware of NC opportunities. It also provides mathematical formulations to evaluate the throughput of NC in any wireless network topology. NC aware unicast and multicast routing strategies are discussed in [30, 31, 32, 33, 34]. The issue of protecting a many-to-one wireless session against any number of failures was addressed in [34].

Kodialam *et al.* presented algorithms for dynamic routing of restorable bandwidth guaranteed paths where no prior knowledge of future connection requests with traffic demands are known [3, 4, 5]. In these works backup sharing is allowed. The non-sharing 1+1 protection is used only for the bandwidth efficiency comparison. In order to maximize as many dynamic restorable connection requests as possible, when backup sharing is not allowed, Kar *et al.* used the minimum-interference idea of nonrestorable routing to develop efficient algorithms for finding disjoint paths for protection [7]. Mathematical programming approaches for efficient SBPP are well addressed in [8, 9, 10]. The failure independent path protection in SBPP with p-cycle is considered in [11].

Backup sharing is not possible in 1+1 protection. Thus, Kar *et al.* suggested an improved path selection strategy to achieve efficiency in [7]. The objective in [7] is the dynamic allocation of restorable connection requests. However, in this research, we consider setting up predefined set of disjoint paths for instantaneous receiver-initiated recovery from any single link failure.

1.4 Requirements for our desired routing scheme

We desire a pre-planned reliable route design scheme that should meet the following requirements:

1. Instantaneous receiver-initiated recovery, where only the destination node is responsible for the recovery operation.

2. Desired scheme should use less backup resources than that of the conventional 1+1 protection.

3. Simple encoding and decoding operations.

4. The route design scheme is applicable even if one or both the source and destination nodes in the network have limited connectivity of two.

In order to fulfill above requirements, each node in the network must be able to perform exclusive-OR (XOR) operation (coding capable network), because XOR operation is the technique for encoding and decoding in both MSCD and TS scenarios. Moreover, all the nodes in the network must be bi-connected, i.e., each node must be connected to two neighbor nodes in order to facilitate 1+1 protection between every possible source destination pair.

1.5 Problem statement

Literature review raises two questions. The first question is, *Should we use coding technique for reliable communication network design?*. The second one is, *How much resource saving can we obtain by coding techniques under the best routing?* The 1+1 protection scheme is the only protection scheme that achieves instantaneous receiver-initiated recovery, and optimum 1+1 protection route design with both scenarios, described in Section 1.2, were not addressed in any previous work. In order to answers these two questions, this thesis addresses the coding based route design problem to find the amount of resource saving achieved by using coding technique with protection techniques.

In [13] only a model, with two sources and a common destination, was presented to show the effectiveness of network coding in resource saving with 1+1 protection. In [13] conventional shortest path algorithm was used for determining routes. Once routes are determined, if any coding opportunity occurs then apply it to save resources. Thus the results presented in [13] is not optimum.

1+1 protection without coding was considered as the conventional protection scheme in [14, 15]. The $1 + N$ protection scheme in [14, 15] is resource efficient, however it suffers from the high node degree requirements of N for both the encoding and decoding nodes.

In [20] it was shown that splitting traffic at the source and sending them through disjoint paths reduces the resource requirement on each of the disjoint paths used. Splitting can be used for resource efficiency with 1+1 protection while maintaining its instantaneous recovery advantage. The more we split, the more reduction in required bandwidth per disjoint path we get. However, as the number of splitting increases, the total length of the employed disjoint paths also increases. There must be a trade-off between the bandwidth reduction achieved per path due to traffic splitting, and the total length of the employed disjoint paths. This issue was not addressed in [20].

1.6 Our contributions

In this thesis we propose two coding aware resource efficient 1+1 protection route design schemes for the two different MSCD and TS scenarios. The objective of both proposed schemes is to ensure instantaneous recovery from failure using minimum backup resources. The two schemes are proposed because each of the MSCD and TS scenario differ in terms of applicability, node connectivity requirements, and how coding is performed. For both the proposed scheme, our contributions consist of two parts. First, we formulate the optimization (route design) problem, where an optimum coding aware set of routes that minimizes the required network resources for 1+1 protection are determined, as an Integer Linear Programming (ILP) formulation. Second, in order to design coding aware 1+1 protection routes for all possible source destination pairs, a heuristic routing algorithm is presented where the ILP model is used iteratively.

In order to evaluate the effectiveness and applicability, the performance of our two proposed route design schemes are evaluated in terms of the total resource saving with respect to the conventional 1+1 protection technique. Numerical results observed that the proposed routing scheme with MSCD scenario achieves almost 15% of resource saving, while the routing scheme with the TS scenario achieves almost 20% of resource saving w.r.t. the conventional 1+1 protection cost (without coding) in our examined networks. The applicability of both the proposed schemes depend on the network characteristics, specially on the node connectivity. In terms of resource saving the scheme with the TS scenario is preferred, but it requires high node connectivity for both the source and the destination. For the scheme with the MSCD scenario, resource saving is smaller than that of the other proposed scheme, but it requires minimum node connectivity of two between the source and the destination.

1.7 Organization of the thesis

The organization of this thesis is summarized in Fig. 1.4.

We formulate the network coding based optimum route design problem, in any scenario with $k = 2$ sources and a common destination ($2SD$) (*MSCD scenario*

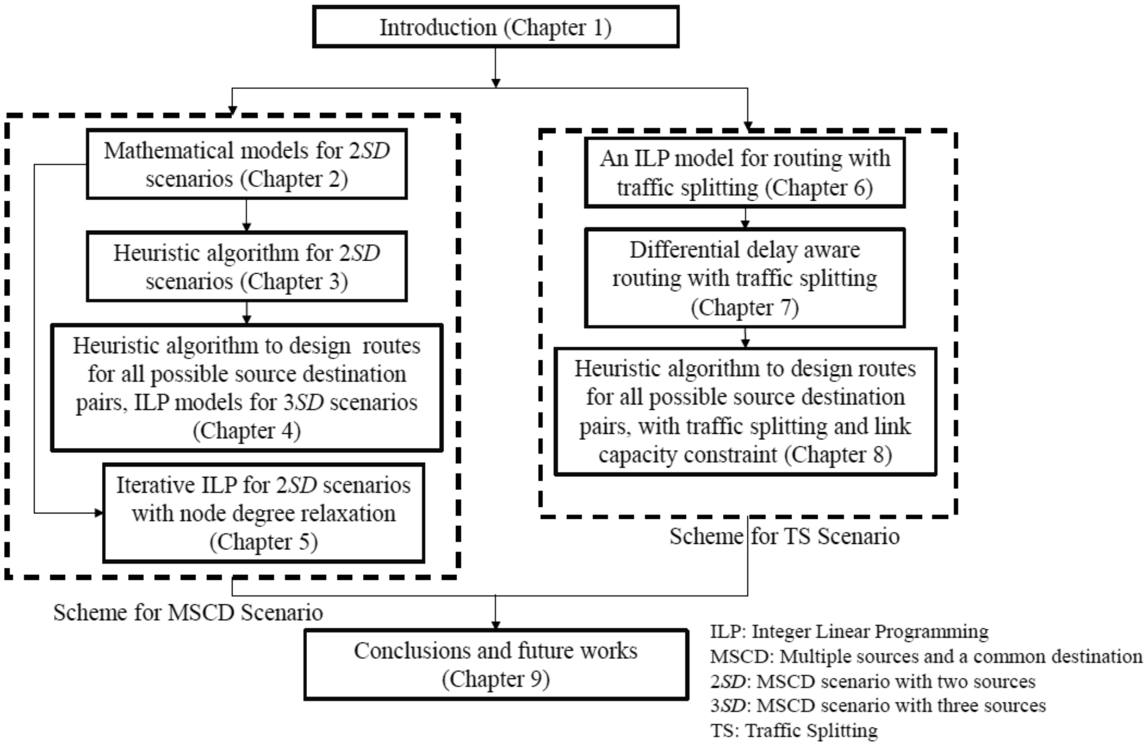


Figure 1.4: Thesis organization.

with two sources), as an integer quadratic programming (IQP) formulation. In order to avoid uncertain computer memory demand and dependency on special algorithm for solution, we re-formulate the optimization problem as an integer linear programming (ILP) formulation. Performance of both the ILP and IQP is evaluated. These issues are covered in chapter 2.

Network coding based optimum route design in $2SD$ scenarios is an OPEN problem, and no efficient algorithm is available to solve this problem. Solving the ILP model demands large amount of random access memory (RAM) and CPU processing power, and the demand increases with the number of nodes in the network. The ILP formulation provides an optimum and quick solution in small-scale and medium-scale networks, but increasing dependency on RAM with increasing number of nodes restricts the applicability of ILP to large-scale networks.

In order to achieve a routing solution in polynomial time, overcome the large memory requirement of ILP, presented in chapter 2, and get the best possible (near optimal) routing solution in large-scale networks, a heuristic algorithm is presented in chapter 3 for the $2SD$ scenarios. The algorithm achieves a high resource saving with the NC effect for the 1+1 protection routing decision. To confirm the accuracy and optimality, the presented algorithm is evaluated in the same small-scale and medium-scale networks as ILP. Evaluation results confirms that in almost 96% cases the presented algorithm achieves the same minimum cost solution as the ILP formulation in the examined networks and the computation time of the presented algorithm is faster than that of ILP. When multiple sources have a common destination, the proper selection of two nodes one by one ensures the best resource saving. Two policies to select two nodes in a kSD scenario are also discussed in chapter 3. This chapter also discusses about the solution in general scenarios with k sources and common destination.

Chapter 4 presents a set of ILP models in order to determine an NC based optimum set of routes in scenarios with three sources and a common destination, $3SD$ (MSCD scenario with three sources). An ILP approach for general scenario is not solvable in reasonable time, because an exponential number of mathematical models need to be solved. In order to design 1+1 protection routes for all possible source destination pairs, a heuristic routing algorithm is presented. In the presented algorithm, a largest effect gain first policy is described in order to select either two or three sources out of k sources at a time, and routing is assigned to the selected $2SD$ and $3SD$ scenario by using corresponding mathematical models. The effects of resource saving, by implementing NC aware protection routes with the selection policy, for all possible source destination pairs in various networks are also discussed in chapter 4.

A conventional model to determine network coding (NC) aware minimum cost routes, for employing 1+1 protection in $2SD$ scenarios, was addressed in chapter 2. When the common destination's node degree is only two, NC cannot be employed, because failure of an adjacent link carrying original data restricts recovery operation at the destination. Chapter 5 presents an iterative mathematical model for optimum NC aware 1+1 protection routing in $2SD$ scenarios with destination's node degree ≥ 2. Numerical results observe that our presented model achieves 5.5% of resource saving, compared to the conventional model, in our examined networks.

In chapter 6, we present an ILP model for instantaneous recovery route design by using the *TS scenario*. The presented ILP model determines a set of $K+1$ disjoint paths which minimizes the total cost of network resource utilization of traffic splitting based protection technique.

Chapter 7 presents a differential delay aware erasure correcting code based instantaneous recovery technique with traffic splitting for networks with the coding capability. In this technique traffic is split into K equal parts and forwarded independently through a set of K disjoint paths. P protection paths are employed in order to tackle any $t \geq 1$ failures on the $(K+P)$ disjoint paths, where $1 \leq t \leq P$. A mathematical model is presented to determine the $K+P$ disjoint paths. However, each disjoint path may experience a different delay, and the destination node has to buffer all the split parts, until the split part experiencing the longest delay reaches the destination. The amount of memory buffers depends on *the maximum allowable differential delay* Δ, the highest difference of delays of any two among the $(K+P)$ disjoint paths, supported at the destination. A large value of Δ increases the amount of required memory buffers and the service providers must consider Δ according to their budgets and service requirements. The effect of Δ on routing with traffic splitting is thoroughly investigated in various networks with respect to encoding/decoding costs, optical interface rates, buffering costs, the number of protection paths, the number of nodes in the network, and the number of minimum adjacent nodes.

Chapter 8 presents a heuristic routing algorithm to design routes for all possible source destination pairs by provisioning erasure correcting code based instantaneous recovery technique with optimal traffic splitting, which was addressed for a source destination pair in chapter 6. We consider a static routing problem in networks having the coding capability. When the links in a network have finite capacities, assigning routing for all possible source destination pair by using this instantaneous recovery technique are mutually dependent. In order to achieve a routing solution within a practical time, the presented heuristic algorithm gives highest priority to the pair either with the largest cost or with the largest resource saving effect. For all source destination pairs the total path costs of implementing erasure correcting code based instantaneous recovery technique, and conventional 1+1 protection technique are computed.

Chapter 9 summarizes the thesis, discusses about the effectiveness, applicability of the proposed schemes, and concludes about the two routing schemes with coding. Moreover, possible future works are also discussed.

Chapter 2

Mathematical models for the MSCD scenario with two sources

This chapter first presents an Integer Quadratic Programming (IQP) model to determine an optimum set of routes that minimizes the required network resources for 1+1 protection using the network coding (NC) technique, in MSCD scenarios with two sources ($2SD$), while maintaining the capability of instantaneous recovery. Solving an IQP problem requires large amount of memory and the Barrier algorithm, which is implemented in limited commercial mathematical programming solvers. Moreover, the amount of memory required for solution cannot be determined exactly for an IQP formulation. In order to avoid uncertain memory demand and dependency on special algorithm, we formulate the optimization problem, corresponding to the IQP model, as an integer linear programming (ILP) formulation. The cost obtained from our developed IQP and ILP models are compared with that of the conventional minimal-cost routing policy, where all three approaches include the NC effect. Numerical results show that the IQP and ILP models achieves almost double (and the same) resource saving effect in our examined networks.

2.1 Network coding (NC) and 1+1 protection

2.1.1 NC based 1+1 versus $1 + N$ protection techniques

Network coding based 1+1 protection technique, which is a *MSCD Scenario* with two sources, is already discussed in section 1.2.1 of chapter 1.

Backup resources are shared in our NC based 1+1 protection (for any scenario with two sources and a common destination $2SD$) and the $1 + N$ protection [14] techniques. In the $1 + N$ protection technique, backup resource sharing is *100%* compared to each working path. In our technique, backup resource sharing is *partial* or up to 100%. When backup sharing is exactly 100%, our considered technique is also categorized into the $1 + N$ protection technique, where $N = 2$.

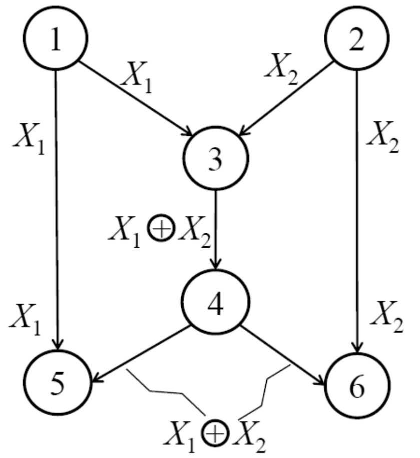
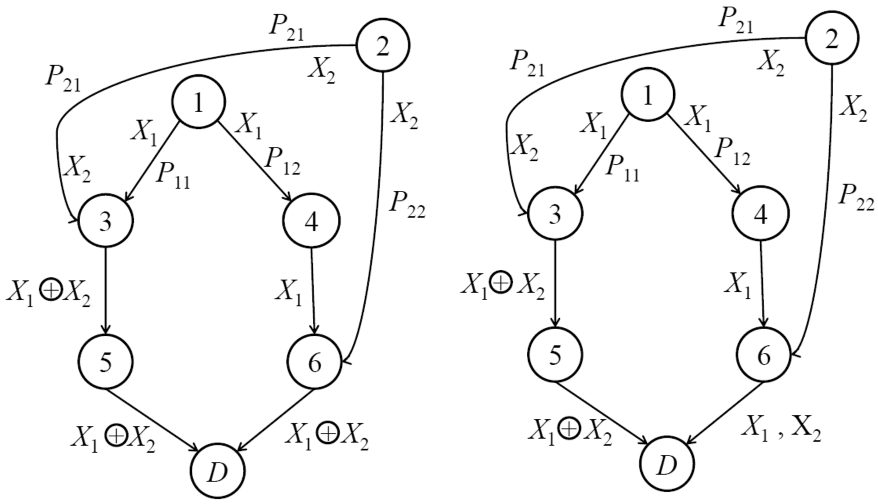


Figure 2.1: Different destination nodes suppress NC advantages.

2.1.2 Problems that may arise when NC is applied

In the butterfly network [12] of Fig. 2.1, source nodes 1 and 2 send their data (X_1 and X_2 respectively) to different destination nodes 5 and 6, respectively. Suppose link $1 - 5$ fails. Destination node 5 cannot recover plain message X_1 instantly from encoded message $X_1 \oplus X_2$. One way to recover is to request node 6 for X_2, and after receiving it node 5 recovers X_1. This recovery procedure is not permitted in our work, because in our assumption destination nodes are not allowed to share received data. From the examples of Fig. 1.2 and Fig. 2.1 it is clear that *common destination node* and *node degree* $\geq$ *3* are two necessary conditions for applying NC to 1+1 protection.



(a) NC is applied at nodes 3 and 6. (b) NC is applied at node 3 only.

Figure 2.2: No NC opportunity is usable.

There are some scenarios where opportunities for network coding exist, but we cannot use all the available opportunities, because it may violate the 1+1 protection in the network. For example, in Fig. 2.2(a) nodes 1 and 2 are sources, and node D is the common destination. Both nodes 3 and 6 have opportunities of applying NC. Let NC is applied at both nodes. Destination node D receives the encoded data ($X_1 \oplus X_2$) only. Let us consider Fig. 2.2(b), where NC is employed at only one node, say node 3. When link $6 - D$ fails, node D has no plain data, either. In these cases NC based 1+1 protection is not possible.

2.2 Necessity of an optimization problem

Three possible 1+1 protection routing solutions (one without NC, and two with NC) for any $2SD$ scenario are shown in Fig. 2.3. Without solving an optimization problem we cannot determine which solution is the desired solution that minimizes the total path cost (sum of costs of the used links).

The optimization problem is described in the following:

- Given conditions and parameters: network topology, link costs, two source, and a common destination.

- Objective: Minimize the total path cost of 1+1 protection among the two sources and common destination using the network coding (NC) technique, where total path cost is the sum of costs of the used links

- Decision variables: Binary variables having a value either 0 or 1. Routing variables - determine which links should be used, and NC decision variable - determine the intermediate node (to apply NC) and common links for achieving NC effect.

- Constraints: Path disjoint constraints, and flow conservation constraints.

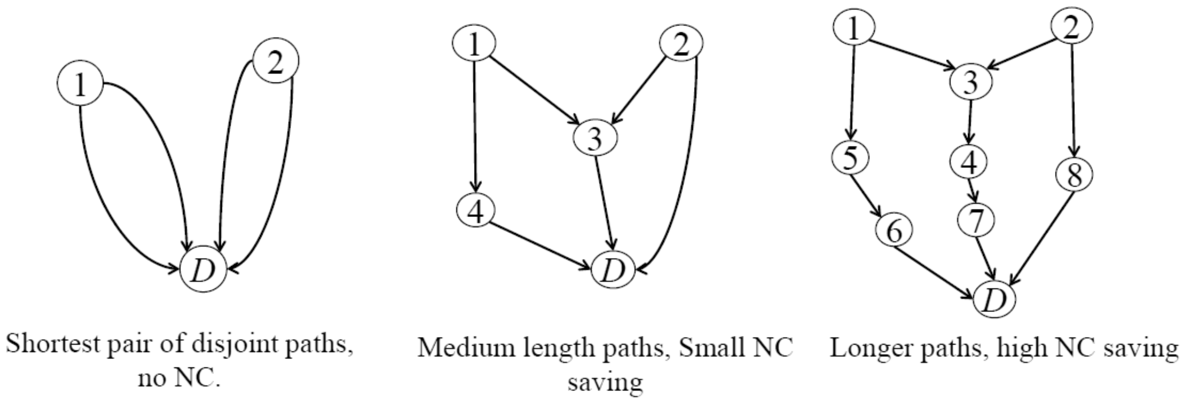


Figure 2.3: Three possible 1+1 protection routing solutions in any $2SD$ scenario.

2.3 Mathematical model

2.3.1 Terminologies

The network is represented as an undirected graph[1] $G(V, E)$, where V is the set of vertices (nodes) and E is the set of links. There are $\mathcal{N}$ nodes and $|E|$ links in $G(V, E)$. A link from node $i \in V$ to node $j \in V$ is denoted as $(i, j) \in E, i \neq j$. $x_{ij}^{sd,k}$ is the binary routing variable. If any link $(i, j) \in E$ belongs to the disjoint path number k between source node s and destination node d, then value of $x_{ij}^{sd,k}$ is 1, otherwise 0, where k is either 1 or 2. c_{ij} is the cost of $(i, j) \in E$, which is a given parameter. The link capacity constraints are not considered, and the traffic demands for all possible source destination pairs are considered the same in the presented mathematical model[2]. The network is bi-connected, i.e., it is ensured that in the network, for every possible source-destination pair, there exist two disjoint paths for applying 1+1 protection. Every node with degree at least three has the NC capability, but encoded data are only decoded at the destination node. P_{sk} indicates the kth path between source node s and common destination node d, where k is either 1 or 2. The two source nodes are represented as 1 and 2 in this chapter.

2.3.2 Network coding (NC) gain

Network coding gain, denoted by G_{NC}, implies that how much network resources is saved in 1+1 protection by using the NC technique. Let us assume that C_{NC} indicates the total path cost of employing 1+1 protection (in an MSCD scenario with $k \geq 2$ sources, or in a TS scenario) considering the resource saving due to NC, and C_{NO_NC} indicates the total path cost without considering the NC effect. G_{NC} is defined by,

$$G_{NC} = \max \left\{ \frac{C_{NO\_NC} - C_{NC}}{C_{NO\_NC}}, 0 \right\}. \tag{2.1}$$

[1]In communication network a link is usually bi-directional, and link costs in both directions are the same. This is why the given network is represented as an undirected graph [44].

[2]A mathematical model with unequal traffic demands and link capacity constraints is presented in Appendix A.

Equation (2.1) states that, if $C_{NC} < C_{NO\_NC}$ we achieve some positive gain, and routing solution considering the NC effect is adopted. Otherwise, $C_{NC} \geq C_{NO\_NC}$ there is no (or zero) gain, and the routing solution without considering the NC effect is adapted.

2.3.3 Integer quadratic programming (IQP) model formulation

The mathematical model for any MSCD scenario with two sources, $2SD$, is presented in this section. The objective function and the path disjoint constraints are also discussed here. In any $2SD$ scenario five path disjoint constraints are needed to employ NC with 1+1 protection. Let us consider the scenario of Fig. 1.2(b), where the first path between source node 1 and common destination node D is denoted as P_{11} (working path) and the second path as P_{12} (backup path). Similarly, for the second source node 2 the two paths, working and backup, are denoted as P_{21} and P_{22}, respectively. The path disjoint constraints are:

1. P_{11} and P_{12} must be disjoint,
2. P_{21} and P_{22} must be disjoint,
3. P_{11} and P_{21} must be disjoint,
4. P_{11} and P_{22} must be disjoint, and
5. P_{12} and P_{22} must be disjoint.

Constraints 1 and 2 are the compulsory conditions that ensure 1+1 protection. Constraints 3, 4, and 5 are set carefully to achieve proper NC effect from the objective function. These five constraints together ensure that the two paths indexed by P_{12} and P_{22} must include the effect of NC. The constraint that P_{11} and P_{12} must be disjoint is expressed in the mathematical model by,

$$x_{ij}^{s_1d,1} + x_{ij}^{s_1d,2} \leq 1, \forall (i,j) \in E,$$

where $x_{ij}^{s_1d,1}$ and $x_{ij}^{s_1d,2}$ are the binary routing variables.

The IQP mathematical model for any $2SD$ scenario is formulated as follows:

$$\textbf{min} \quad \sum_{(i,j)\in E}\sum_{k=1}^{2} c_{ij} \times (x_{ij}^{s_1d,k} + x_{ij}^{s_2d,k}) - \sum_{(i,j)\in E} (c_{ij} \times x_{ij}^{s_1d,2} \times x_{ij}^{s_2d,2}) \tag{2.2a}$$

$$\textbf{s.t.} \quad x_{ij}^{s_1d,1} + x_{ij}^{s_1d,2} \leq 1, \quad \forall (i,j) \in E \tag{2.2b}$$

$$x_{ij}^{s_2d,1} + x_{ij}^{s_2d,2} \leq 1, \quad \forall (i,j) \in E \tag{2.2c}$$

$$x_{ij}^{s_1d,1} + x_{ij}^{s_2d,1} \leq 1, \quad \forall (i,j) \in E \tag{2.2d}$$

$$x_{ij}^{s_1d,1} + x_{ij}^{s_2d,2} \leq 1, \quad \forall (i,j) \in E \tag{2.2e}$$

$$x_{ij}^{s_1d,2} + x_{ij}^{s_2d,1} \leq 1, \quad \forall (i,j) \in E \tag{2.2f}$$

$$\sum_{j\in V}(x_{ij}^{s_1d,1} - x_{ji}^{s_1d,1}) = 1, \quad \forall i \in V, i = s_1 \tag{2.2g}$$

$$\sum_{j\in V}(x_{ij}^{s_1d,2} - x_{ji}^{s_1d,2}) = 1, \quad \forall i \in V, i = s_1 \tag{2.2h}$$

$$\sum_{j\in V}(x_{ij}^{s_2d,1} - x_{ji}^{s_2d,1}) = 1, \quad \forall i \in V, i = s_2 \tag{2.2i}$$

$$\sum_{j\in V}(x_{ij}^{s_2d,2} - x_{ji}^{s_2d,2}) = 1, \quad \forall i \in V, i = s_2 \tag{2.2j}$$

$$\sum_{j\in V}(x_{ij}^{s_1d,1} - x_{ji}^{s_1d,1}) = 0, \quad \forall i \in V, i \neq s_1, i \neq d \tag{2.2k}$$

$$\sum_{j\in V}(x_{ij}^{s_1d,2} - x_{ji}^{s_1d,2}) = 0, \quad \forall i \in V, i \neq s_1, i \neq d \tag{2.2l}$$

$$\sum_{j\in V}(x_{ij}^{s_2d,1} - x_{ji}^{s_2d,1}) = 0, \quad \forall i \in V, i \neq s_2, i \neq d \tag{2.2m}$$

$$\sum_{j\in V}(x_{ij}^{s_2d,2} - x_{ji}^{s_2d,2}) = 0, \quad \forall i \in V, i \neq s_2, i \neq d \tag{2.2n}$$

$$x_{ij}^{sd,k} = \{0,1\}, \forall (i,j) \in E, i \neq j, s = s_1, s_2, k = 1,2. \tag{2.2o}$$

Equation (2.2a) is the objective function which provides the optimum minimum cost of employing NC based 1+1 protection. The second term in Eq. (2.2a) is a quadratic term. It is responsible for providing the NC effect. Equations (2.2b)-(2.2f) specify the path disjoint constraints. Constraints (2.2b)-(2.2c) are the compulsory conditions that ensure 1+1 protection. Constraints (2.2d)-(2.2f) are set carefully to achieve proper network coding effect from the two paths indexed by P_{12} and P_{22}. The flow conservation constraints are specified by Eqs. (2.2g)-(2.2n). The flow conservation constraints at the source nodes are expressed by Eqs. (2.2g)-(2.2j), while Eqs. (2.2k)-(2.2n) are the flow conservation constraints at the intermediate nodes. The binary routing variable is described by Eq. (2.2o).

2.3.4 Integer linear programming (ILP) model formulation

The ILP model is presented as follows.

$$\textbf{min} \quad \sum_{(i,j)\in E}\sum_{k=1}^{2} c_{ij} \times (x_{ij}^{s_1 d,k} + x_{ij}^{s_2 d,k}) - \sum_{(i,j)\in E} (c_{ij} \times z_{ij}) \tag{2.3a}$$

$$\textbf{s.t.} \quad x_{ij}^{s_1 d,1} + x_{ij}^{s_1 d,2} \leq 1, \forall (i,j) \in E \tag{2.3b}$$

$$x_{ij}^{s_2 d,1} + x_{ij}^{s_2 d,2} \leq 1, \forall (i,j) \in E \tag{2.3c}$$

$$x_{ij}^{s_1 d,1} + x_{ij}^{s_2 d,1} \leq 1, \forall (i,j) \in E \tag{2.3d}$$

$$x_{ij}^{s_1 d,1} + x_{ij}^{s_2 d,2} \leq 1, \forall (i,j) \in E \tag{2.3e}$$

$$x_{ij}^{s_1 d,2} + x_{ij}^{s_2 d,1} \leq 1, \forall (i,j) \in E \tag{2.3f}$$

$$\sum_{j\in V}(x_{ij}^{s_1 d,1} - x_{ji}^{s_1 d,1}) = 1, \ i = s_1 \tag{2.3g}$$

$$\sum_{j\in V}(x_{ij}^{s_1 d,2} - x_{ji}^{s_1 d,2}) = 1, \ i = s_1 \tag{2.3h}$$

$$\sum_{j\in V}(x_{ij}^{s_2 d,1} - x_{ji}^{s_2 d,1}) = 1, \ i = s_2 \tag{2.3i}$$

$$\sum_{j\in V}(x_{ij}^{s_2 d,2} - x_{ji}^{s_2 d,2}) = 1, \ i = s_2 \tag{2.3j}$$

$$\sum_{j\in V}(x_{ij}^{s_1 d,1} - x_{ji}^{s_1 d,1}) = 0, \ \forall i \in V, i \neq s_1, d \tag{2.3k}$$

$$\sum_{j\in V}(x_{ij}^{s_1 d,2} - x_{ji}^{s_1 d,2}) = 0, \ \forall i \in V, i \neq s_1, d \tag{2.3l}$$

$$\sum_{j\in V}(x_{ij}^{s_2 d,1} - x_{ji}^{s_2 d,1}) = 0, \ \forall i \in V, i \neq s_2, d \tag{2.3m}$$

$$\sum_{j \in V}(x_{ij}^{s_2d,2} - x_{ji}^{s_2d,2}) = 0,\ \forall i \in V, i \neq s_2, d \tag{2.3n}$$

$$x_{ij}^{sd,k} = \{0,1\},\ \forall (i,j) \in E, \quad s = s_1, s_2, k = 1,2 \tag{2.3o}$$

$$z_{ij} = \{0,1\},\ \forall (i,j) \in E \tag{2.3p}$$

$$z_{ij} \leq x_{ij}^{s_1d,2},\ \forall (i,j) \in E \tag{2.3q}$$

$$z_{ij} \leq x_{ij}^{s_2d,2},\ \forall (i,j) \in E \tag{2.3r}$$

$$z_{ij} \geq x_{ij}^{s_1d,2} + x_{ij}^{s_2d,2} - 1,\ \forall (i,j) \in E. \tag{2.3s}$$

Eq. (2.3a) is the objective function of the ILP model which provides an optimum minimum cost of employing NC based 1+1 protection among two sources and a common destination. Eqs. (2.3p)-(2.3s) describe the binary decision variable z_{ij} which determine the links that are subject to NC. All the others constraints are the same in both models.

2.3.4.1 Explanation for excluding flow conservation constraints at the destination in the presented ILP

At the destination node the following four conditions are necessary to maintain flow.

$$\sum_{j \in V}(x_{ij}^{s_1d,1} - x_{ji}^{s_1d,1}) = -1,\ i = d \tag{2.4a}$$

$$\sum_{j \in V}(x_{ij}^{s_1d,2} - x_{ji}^{s_1d,2}) = -1,\ i = d \tag{2.4b}$$

$$\sum_{j \in V}(x_{ij}^{s_2d,1} - x_{ji}^{s_2d,1}) = -1,\ i = d \tag{2.4c}$$

$$\sum_{j \in V}(x_{ij}^{s_2d,2} - x_{ji}^{s_2d,2}) = -1,\ i = d \tag{2.4d}$$

The constraints of Eqs. (2.4a)-(2.4d) must be satisfied to maintain flow at the destination node. However, these four flow conservation constraints at the destination are deducted by Eqs. (2.2g)-(2.2n). This is because Eqs. (2.4a)-(2.4d) are guaranteed by Eqs. (2.2g)-(2.2n). In general, flow conservation constraints at the destination are derived from the flow conservation constraints at the source and intermediate nodes, the proof of which is given in [41].

2.3.4.2 Technique used to convert the presented IQP model into an ILP model

The second term in Eq. (2.2a), which is the objective function of the presented IQP model, is a quadratic term, because it includes multiplication of two binary routing variables. The multiplication of two binary variables can be expressed in linear form by introducing a third binary variable as explained in the following.

Let x_1 and x_2 be two binary routing variables, and z be a third binary variable. The equivalent of multiplication of two binary variables $x_1 \times x_2$ can be expressed in linear form by the following Eqs. (2.5a)-(2.5f),

$$x_1 = \{0, 1\} \tag{2.5a}$$

$$x_2 = \{0, 1\} \tag{2.5b}$$

$$z = \{0, 1\} \tag{2.5c}$$

$$z \leq x_1 \tag{2.5d}$$

$$z \leq x_2 \tag{2.5e}$$

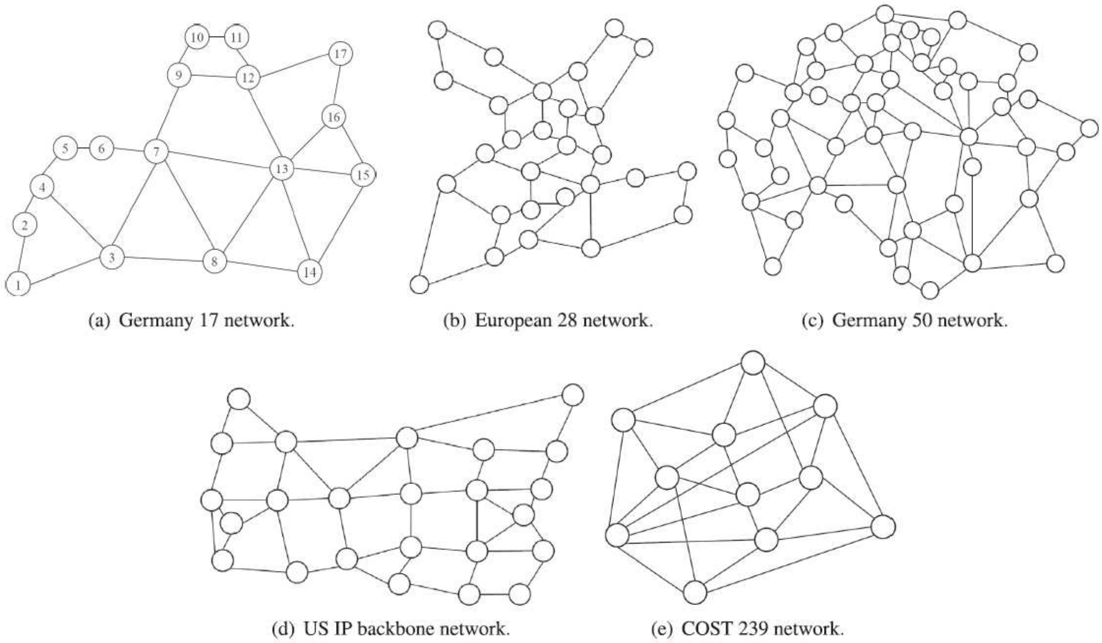
$$z \geq x_1 + x_2 - 1 \tag{2.5f}$$

2.4 Results and discussions

The performances of the presented IQP and ILP models are evaluated in terms of network coding gain (G_{NC}) and required computer memory for solution. The G_{NC} achieved by our presented models are compared with that of the conventional minimal-cost routing policy with NC.

2.4.1 G_{NC} performance of our ILP model

The performance of the presented mathematical model is evaluated and compared in the five network topologies of Fig. 2.4. The mathematical model is solved by using CPLEX® [46]. The costs with and without NC are evaluated to compute the G_{NC}. The important characteristics of the five examined network topologies are summarized in Table 2.1. In the evaluation all nodes in the network are assumed to behave as edge nodes. In the network topologies, link utilization cost is unity.



(a) Germany 17 network. (b) European 28 network. (c) Germany 50 network.

(d) US IP backbone network. (e) COST 239 network.

Figure 2.4: Examined network topologies.

We arbitrarily pick 100 $2SD$ scenarios in each of five the evaluated topologies of Fig. 2.4, and the G_{NC} results presented in Fig. 2.5 are the average of the 100 observations in each of the cases.

In our network model it is assumed that each node in the network must have two links connected to it. This is a necessary condition for employing 1+1 protection. If a node has degree only one, no protection technique is possible for that node.

Table 2.1: Characteristics of the five networks

	Germany 17	Europe 28	Germany 50	US IP Backbone	COST239
#of nodes ($\mathcal{N}$)	17	28	50	24	11
Avg. node deg. (D_{avg})	3.06	2.93	3.56	3.54	4.73
Nodes with deg. > 2	59%	68%	74%	87.5%	100%

The G_{NC} of the presented ILP formulation is compared with that of the

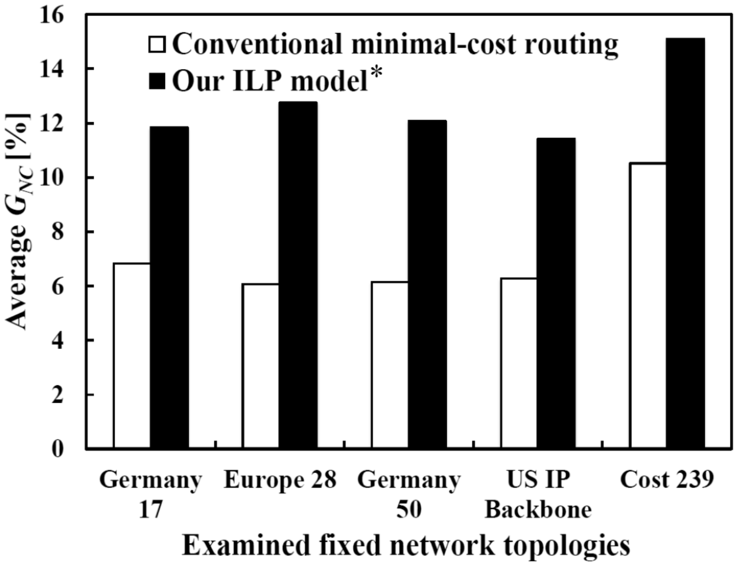


Figure 2.5: Comparison of network coding gains. * Both ILP and IQP in [36] provides same cost performances.

conventional minimal-cost routing policy. The conventional routing policy is explained in the following. More than two disjoint paths between a source and destination pair may exist. Among those disjoint paths, the conventional minimal-cost routing policy selects only two paths whose sum of costs is minimum, where the edge-disjoint shortest pair algorithm [44] is used. In this routing policy, routes are determined at first, and then network coding (NC) is applied only if an NC opportunity is created and a backup path from one source does not overlap with the working path from the other source.

The theoretical maximum achievable G_{NC} in fully connected mesh network is $\frac{Y-1}{3Y}$, where Y is the number of source nodes having a common destination node [13]. If $Y >> 1$ the maximum G_{NC} is about 33%. For $Y = 2$, i.e., for $2SD$ scenario, the theoretical maximum coding gain yields 16.67%. In our evaluation we obtain the maximum average G_{NC} of 15.1% in COST 239 network, and the average G_{NC} for the other four networks are around 12% as illustrated in Fig. 2.5. Our presented ILP provides a high resource saving and is comparable to the theoretically maximum achievable gain. Figure 2.5 illustrates that the number of nodes in the network is not the main contributing factor for achieving a higher G_{NC}, rather the connection patterns and the source and destination nodes' positions matter.

2.4.2 ILP versus IQP

The ILP and IQP [36] models provide the same optimum cost solution, as illustrated for our ILP model in Fig. 2.5. However, ILP requires less amount of memory and time for its solution. For an IQP the lower bound of required memory for solution can be determined from the number of constraints and number of variables in the objective function, while for an ILP the memory is determined exactly from the number of constraints in the model [47]. Our presented ILP model reduces the memory requirement to about 11.6% than that of the IQP model in our five examined networks.

2.5 Summary

This chapter has formulated the problem of finding optimum network coding aware set of routes as an IQP and then as an ILP problem, where each possible set of routes is evaluated, to determine the optimum solution that minimizes the cost of provisioning 1+1 protection. G_{NC} of the conventional minimal-cost routing policy is lower than that of our presented models, because the conventional routing policy creates less frequent NC opportunities. The performance of our IQP model is thoroughly evaluated and analyzed in five different network topologies. Numerical results have observed that our mathematical models achieves almost double resource saving effect than that of the conventional minimal-cost routing policy in the examined networks.

Chapter 3

Heuristic routing algorithm for MSCD scenario with two sources

Optimum network coding (NC) based 1+1 protection route design that minimizes the total cost of protection, in an MSCD scenario with k sources (kSD) where k is a variable, is an NP-Hard problem. However, the complexity of this routing problem, when k is fixed, is still an OPEN issue[1]. No efficient algorithm is available to solve the NC based routing problem in MSCD scenarios with two sources ($2SD$). Therefore, a heuristic approach is required. In chapter 2, an integer linear programming (ILP) formulation was presented to solve the optimization problem for an MSCD scenario with two sources ($2SD$). The presented ILP model in chapter 2, works well in small-scale and medium-scale networks, but fails to support large-scale networks due to excessive memory requirements and calculation time. In order to deal with these issues and achieve a feasible routing solution in polynomial time a heuristic algorithm is presented in this chapter to determine the best possible NC aware set of routes, for MSCD scenarios with two sources ($2SD$), in large-scale networks. Numerical results show that our presented algorithm achieves almost double resource saving effect than the conventional minimal-cost routing policy in the examined medium-scale and large scale networks. In MSCD scenarios with $k \geq 2$ sources kSD, 1+1 protection routing with NC can be implemented by selecting 2 sources at a time that

[1]NP-Hard proof, and OPEN complexity issues are discussed in Appendix B of this thesis

ensures best possible resource saving. We also present two policies to select two sources in kSD scenarios. The general solution approach in the kSD scenarios and its complexity are also discussed.

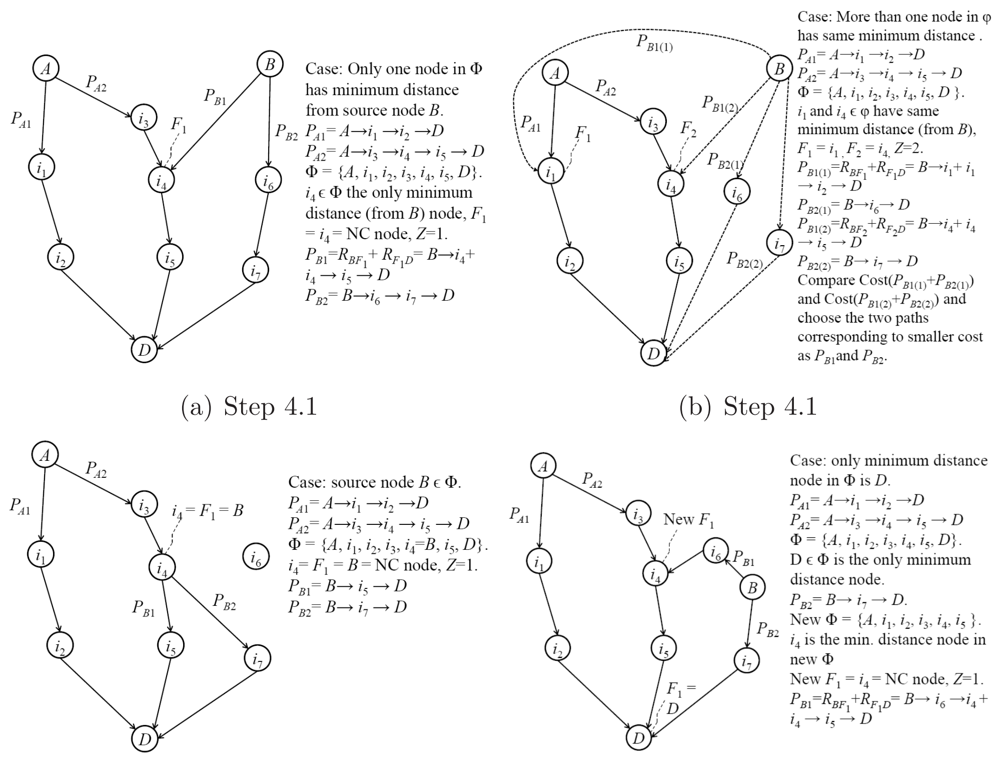
3.1 Presented heuristic algorithm

The presented heuristic algorithm, its demonstration, and two policies to select two sources out of k sources in any $kSD, k \geq 2$ scenario are discussed in this section.

3.1.1 Terminologies

Let R_{lm} denote a partial path between any two nodes L and M, where one node is an intermediate node and the other node is a source or destination node. Let $\zeta(path)$ denote the cost of a path, where $path$ can be the disjoint one (P_{sk}) or the partial one (R_{lm}). $\zeta(path)$ is proportional to the distance between the two end nodes. In the algorithm description the two source nodes are denoted as A and B, and destination node as D. The four disjoint paths are denoted as P_{A1}, P_{A2}, P_{B1}, and P_{B2}, where P_{A1} means the first disjoint path between source node A and destination node D. Among these four paths P_{A2} and P_{B1} are the desired candidates to enjoy the NC effect. Let Φ be a set of the nodes, where the nodes belong to either P_{A1} and P_{A2}, or to P_{B1} and P_{B2}, including source and destination nodes. Let $F_i \in \Phi$ be the node that has the minimum distance from the second source node (if A is the first source node, then B is the second source node and vice versa). More than one node $\in \Phi$ may have the same minimum distance which implies that $i \geq 1$. Let $\Psi \subset E$ be a set of links.

Let us define $P_{B1(i)}$ to be the sum of two partial paths R_{BF_i} and R_{F_iD}, i.e., $P_{B1(i)} = R_{BF_i} + R_{F_iD}$, where $i \geq 1$. The partial path R_{BF_i} is determined by an exhaustive search among possible candidates to achieve the NC effect, where F_i is the node that performs NC. The partial path R_{F_iD} is overlapped with either P_{A1} or P_{A2}. The notations, path descriptions, and sets discussed so far are illustrated in Fig. 3.1 for easy understanding of the presented algorithm.



(a) Step 4.1 (b) Step 4.1

(c) Step 4.5 (d) Step 4.7

Figure 3.1: Illustration of four different cases that may arise when selecting the NC node in stage 1 of the presented algorithm.

3.1.2 Algorithm description

The algorithm has two stages. In the first stage, P_{A1}, P_{A2} are determined by the MIN-SUM disjoint paths algorithm [43, 44], and P_{B1} and P_{B2} are determined by an exhaustive search. In the second stage, P_{B1} and P_{B2} are determined first, and then P_{A1} and P_{A2} are determined by an exhaustive search. In both stages, the cost of the four paths, mentioned above, are computed. The four paths, either from stage 1 or from stage 2, with the smaller total cost are the desired solution.

STAGE 1

- Step 1: Evaluate and find the two disjoint paths (P_{A1} and P_{A2}) between source node A and destination node D.

- Step 2: Construct the set Φ that includes the nodes belonging to the paths P_{A1} and P_{A2} including nodes A and D.

- Step 3: From source node B, compute the distances of all the nodes in Φ.

- Step 4: This step determines the routes for partial path R_{BF_i}, and the path P_{B2}. Pick one or more nodes $F_i \in \Phi$, $i = 1, 2, \cdots, Z$, where Z is the number of nodes that have the same minimum distance from source node B. If the distance between source node B and F_i is zero, set $i = 1$ and go to Step 4.5. If F_i is the destination node D, set $i = 1$ and go to Step 4.7. Otherwise set $i = 0$ and go to step 4.1.

  - Step 4.1: Set $i = i + 1$. If $i > Z$ go to Step 4.4. Let the partial path from source node B to node F_i be R_{BF_i}. Partial path R_{F_iD} is overlapped with either P_{A1} or P_{A2}. Addition of R_{BF_i} and R_{F_iD} yields $P_{B1(i)}$.

  - Step 4.2: Construct the set Ψ that includes the links that belong to paths P_{A1}, P_{A2} and $P_{B1(i)}$. Delete all the links in set Ψ from the graph G. Find the shortest path $P_{B2(i)}$ in the modified graph $G - \Psi$. This path is disjoint from the other three paths P_{A1}, P_{A2} and $P_{B1(i)}$.

  - Step 4.3: If $i \leq Z$ go to Step 4.1.

  - Step 4.4: Check the sums of $\zeta(P_{B1(i)})+\zeta(P_{B2(i)})$, $i = 1, 2, \cdots, Z$. From these Z costs, choose the combination with minimum cost. Corresponding $P_{B1(i)}$ and $P_{B2(i)}$ are chosen as path P_{B1} and P_{B2} and corresponding F_i is chosen as the network coding node. Go to Step 5.

  - Step 4.5: Distance = 0 means $\zeta(R_{BF_i})$=0. In other words, F_i itself is the source node B and the path P_{B1} overlaps with either P_{A1} or P_{A2} and node B applies network coding. Determine P_{B2} following the procedure in Step 4.2.

  - Step 4.6: Distance=0 means Z=1. Go to Step 5.

  - Step 4.7: If F_i is destination node D, we denote the path from source node B to the chosen node as $P_{B2(i)}$. $P_{B2(i)}$ is disjoint from the two paths P_{A1} and P_{A2}. Exclude destination node D from the set Φ and set $i = i + 1$. Again pick a new node (having minimum distance from source node B) from the modified Φ to determine R_{BF_i} and this new node is the candidate for applying NC.

- Step 4.8: If i $\leq$ Z, go to Step 4.1.

- Step 5: Compute the total cost of these four paths and denote it as *Stage One Cost* = $\zeta(P_{A1})+\zeta(P_{A2})+\zeta(R_{BF_i})$ $+\zeta(P_{B2})$. The *Stage One Cost* includes the cost of partial path $\zeta(R_{BF_i})$. The partial path R_{F_iD} is common between either paths P_{A1} and P_{B1}, or paths P_{A2} and P_{B1}. When NC is applied at node F_i, two data streams from nodes A and B become one data stream, and the shared links along the partial path R_{F_iD} is used only once. $\zeta(R_{F_iD})$ amount of cost is saved due to the use of NC at node F_i. The cost of the partial path $\zeta(R_{F_iD})$ is already included in either $\zeta(P_{A1})$ or $\zeta(P_{A2})$. That is why, the cost of the partial path $\zeta(R_{BF_i})$ is included in the *Stage One Cost*.

STAGE 2

- Step 6: Repeat Step 1 to Step 5, where source node B is considered first and source node A is considered next.
  - Step 6.1: Find MIN-SUM disjoint paths P_{B1} and P_{B2} and construct Φ that includes the nodes along the paths P_{B1} and P_{B2} including source node B and destination node D.
  - Step 6.2: From source node A, find the distance of all the nodes in set Φ. Following Steps 4.1-4.8 of stage 1, determine two partial paths R_{AF_i}, R_{F_iD} and disjoint path P_{A1}. Addition of the two partial paths results in path P_{A2}.
  - Step 6.3: Compute the cost of these four paths and denote it as *Stage Two Cost* = $\zeta(P_{A1})+\zeta(R_{AF_i})+$ $\zeta(P_{B1})+\zeta(P_{B2})$. **Stage 2 ends**.

- Step 7: Compare the two costs *Stage One Cost* and *Stage Two Cost* and select the four paths corresponding to the minimum cost as the desired solution for 1+1 protection with NC.

- Step 8: End of the algorithm.

3.1.2.1 How to use presented heuristic algorithms solution

In the presented heuristic algorithm, in order to select an NC opportunity, detour may be required, which in turn increases the total path cost. Thus the routing solution achieved by our heuristic algorithm can be worse (a higher total path cost) than that achieved by conventional minimal-cost routing. Let $G_{NC-Heuristic}$ denote the network coding gain achieved by our heuristic algorithm, and $G_{NC-conventional}$ denote the network coding gain achieved by the conventional minimal cost routing. $G_{NC-Heuristic} < G_{NC-conventional}$ implies that the total path cost achieved by the heuristic algorithm is smaller than that achieved by conventional minimal-cost routing, and in this case routing with our heuristic algorithm is adopted. Otherwise, i.e., if $G_{NC-Heuristic} \geq G_{NC-conventional}$, the solution with the conventional minimal cost routing results is adopted. Due to this policy the worst case solution achieved by our heuristic algorithm does not matter.

3.1.3 Numerical demonstration of heuristic algorithm

The heuristic algorithm for any $2SD$ scenario is illustrated here with the help of an example. Let us consider Germany 17 network, as shown in Fig. 2.4(a), where the cost of each link is 1. Considered source and destination nodes are $A = 7$, $B = 14$, and $D = 16$. The distance between any two nodes l and m is denoted by $\Pi(lm)$. The two stages are as follows:

Stage 1: In steps 1-3, we obtain P_{A1} = 7-8-14-15-16, P_{A2} = 7-13-16, Φ = {7,8,13,14,15,16}, and all the distances from source node $B = 14$ to the nodes in Φ as $\Pi(14,7)$=2, $\Pi(14,8) = 1$, $\Pi(14,13)$=1, $\Pi(14,14)$=0, $\Pi(14,15)$=1, and $\Pi(14,16)$=2. In Step 4 it is found that the second source node $B = 14 \in \Phi$. Go to Step 4.5 to obtain P_{B1} = 14-15-16. P_{B1} overlaps with P_{A1} and node 14 is the network coding node. Due to NC effective cost of path $P_{B1} = 0$. P_{B2} is determined following the procedure in Step 4.2. Construct Ψ = {(7,8), (7,13), (8,14), (13,16), (14,15), (15,16)}. Delete these six links from graph G and find the shortest path P_{B2} = 14-13-12-17-16. Step 5 computes the total cost of the four paths as *Stage One Cost* = 10.

Stage 2: Though stage 2 includes step 6, it actually repeats steps of stage 1. Following Steps 1-3, we obtain P_{B1} = 14-13-16, P_{B2} = 14-15-16, Φ = {13,14,15,16} and the distances from $A = 7$ to all the nodes in Φ as $\Pi(7,13)$=1, $\Pi(7,14)$=2, $\Pi(7,15)$=2, and $\Pi(7,16)$=2. Step 4 indicates that node 13 is the only shortest distance node in Φ, i.e., $F_i = 13$, where $i = 1 = Z$. Following Steps 4.1-4.2 we obtain P_{A1} and P_{A2} as follows: R_{AF_i} = 7-13 and R_{F_iD} = 13-16 addition of which yields P_{A2} =7-13-16. The effective cost for path P_{A2} due to network coding = 1 = cost of path R_{AF_i}. The set $\Psi = \{$(7,13), (14,13), (14,15), (13,16), (15,16)$\}$. Delete these five links from graph G and find the shortest path P_{A1} = 7-9-12-17-16. Step 6.3 computes the total cost of the four paths as *Stage Two Cost* = 9.

In Step 7 we find that the path set from stage 2 provides the smaller cost. The cost of the four paths without NC is 10. According to the definition of G_{NC}, the presented algorithm provides $\frac{10-9}{10} \times 100 = 10\%$ NC gain.

3.1.4 Computational complexity

The presented algorithm has time complexity $O(\mathcal{N}^2(|\Phi|) + |E|)$, where $\mathcal{N}$ and $|E|$ are the number of nodes and the number of links in the network, respectively, and $|\Phi|$ is the number of nodes in Φ. In the presented algorithm's implementation Dijkstra's shortest path algorithm is used. Dijkstra's algorithm has the worst case complexity of $O(\mathcal{N}^2)$ [44]. In the first stage of the presented algorithm, in order to determine P_{A1}, P_{A2}, and P_{B2}, the shortest path algorithm is used three times. In order to find the best *NC node*, the shortest path algorithm has to be run $|\Phi|$ times. Before determining P_{B2} in stage 1, and P_{A1} in stage 2, the edges belonging to the other three disjoint paths (P_{A1}, P_{A2}, and P_{B1} for stage 1, P_{A2}, P_{B1}, and P_{B2}) for stage 2, respectively,) must be deleted from the original graph resulting in two temporary graphs. These temporary graphs required all the edges to in G to be considered twice. There are two stages in the presented algorithm. The required timing computational complexity is therefore $O(2(3\mathcal{N}^2 + \mathcal{N}^2|\Phi| + 2|E|))$. Since, the $O()$ notation disregards any constants [65], the final timing complexity notation yields $O(\mathcal{N}^2(|\Phi|) + |E|)$.

3.1.5 Policies to select two sources

Let φ be the set of source nodes having a common destination, where $s_i \in \varphi, i = 1, 2, \cdots, Y$, where $Y \geq 2$. A combination of two sources, s_i and s_j, out of Y

sources, is expressed by $(s_i, s_j) \in \Theta, (i < j)$, where Θ is a set of (s_i, s_j). Θ includes $\binom{Y}{2}$ combinations of two sources, i.e., pairs. We have two policies to select a combination of two sources out of Y sources, which are explained in the following.

Prior to explain the two policies, let $\rho(s_i, s_j)$ be the product of the NC gain and bandwidth demand for (s_i, s_j). We call $\rho(s_i, s_j)$ an effective gain, which is expressed by,

$$\rho(s_i, s_j) = G_{NC}(s_i, s_j) \times \min(\omega_{s_i d}, \omega_{s_j d}), \tag{3.1}$$

where $\omega_{s_i d}$ and $\omega_{s_j d}$ are the traffic demands of source nodes s_i and s_j to common destination node d, respectively. $\omega_{s_i d}$ and $\omega_{s_j d}$ are either equal or unequal.[1] In case that two traffic demands are not equal, the effective gain depends on the minimum traffic demand between the two.

3.1.5.1 Exhaustive search policy

- Step 1: For all $(s_i, s_j) \in \Theta$, compute $\rho(s_i, s_j)$.

- Step 2: From the $\binom{Y}{2}$ pairs in Θ, compute the sum of effective gains of all the $\lfloor \frac{Y}{2} \rfloor$ source pair (s_i, s_j) combinations, where each source appears only once. Note that $\lfloor x \rfloor$ is the largest integer not greater than x. The number of times we compute the sum of effective gains of $\lfloor \frac{Y}{2} \rfloor$ source pair combinations is expressed as $Z(Y)$,[2] which is given by,

$$Z(Y) = \begin{cases} \frac{Y!}{\frac{Y}{2}!2^{\frac{Y}{2}}}, & \text{if } Y \text{ is even} \\ \frac{Y!}{\frac{Y-1}{2}!2^{\frac{Y-1}{2}}}, & \text{if } Y \text{ is odd.} \end{cases} \tag{3.2}$$

- Step 3: From the $Z(Y)$ combinations of $\lfloor \frac{Y}{2} \rfloor$ pairs, pick up the combination with the highest total sum of effective gains.

---

[1]The ILP model presented in Appendix B is used when traffic demands are unequal.

[2]When Y is even, $Z(Y)$ is derived from, $Z(Y) = \frac{\prod_{k=0}^{k=\frac{Y-2}{2}} \binom{Y-2k}{2}}{\frac{Y}{2}!}$. We divide the numerator by $\frac{Y}{2}!$ in order to ensure that each source appears only once in the source pair combinations. When Y is odd, $Z(Y)$ is derived from $Z(Y) = Y \times Z(Y-1)$.

This exhaustive search policy provides the best pair selection. However, as the number of source node Y increases, the number of combinations $Z(Y)$ also increases abruptly. A selection policy based on a heuristic approach is required.

3.1.5.2 Largest effective gain policy

- Step 1: For all $(s_i, s_j) \in \Theta$, compute $\rho(s_i, s_j)$.
- Step 2: REPEAT
  - Step 2.1: Select a pair $(s_i, s_j) \in \Theta$ with the highest $\rho(s_i, s_j)$.
  - Step 2.2: Remove (s_i, s_j) and all other pairs that include either s_i or s_j from Θ.

  UNTIL Θ is empty.

At the end we select a set of $\lfloor \frac{Y}{2} \rfloor$ pairs with the highest effective gain in each of the case. The number of times we compute the effective gains is expressed as $W(Y)$,[1] which is given by,

$$W(Y) = \begin{cases} \frac{Y(Y-1)}{2}, & \text{if } Y \text{ is even} \\ \frac{Y(Y-1)(Y-2)}{2}, & \text{if } Y \text{ is odd.} \end{cases} \tag{3.3}$$

This policy has the computational complexity of the order of $O(Y^2)$ when Y is even, and $O(Y^3)$ when Y is odd. Note that the largest effective gain policy does not ensure the best two source combination selection.

3.2 Results and discussions

The performances of the presented heuristic algorithm is evaluated in terms of network coding gain (G_{NC}) and computation times. The G_{NC} achieved by our presented heuristic algorithm is compared with that of the conventional minimal-cost routing policy with NC. The computation times for the presented heuristic, the IQP, and ILP models are also compared in our examined networks (where

[1]When Y is even, $W(Y)$ is derived from, $W(Y) = \binom{Y}{2}$. When Y is odd, $W(Y)$ is derived from $W(Y) = Y \times W(Y-1) = Y \times \binom{Y-1}{2}$.

IQP and ILP are solvable). Moreover, we evaluate the G_{NC} and computation time of our heuristic algorithm on the number of nodes and average node degree in several large-scale random network topologies.

3.2.1 Heuristic algorithm's performance in large-scale random networks

In order to confirm the accuracy and optimality, we evaluate the performance of the presented algorithm in the five fixed network topologies used in the previous chapter, where ILP and IQP are solvable, against the results of ILP. The presented algorithm is implemented on Linux platform by the C programming language with a PC equipped with Intel® Core 2 Duo processor (3.17 GHz) and 4 GB RAM. The ILP formulation is solved by using CPLEX® as the solver [46] to obtain the optimum solution. We also measure the average computation time of ILP, Γ_{ILP}, in the same five networks.

The deviation Δ between the total path cost obtained from the heuristic algorithm ($\xi_{Heuristic}$) and that of the ILP model (ξ_{ILP}) is defined as the ratio:

$$\Delta = \frac{\xi_{Heuristic} - \xi_{ILP}}{\xi_{ILP}}, \tag{3.4}$$

where $\xi_{Heuristic} \geq \xi_{ILP}$.

By observing the results, it is confirmed that in almost 96% cases the presented algorithm provides the same optimum cost solution, as the ILP model in those five network topologies. In the rest of the 4% cases the heuristic algorithm provides marginal higher cost than that of ILP, and still this cost is smaller than the cost without NC. The maximum deviation of the heuristic cost from the optimum one is 0.009. The Γ_{ILP} is many times higher than that of the presented algorithm, $\Gamma_{Heuristic}$, in all the networks shown in Fig. 2.4. The deviation and average computation time results in the five networks are summarized in Table 3.1.

The performance of the presented algorithm is examined for large-scale networks that are randomly generated based on the *Barabasi-Albert* model [48] by using the BRITE internet topology generator [49], where any solution by ILP is impractical. Random networks with 75, 100, 150, and 200 nodes are used in our evaluation. We generate 10 random network topologies for every average node

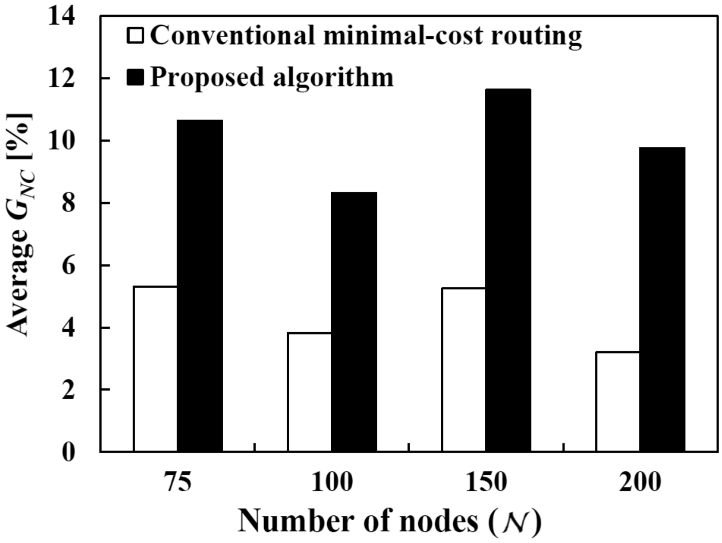


Figure 3.2: Network coding gain for number of nodes ($D_{avg} = 4$).

degree (D_{avg}) and total number of nodes ($\mathcal{N}$) combination. In order to compute average G_{NC}, we pick 50 arbitrary $2SD$ scenarios in each of the evaluated random topologies, and computed the G_{NC} averaged over 10 random topologies. The D_{avg} in the random topologies are kept fixed at four to evaluate the dependency of G_{NC} and $\Gamma_{Heuristic}$ on $\mathcal{N}$. G_{NC} of the presented algorithm is compared to that of the conventional minimal-cost routing policy, where both approaches enjoy the NC effect. For checking the dependency of G_{NC} and $\Gamma_{Heuristic}$ on D_{avg}, random network topologies with fixed $\mathcal{N}$ =150 nodes and varying average node degrees of D_{avg}=4, D_{avg}=6, D_{avg}=8, D_{avg}=10, and D_{avg}=12 are used. A node in the network acts as both source and destination. Link utilization cost is set to unity. The evaluation results are presented in the Figs. 3.2-3.5.

Using the presented algorithm, we obtain the maximum average G_{NC} of 13.1% in a $\mathcal{N}$ =150 node random network with D_{avg}=12 as shown in Fig. 3.3. The average gains for the four random networks with fixed D_{avg}=4 are around 10% as illustrated in Fig. 3.2. The results indicate that the presented algorithm achieves almost double resource saving effect as compared to the conventional minimal-cost routing policy.

Table 3.1: Deviations and average computation times

	Germany 17	Europe 28	Germany 50	US IP Backbone	COST239
Deviation ($\Delta^{\perp}$)	0	0.009	0.006	0.004	0.002
Γ_{IQP} (msec.)	27.3	41.7	148	160	1440
Γ_{ILP} (msec.)	9	18	42	67	263
$\Gamma_{Heuristic}$ (msec.)	$0^{\dagger}$	1	2	2	2

$\perp$ Heuristic algorithm has higher total path cost in almost 4% cases, otherwise equal to that of ILP.

† In the C program implementation, minimum computable time is 1 millisecond.

3.2.1.1 ILP is not solvable for large-scale networks due to insufficient memory

We evaluate the presented ILP in two random topologies of 75 and 100 nodes, in order to determine up to which network size the ILP is solvable in our used computation environment mentioned earlier. For both networks the presented ILP, which is an NP-Hard problem, is not solvable due to insufficient memory.

3.2.2 Observations

By carefully and thoroughly examining the evaluation results of the previous chapter and this chapter, the following points are observed:

- Table 3.1 displays the comparison of the presented ILP model and the presented algorithm with respect to achievable total cost deviation, and average computation time. As ILP searches all possible options to ensure the optimum solution, in the five examined networks (as shown in Fig. 2.4), Γ_{ILP} is higher than $\Gamma_{Heuristic}$. In the presented algorithm, the search range is limited as compared to the ILP model, which results in a faster computation time.

- The objective of conventional minimal-cost routing policy is to keep the costs of the two disjoint paths at minimum. Maintaining this objective, if

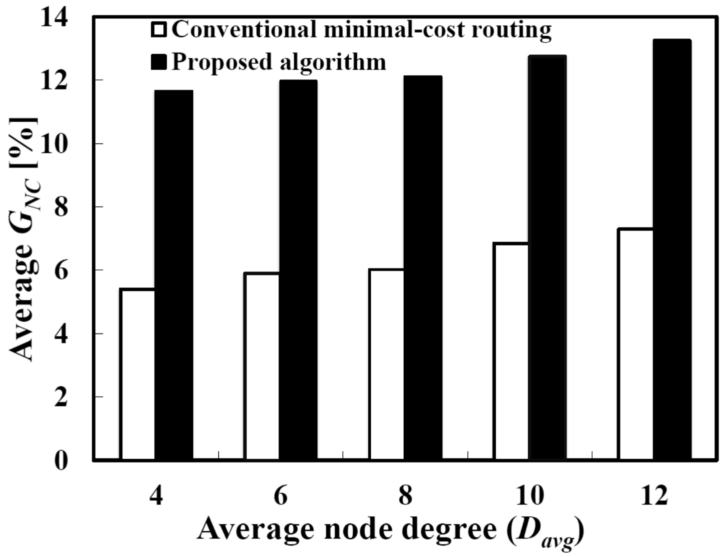


Figure 3.3: Network coding gain for D_{avg} ($\mathcal{N}$=150).

any NC opportunity occurs, NC is applied, and resources are saved. However, in the conventional routing policy based 1+1 protection NC opportunities are not frequently created, and this accounts for the low G_{NC}.

- In almost 96% cases in our observations the presented algorithm provides the same optimum cost solution obtained by the ILP model. In the rest 4% cases the presented algorithm provides a marginal higher cost than the optimal solution, but the cost is smaller than that of the conventional minimal-cost routing.

- Figures 3.2 and 3.3 illustrate that the D_{avg} is an influencing factor for a higher gain. The higher the D_{avg}, the higher the possibility of applying NC and save resources.

- $\Gamma_{Heuristic}$ is increased with the $\mathcal{N}$ in the network as illustrated in Fig. 3.4. The computation time decreases slowly with the increasing D_{avg} (Fig. 3.5). The rate at which the $\Gamma_{Heuristic}$ increases (with increasing $\mathcal{N}$) is higher than the rate at which it decreases (with the increasing D_{avg}). This means that, average computation time is affected more by the $\mathcal{N}$ than the D_{avg}.

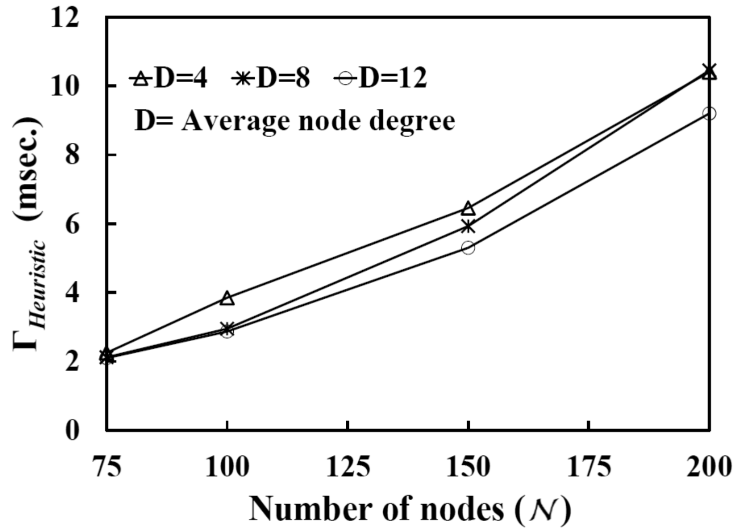


Figure 3.4: Computation time of presented algorithm for number of nodes.

- If the destination is close to the sources, and the two sources are also close to each other, the G_{NC} is relatively higher. If the destination is close to the sources, but the sources are far from each other, and located on the either side, NC opportunity reduces.
- If *Cost* ***without*** *NC* $\leq$ *Cost* ***with*** *NC*, 1+1 protection without NC is the preferred solution. For example, in the Germany 17 network shown in Fig. 2.4(a), consider nodes 7 and 14 as source nodes, and node 9 as the common destination node. The solution without NC has less cost (10) than that with NC (12). Therefore, the former is the best solution for this scenario.

3.3 NC based 1+1 protection in MSCD scenarios with $k \geq 2$ sources (kSD)

3.3.1 Routing solution

Previously we discussed NC based 1+1 protection in MSCD scenarios with $k = 2$ sources. We can extend our scheme for $k \geq 2$. For applying NC with 1+1

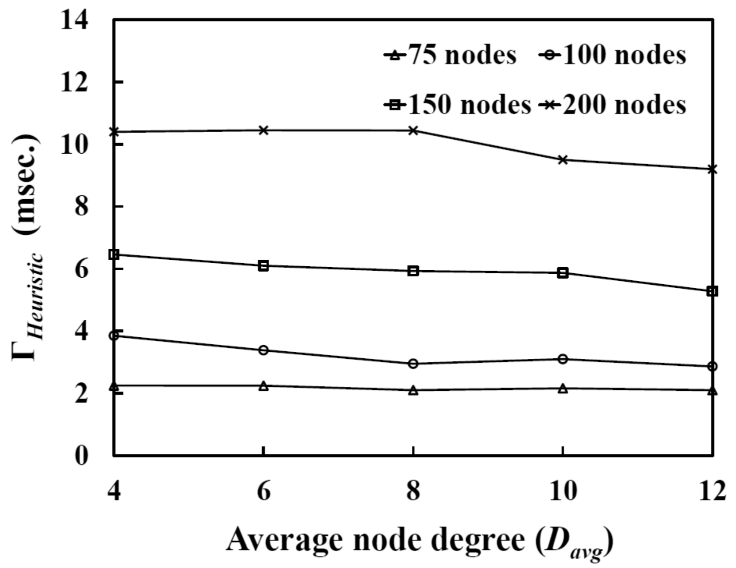


Figure 3.5: Computation time of presented algorithm for D_{avg}.

protection in any $kSD, k \geq 2$ the following conditions must be satisfied:

- k sources have a common destination.
- The intermediate nodes applying NC have node degree ≥ 3.
- The common destination node has node degree ≥ 3.
- At least two working paths carry only plain data to the common destination node.

The intermediate nodes have chances to employ any of the $\sum_{l=2}^{l=k} \binom{k}{l}$ XOR combinations for NC.

For any MSCD scenario with $k = 3$ sources, let nodes $1, 2$, and 3 be the three sources with X_1, X_2, and X_3, respectively, data to send to common destination D. Intermediate nodes need to consider $\sum_{l=2}^{l=3} \binom{3}{l} = 4$ XOR combinations ($X_1 \oplus X_2, X_1 \oplus X_3, X_2 \oplus X_3$, and $X_1 \oplus X_2 \oplus X_3$) for NC operations. Considering four XOR combinations and maintaining the condition that at least two paths must carry plain data, it is necessary to consider 13 different scenarios as shown in Fig. 4.1 of chapter 4. Note that the no NC case is also included in the 13 scenarios.

For $k = 4$, we need to consider $\sum_{l=2}^{l=4} \binom{4}{l} = 11$ XOR combinations and 83 scenarios. The number of XOR operations and scenarios increase abruptly with an increase in k value.

The routing solution presented in this chapter, for any $2SD$ scenario, can be extended to solve NC based 1+1 protection routing in any MSCD scenario with $k \geq 3$ sources. Thus for $k = 3$ we need to solve 13 different ILP formulations, for $k = 4$, 83 different ILP formulations, and so on. The ILP formulations are different because the objective function and path disjoint constraints in each ILP need to be adjusted according to the XOR operation combination(s) considered.

Let $S(k)$ be the set of scenarios for the number of sources $k \geq 2$. Let $\pi \in S(k)$ be a scenario. We solve an optimum routing problem considering NC, which is formulated by ILP, for each of $\pi \in S(k)$. A π with the minimum cost is selected as the most desired scenario. An optimum routing problem for $k \geq 2$ is expressed by,

$$\min_{k \geq 2} \min_{\pi \in S(k)} \min_{x \in X} f(k, \pi, x), \tag{3.5}$$

where $f(k, \pi, x)$ is the objective function of an routing problem with $\pi \in S(k)$ and k, x is a set of routing variables, and X is a set of x.

3.3.2 Complexity analysis

A feasible routing solution in any kSD scenario includes the combination of a Steiner tree, on which a common destination and k sources exist, and multiple disjoint paths from the common destination to the sources. These k disjoint paths must also be disjoint from the Steiner tree.

In order to discuss rigorously about the complexity of our considered network coding based 1+1 protection route design problem (NCR) in any kSD scenario, we first describe NCR as a decision (NP) problem, where k is a variable, and then by using the "polynomial-time reduction" we prove that NCR (with variable k) is an NP-Complete problem. The optimization version of NCR problem, denoted by Min-NCR, is not a decision problem. Therefor Min-NCR is an NP-Hard problrm. The detailed discussion about these proofs are included in Appendix B.

3.4 Summary

The ILP approach, presented in chapter 2, has two weak issues: one - it needs large amount of RAM and CPU processing power to solve and two - these RAM and CPU requirements increase with the number of nodes in the network. For large-scale network topologies, the ILP model is not practically solvable.

In order to tackle the above two issues, this chapter presented a heuristic algorithm to employ the NC technique efficiently with 1+1 protection in large-scale networks. It minimizes the network resource utilization, which in turn allows us to inject additional traffic into the network without creating congestion. The presented algorithm was evaluated and shown to perform satisfactorily in large networks, where any solution by ILP is not possible. The presented heuristic approach employs a two-stage procedure to find the best possible NC aware set of routes in large-scale network topologies. In the presented algorithm, the routes from the second source node to the nodes belonging to the first pair of disjoint paths between the first source and the common destination node are examined, to determine the best possible solution that minimizes the cost of provisioning 1+1 protection. The routing decisions in the presented algorithm create NC opportunities, which allow us to achieve a higher G_{NC} than that of the conventional minimal-cost routing policy. In the conventional routing policy, NC opportunities are purely incidental. This accounts for the low G_{NC}. The performance of the presented algorithm was thoroughly evaluated and analyzed in different network topologies with varying the number of nodes ($\mathcal{N}$), and the D_{avg}. Numerical results observed that our algorithm achieves almost double resource saving effect in the examined large networks, and the average computation time is faster than that of ILP.

Chapter 4

Heuristic algorithm to design protection routes for all possible source destination pairs

This chapter presents a heuristic routing algorithm to design instantaneous recovery protection routes for all possible source destination pairs by provisioning network coding (NC) based 1+1 protection technique with (MSCD scenarios). We consider a static routing problem in networks where each node has the coding capability, and the exact traffic demand matrix is given. In the presented heuristic algorithm a network with N nodes is divided into N scenarios, where each node is chosen as the common destination and k nodes among the remaining ones are the sources, where $2 \leq k \leq N-1$. By dividing the network into several MSCD scenarios with k sources, all the possible source destination pairs, according to the given traffic matrix, are considered. It was reported that a mathematical programming approach to determine NC based 1+1 protection routes for any kSD scenario is an intractable problem for large k values. In the presented heuristic algorithm we tackle this intractable problem by choosing either two or three sources out of k sources at a time according to the largest effective gain first policy, and then routing is assigned to the selected $2SD$ or $3SD$ scenario by using our developed mathematical models. The largest effective gain first policy ensures the best possible resource saving for each of the selected $2SD$ or $3SD$ scenario. We compare the total path costs of NC based 1+1 protection for all possible source destination pairs, obtained by our presented heuristic algorithm, with that of the conventional 1+1 protection technique (without NC). Numerical results observes that almost 15% resource saving is achieved in our examined networks.

4.1 Introduction

Literature reviews show that the NC technique is used for $1 + N$ protection [14, 15, 16], against single and multiple link failures [17, 18, 19, 20, 37], for multicast protection [23], and for NC aware routing in wireless networks [25, 28, 29, 33]. In [14, 15, 16] 1+1 protection without NC is considered as the conventional method of protection, and is compared to NC based $1 + N$ protection. In other works 1+1 protection with NC was not considered.

The issue that the NC technique is able to reduce the resource utilization in the resource hungry 1+1 protection, in MSCD scenarios with two sources ($2SD$), was first addressed in [13]. In order to determine an optimum set of NC aware 1+1 protection routes for $2SD$ scenarios, an integer linear programming (ILP) model was presented in chapter 2 and in [66]. It was also mentioned in [66] that the ILP approach for $2SD$ scenarios can be extended to solve NC based 1+1 protection routing in any MSCD scenario with $k \geq 3$ sources.

4.1.1 Issues to be addressed

There are two issues that were not addressed in a prior work [66]. One, no mathematical model for MSCD scenarios with $k \geq 3$ sources was presented in [66]. Only 13 possible NC situations for $3SD$ scenarios were presented in [66]. Two, NC based 1+1 protection route design for the whole network was not addressed. The resource saving reported in [66] considered several arbitrarily chosen $2SD$ scenarios only. Two problems can arise due to arbitrary choice. First, the same source node can appear in multiple $2SD$ scenarios considered. Second, it may not ensure the best possible resource saving when multiple connections having a common destination are served by selecting two sources arbitrarily at a time.

4.1.2 Contributions of this chapter

Our contributions consist of two parts. In the first part, we present ILP models for all the 13 NC situations for $3SD$ scenarios. In the second part, we present a heuristic routing algorithm to design instantaneous recovery protection routes for all possible source destination pairs by provisioning network coding (NC) based 1+1 protection technique. Note that the ILP models presented in the first part are used in the presented heuristic algorithm.

First, we describe a systematic analysis procedure for deriving mathematical models for the $3SD$ scenarios, where 4 XOR operations are considered for the NC operation. This procedure results in a framework of 13 mathematical models. Using our systematic analysis procedure it is possible to derive a mathematical model framework for $kSD, k \geq 4$ scenarios.

Second, we present a heuristic routing algorithm to implement NC based 1+1 protection routes for all possible source destination pairs in the network. In the presented algorithm a network with N nodes is divided into N scenarios, where each node is chosen as the common destination and k nodes among the remaining ones are the sources, where $2 \leq k \leq N-1$. For each of the N divided MSCD scenario with k sources (kSD), we choose either two or three sources out of k sources at a time according to the largest effective gain first policy, and then routing is assigned to the selected $2SD$ or $3SD$ scenario by using our developed mathematical models.

The remainder of this chapter is organized as follows. Section 4.2 introduces a systematic analysis procedure for analyzing various NC decision situations for $kSD, k \geq 3$, scenarios with the scenario classifications and descriptions. This analysis procedure is described for $3SD$ scenarios. Section 4.3 uses a network model to introduce the terminologies of this chapter. Section 4.4 illustrates the mathematical models for $3SD$ scenarios, which are derived from the analysis procedure. Section 4.5 describes the presented heuristic routing algorithm based on the largest effective gain first policy. Section 4.6 evaluates the performance of the presented algorithm in terms of the total path cost. Finally, section 4.7 summarizes the key points.

4.2 Systematic analysis procedure

The systematic analysis procedure for analyzing any kSD scenario, where $k \geq 3$, consists of three parts, which are described in the following:

- Classify the scenarios considering all possible combinations of XOR operations, and specify for each scenario that how many paths carry plain data, and how many carry encoded (NC) data.
- Analyze the scenarios properly to determine the specific paths that are subject to NC. Because path disjoint constraints are not imposed among these paths. Note that, for a kSD scenario, there are $2k$ paths, and $\binom{2k}{2}$ possible path disjoint constraints to ensure that $2k$ paths are mutually disjoint.
- Specify the objective function for every possible scenario considering the selected paths that are subject to NC, and construct the mathematical model. The flow conservation constraints for all the possible scenarios are the same.

The above procedure is illustrated with respect to the $3SD$ scenarios below.

4.2.1 Scenario classification

The $3SD$ network configurations are classified into five scenarios. From scenario 2 to scenario 5 each has three sub-scenarios. The scenarios and sub scenarios are shown in Fig. 4.1[1], and described as follows:

- Scenario 1 (Figs. 4.1(a)): Among the six paths, no path is subject to NC, i.e., non-NC scenario.
- Scenario 2 (Figs. 4.1(b),(c),(d)): Three backup paths from the three sources carry network coded data.
- Scenario 3 (Figs. 4.1(e),(f),(g)): Among the six paths, four paths experience the NC effect in pairs.

[1] This figure was presented in [66]. However, for the ease of readers, it is reproduced here.

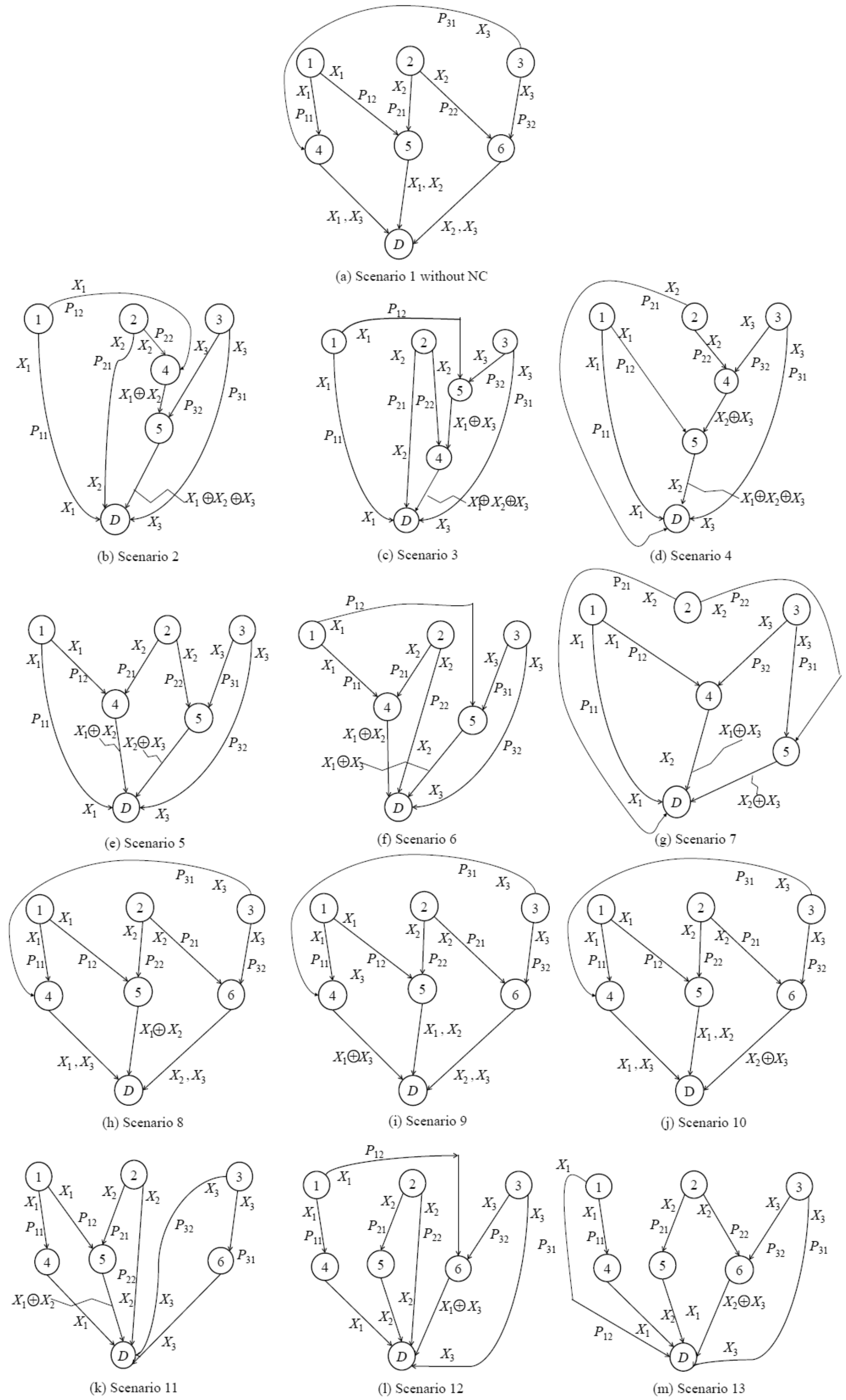


Figure 4.1: 13 possible NC situations for 3SD scenarios.

- Scenario 4 (Figs. 4.1(h),(i),(j)): Only two paths from any two sources are network coded, where common destination node's degree is three.

- Scenario 5 (Figs. 4.1(k),(l),(m)): Only two paths from any two sources are network coded, where common destination node's degree is five.

4.2.2 Determine the path disjoint constraints and the objective functions

One question arises: *How path disjoint constraints are assigned?* We answer this question with respect to the scenario shown in Fig. 4.1(b). In this scenario, for six paths $P_{sm}, s = 1, 2, 3, m = 1, 2$ to be mutually disjoint $\binom{2\times3}{2} = 15$ path disjoint constraints are needed . As shown in the Fig. 4.1(b), the paths indexed by P_{12}, P_{21}, and P_{31} experience the NC effect. Therefore, they need not be mutually disjoint. That is why we *do not impose* the following three path disjoint constraints in the mathematical model for this particular scenario:

- P_{12} and P_{21} are not disjoint.

- P_{12} and P_{31} are not disjoint.

- P_{21} and P_{31} are not disjoint.

Then the objective function is assigned correctly considering the selected constraints to get the proper NC effect.

The objective functions and the path disjoint constraints for all the 13 sub-scenarios are specified in Table 4.1. Note that the objective functions in the 12 NC cases are presented in quadratic form with respect to the selected path disjoint constraints. While forming ILP models these quadratic objective functions can be expressed into linear form by using the technique used in Eqs. (4.1f)-(4.1m). The flow conservation constraints in all the above mentioned 13 sub-scenarios are the same.

4.3 Network model

The network is represented as an undirected graph $G(V, E)$, where V is the set of vertices (nodes) and E is the set of links. There are N nodes and $|E|$ links in $G(V, E)$. A link from node $i \in V$ to node $j \in V$ is denoted as $(i, j) \in E, i \neq j$. $x_{ij}^{sd,m}$ is the binary routing variable. If any link $(i, j) \in E$ belongs to the disjoint path number m between source node s and destination node d, then value of $x_{ij}^{sd,m}$ is 1, otherwise 0, where m is either 1 or 2. c_{ij} is the cost of $(i, j) \in E$. It is assumed that the traffic demands for all possible source destination pairs are equal. The network is bi-connected, i.e., it is ensured that in the network, for every possible source-destination pair, there exist two disjoint paths for applying 1+1 protection. Every node with degree at least three has the NC capability, but encoded data are only decoded at the destination node. P_{sm} indicates the mth path between source node s and common destination node d, where m is either 1 or 2. Let $\mathcal{S}$ be a set of vectors (a, b, c, e), where a and c are source node indexes, and b and e are path indexes having a value either 1 or 2. a and c can have any integer value from 1 to k, where k is the number of source nodes having a common destination. Each vector $(a, b, c, e) \in \mathcal{S}$ corresponds to a path disjoint constraint for the scenario under consideration. $\mathcal{S}$ is an input parameter to the mathematical model.

4.4 Mathematical formulations for the $3SD$ scenarios

This section describes how the mathematical programming formulation is derived for the scenarios and sub-scenarios described in Fig. 4.1. Following the objective function and the path disjoint constraints for the 13 sub-scenarios are summarized in Table 4.1, we present the integer linear programming (ILP) formulation for the scenario presented in Fig. 4.1(b) in the following.

$$\begin{aligned}
\mathbf{min} \quad & \sum_{(i,j)\in E} \sum_{m=1}^{2} c_{ij} \times (x_{ij}^{s_1 d,m} + x_{ij}^{s_2 d,m} + x_{ij}^{s_3 d,m}) \\
& - \sum_{(i,j)\in E} c_{ij} \times (z_{ij}^{1} + z_{ij}^{2}) \qquad (4.1a)
\end{aligned}$$

subject to:

$$x_{ij}^{s_a d,b} + x_{ij}^{s_c d,e} \leq 1, \forall (i,j) \in E, \forall (a,b,c,e) \in \mathcal{S} \qquad (4.1b)$$

$$\sum_{j\in V} (x_{ij}^{sd,m} - x_{ji}^{sd,m}) = 1, \forall i, s \in V, \quad i = s,\ s = s_1, s_2, s_3, m = 1,2 \qquad (4.1c)$$

$$\sum_{j\in V} (x_{ij}^{sd,m} - x_{ji}^{sd,m}) = 0, \quad \forall i, s \in V, \quad i \neq s, d,\ s = s_1, s_2, s_3, m = 1,2 \qquad (4.1d)$$

$$x_{ij}^{sd,m} = \{0,1\}, \forall (i,j) \in E, \quad s = s_1, s_2, s_3, m = 1,2 \qquad (4.1e)$$

$$z_{ij}^{1} = \{0,1\},\ \forall (i,j) \in E \qquad (4.1f)$$

$$z_{ij}^{1} \leq x_{ij}^{s_1 d,2},\ \forall (i,j) \in E \qquad (4.1g)$$

$$z_{ij}^{1} \leq x_{ij}^{s_2 d,2},\ \forall (i,j) \in E \qquad (4.1h)$$

$$z_{ij}^{1} \geq x_{ij}^{s_1 d,2} + x_{ij}^{s_2 d,2} - 1,\ \forall (i,j) \in E \qquad (4.1i)$$

$$z_{ij}^{2} = \{0,1\},\ \forall (i,j) \in E \qquad (4.1j)$$

$$z_{ij}^{2} \leq x_{ij}^{s_1 d,2},\ \forall (i,j) \in E \qquad (4.1k)$$

$$z_{ij}^{2} \leq x_{ij}^{s_3 d,2},\ \forall (i,j) \in E \qquad (4.1l)$$

$$z_{ij}^{2} \geq x_{ij}^{s_1 d,2} + x_{ij}^{s_3 d,2} - 1,\ \forall (i,j) \in E. \qquad (4.1m)$$

Eq. (4.1a) is the objective function which provides optimum minimum cost of employing NC aware disjoint path pairs among three sources and a common destination. Eq. (4.1b) specifies the path disjoint constraints while flow conservation constraints are specified by the Eqs. (4.1c)-(4.1d). Eq. (4.1c) is the flow conservation constraint at the source nodes. Eq. (4.1d) describes the flow conservation constraints at the intermediate nodes. The binary routing variable is described by Eq. (4.1e). Eqs. (4.1f)-(4.1m) describe the binary variables, z_{ij}^1 and z_{ij}^2, that determine the proper links to employ NC. These two binary variables can be defined for each scenario following the objective function in the Table 4.1.

The set of vectors $\mathcal{S}$ for the presented ILP model is given as: $\mathcal{S} = \{(1,1,1,2),$ $(2,1,2,2)$, $(1,1,2,1)$, $(1,1,2,2)$, $(1,2,2,1)$, $(1,1,3,1)$, $(1,1,3,2)$, $(2,2,3,2)$, $(2,2,3,1)$, $(1,2,3,1)$, $(2,1,3,2)$, $(3,1,3,2)\}$.

For each of the 13 sub-scenarios (which includes non-NC case) in the $3SD$ case, an optimal mathematical formulation can be derived with the help of Table 4.1. All the sub-scenarios are evaluated using the specific model, and the solution with the minimum cost is selected as the desired solution.

4.5 Presented heuristic routing algorithm

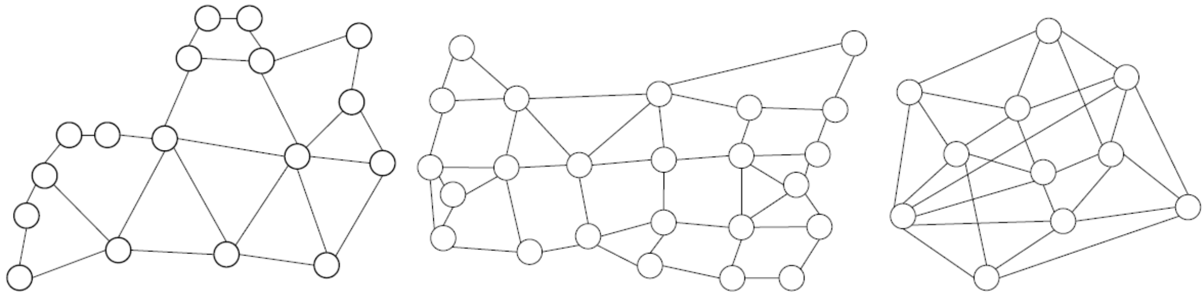
We present our presented heuristic routing algorithm in order to implement NC aware 1+1 protection routes for all possible source destination pair in a network. According to the given traffic demand matrix, the network with N nodes is divided into N scenarios, where each nodes is chosen as a common destination and k nodes from the remaining ones are the sources, where $2 \leq k \leq N-1$. Assigning NC based 1+1 routes for each of the kSD scenario implies that routing for all possible source destination pairs are considered. We assume that the network has sufficient capacity to serve all the traffic demands. Thus link capacity constraints are not considered.

In the presented algorithm the ILP models for $2SD$ and $3SD$ scenarios are used. This is because for $kSD, k \geq 4$ scenarios the number of ILPs to be considered increases exponentially, which are not tractable in a practical time [66].

4.5.1 Algorithm description

The algorithm is as follows,

- Step 1: Divide the given traffic matrix into N MSCD scenarios with k sources, where $2 \leq k \leq N-1$. For each of the kSD scenario, repeat step 2.



(a) Germany 17 network (b) US IP Backbone network (c) COST 239 network

Figure 4.2: Evaluated networks

- Step 2: Select x sources, where $x = 2$ or 3, out of k sources at a time according to the largest effective gain first policy (to be explained later).
  - Step 2.1: Assign routing to the selected xSD scenario by using the corresponding ILP models presented in chapter 2 [66] (for $x = 2$) or in this chapter (for $x = 3$). Update k as $k = k - x$, which is the remaining number of source nodes.
  - Step 2.2: Select the next x sources from the remaining sources, and assign routing according to step 2.1.
  - Step 2.3: Repeat Step 2.2 until $k - x$ equals 2, 1, or 0. When only 1 source is left, 1+1 protection without NC is applied for that pair. If two sources are left (in case $x = 3$), NC based 1+1 protection is applied for this remaining $2SD$ scenario.
- Step 3: The algorithm stops.

4.5.2 Policy to select x sources

We describe here the largest effective gain first policy to select $x = 2$ or 3 sources out of k sources. Prior to describe the policy, we describe the related definitions and notations in the following.

Let φ be the set of source nodes having a common destination, where $s_i \in \varphi, i = 1, 2, \cdots, k$, where $k \geq 2$. A combination of two sources, s_i and s_j, out of

k sources, is expressed by $(s_i, s_j) \in \Theta_2, (i < j)$, where Θ_2 is a set of (s_i, s_j). Θ_2 includes $\binom{k}{2}$ combinations of two sources, i.e., pairs. Again, a combination of three sources, s_i, s_j, and s_l, out of k sources, is expressed by $(s_i, s_j, s_l) \in \Theta_3, (i < j < l)$, where Θ_3 is a set of (s_i, s_j, s_l). Θ_3 includes $\binom{k}{3}$ combinations of three sources, i.e., triplets.

Let $\rho_2(s_i, s_j)$ be the product of the NC gain and bandwidth demand for (s_i, s_j). We call $\rho_2(s_i, s_j)$ an effective gain for a $2SD$ scenario, which is expressed by,

$$\rho_2(s_i, s_j) = G_{NC}(s_i, s_j) \times \min(\omega_{s_i d}, \omega_{s_j d}), \tag{4.2}$$

where $\omega_{s_i d}$ and $\omega_{s_j d}$ are the traffic demands of source nodes s_i and s_j to common destination node d, respectively. $\omega_{s_i d}$ and $\omega_{s_j d}$ are either equal or unequal. In case that two traffic demands are not equal, the effective gain depends on the minimum traffic demand between the two.

Let $\rho_3(s_i, s_j, s_l)$ be the product of the NC gain and bandwidth demand for (s_i, s_j, s_l). We call $\rho_3(s_i, s_j, s_l)$ an effective gain for a $3SD$ scenario, which is expressed by,

$$\rho_3(s_i, s_j, s_l) = G_{NC}(s_i, s_j, s_l) \times \min(\omega_{s_i d}, \omega_{s_j d}, \omega_{s_l d}), \tag{4.3}$$

where $\omega_{s_i d}$, $\omega_{s_j d}$, and $\omega_{s_j d}$ are the traffic demands of source nodes s_i, s_j, and s_l to common destination node d, respectively. $\omega_{s_i d}$, $\omega_{s_j d}$, and $\omega_{s_l d}$ are either equal or unequal.[1] In case that three traffic demands are not equal, the effective gain depends on the minimum traffic demand between the three.

4.5.2.1 Largest effective gain first policy

- Step 1: For all $(s_i, s_j) \in \Theta_2$, compute $\rho_2(s_i, s_j)$, or for all $(s_i, s_j, s_l) \in \Theta_3$ compute $\rho_3(s_i, s_j, s_l)$. If we compute $\rho_2(s_i, s_j)$, then go to Step 2. If $\rho_3(s_i, s_j, s_l)$ is computed, go to Step 3.

- Step 2: REPEAT

[1] In this chapter we consider equal traffic demands for all the pairs. However, our presented model for $3SD$ scenarios is applicable for unequal traffic demands. We have to add the related constraints.

  - Step 2.1: Select a pair $(s_i, s_j) \in \Theta_2$ with the highest $\rho_2(s_i, s_j)$.
  - Step 2.2: Remove (s_i, s_j) and all other pairs that include either s_i or s_j from Θ_2.

UNTIL Θ_2 is empty.

- Step 3: REPEAT
  - Step 3.1: Select a pair $(s_i, s_j, s_l) \in \Theta_3$ with the highest $\rho_3(s_i, s_j, s_l)$.
  - Step 2.2: Remove (s_i, s_j, s_l) and all other triplets that include either s_i or s_j or s_l from Θ_3.

UNTIL Θ_3 is empty.

At the end we select a set of $\lfloor \frac{k}{2} \rfloor$ pairs or $\lfloor \frac{k}{3} \rfloor$ triplets with the highest effective gain in each of the case. The number of times we compute the effective gains for pairs $\in \Theta_2$ is expressed as $W_2(k)$,[1] which is given by,

$$W_2(k) = \begin{cases} \frac{k(k-1)}{2}, & \text{if } k \text{ is even} \\ \frac{k(k-1)(k-2)}{2}, & \text{if } k \text{ is odd.} \end{cases} \tag{4.4}$$

The number of times we compute the effective gains for triplets $\in \Theta_3$ is expressed as $W_3(k)$, which is given by,

$$W_3(k) = \begin{cases} \frac{k(k-1)(k-2)}{6}, & \text{if } k = 3 \times z, z\text{=integer} \\ \frac{k(k-1)(k-2)(k-3)}{6}, & \text{if } k = 3 \times z + 1 \\ \frac{k(k-1)(k-2)(k-3)(k-4)}{12}, & \text{if } k = 3 \times z + 2. \end{cases} \tag{4.5}$$

When we select a pair at a time, this policy has the computational complexity of the order of $O(k^2)$ when k is even, and $O(k^3)$ when Y is odd. When we select a triplet at a time, this policy has the computational complexity of the order of $O(k^3)$ when $k = 3 \times z$, $O(k^4)$ when $k = 3 \times z + 1$, and $O(k^5)$ when $k = 3 \times z + 2$.

[1] When k is even, $W_2(k)$ is derived from, $W_2(k) = \binom{k}{2}$. When k is odd, $W_2(k)$ is derived from $W_2(k) = k \times W_2(k-1) = k \times \binom{k-1}{2}$. A similar computation is applicable to $W_3(k)$.

4.6 Results and discussion

We compute the total path cost of implementing NC based 1+1 protection for all possible source destination pairs by using our presented heuristic algorithm, where either a $2SD$ or a $3SD$ scenario is selected one by one according to the largest effective gain first policy. We compared these total path costs, achieved by our presented algorithm, to the same total cost when conventional 1+1 protection technique without NC is employed. We evaluated the above mentioned total costs in the three networks as shown in Fig. 4.2. The mathematical models are solved by using CPLEX®[46] as the LP solver. In the evaluation all nodes in the network are assumed to have NC capability. In the network topologies, link utilization cost is unity, and the traffic demands are the same for all source-destination pairs.

From the results of Fig. 4.3 it is observed that, the presented algorithm with the $3SD$ scenario achieves the highest resource saving effect in our examined networks. Figure 4.3 illustrates that almost 15% resource saving is achieved in the COST 239 network by selecting three sources at a time with the largest effective gain first policy. In the COST 239 network the smallest node degree is four, and all the source destination pairs in this network can avail the resource saving advantage of the NC based 1+1 protection technique. In the US IP Backbone and Germany 17 networks, there are a number of pairs to which NC based 1+1 protection technique is not applicable, because for those pairs the destination node degree is only two. Thus some of the $2SD$ and $3SD$ scenarios cannot obtain the benefit of resource saving due to NC in these two networks. This is the reason of lower resource saving in the US IP backbone and Germany 17 networks.

4.7 Summary

This chapter has presented a heuristic routing algorithm to design instantaneous recovery protection routes for all possible source destination pairs by provisioning network coding (NC) based 1+1 protection technique. In the presented algorithm, a network with N nodes is divided into N scenarios, where each node is chosen as a common destination, and k nodes from the remaining ones are selected as

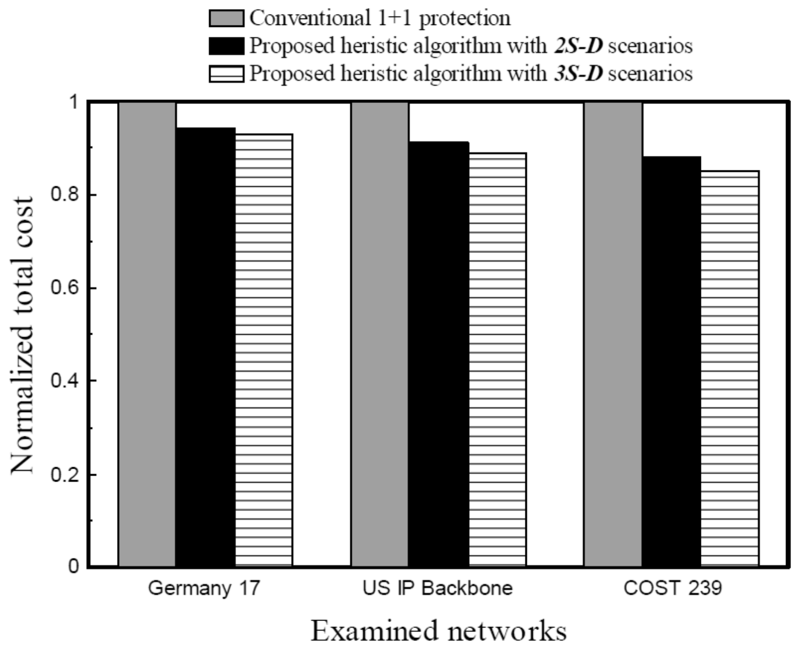


Figure 4.3: Comparison of total costs. The costs are normalized w.r.t. the total cost of 1+1 protection.

sources, where $2 \leq k \leq N-1$. A mathematical programming solution for NC based 1+1 routing in MSCD scenarios with k sources (kSD) is an intractable problem for large k values. In the presented algorithm, for each of the divided kSD scenario, out of k sources we choose either two or three sources at a time according to the largest effective gain first policy, and then routing is assigned to the selected $2SD$ or $3SD$ scenario using the respective mathematical formulations. We compared the total path costs of NC based 1+1 protection for all possible source destination pairs, obtained by our presented heuristic algorithm, with that of the conventional 1+1 protection technique (without NC). Almost 15% resource saving w.r.t. 1+1 protection is achieved in our examined networks. The presented heuristic algorithm with three sources selected at a time achieves a higher resource saving as compared to that with two sources selected at a time.

Table 4.1: Objective Functions and Constraints

Scenario (SC) No.	Objective Function	Disjoint Path Pairs
SC 1	$\sum_{(i,j)\in E} \sum_{m=1}^{2} C_{ij} \times (x_{ij}^{s_1 d,m} + x_{ij}^{s_2 d,m} + x_{ij}^{s_3 d,m})$	$P_{11}\&P_{12}$, $P_{21}\&P_{22}$, $P_{31}\&P_{32}$
SC 2 Sub-SC 1	$\sum_{(i,j)\in E} \sum_{m=1}^{2} C_{ij} \times (x_{ij}^{s_1 d,m} + x_{ij}^{s_2 d,m} + x_{ij}^{s_3 d,m}) - \sum_{(i,j)\in E}(C_{ij} \times x_{ij}^{s_1 d,2} \times x_{ij}^{s_2 d,1} + C_{ij} \times x_{ij}^{s_1 d,2} \times x_{ij}^{s_3 d,1})$	$P_{11}\&P_{22}$, $P_{11}\&P_{32}$, $P_{11}\&P_{12}$, $P_{11}\&P_{21}$, $P_{11}\&P_{31}$, $P_{22}\&P_{32}$, $P_{22}\&P_{12}$, $P_{22}\&P_{21}$, $P_{22}\&P_{31}$, $P_{32}\&P_{12}$, $P_{32}\&P_{21}$, $P_{32}\&P_{31}$
SC 2 Sub-SC 2	$\sum_{(i,j)\in E} \sum_{m=1}^{2} C_{ij} \times (x_{ij}^{s_1 d,m} + x_{ij}^{s_2 d,m} + x_{ij}^{s_3 d,m}) - \sum_{(i,j)\in E}(C_{ij} \times x_{ij}^{s_1 d,2} \times x_{ij}^{s_3 d,1} + C_{ij} \times x_{ij}^{s_2 d,1} \times x_{ij}^{s_3 d,1})$	$P_{11}\&P_{22}$, $P_{11}\&P_{32}$, $P_{11}\&P_{12}$, $P_{11}\&P_{21}$, $P_{11}\&P_{31}$, $P_{22}\&P_{32}$, $P_{22}\&P_{12}$, $P_{22}\&P_{21}$, $P_{22}\&P_{31}$, $P_{32}\&P_{12}$, $P_{32}\&P_{21}$, $P_{32}\&P_{31}$
SC 2 Sub-SC 3	$\sum_{(i,j)\in E} \sum_{m=1}^{2} C_{ij} \times (x_{ij}^{s_1 d,m} + x_{ij}^{s_2 d,m} + x_{ij}^{s_3 d,m}) - \sum_{(i,j)\in E}(C_{ij} \times x_{ij}^{s_2 d,1} \times x_{ij}^{s_3 d,1} + C_{ij} \times x_{ij}^{s_1 d,2} \times x_{ij}^{s_2 d,1})$	$P_{11}\&P_{22}$, $P_{11}\&P_{32}$, $P_{11}\&P_{12}$, $P_{11}\&P_{21}$, $P_{11}\&P_{31}$, $P_{22}\&P_{32}$, $P_{22}\&P_{12}$, $P_{22}\&P_{21}$, $P_{22}\&P_{31}$, $P_{32}\&P_{12}$, $P_{32}\&P_{21}$, $P_{32}\&P_{31}$
SC 3 Sub-SC 1	$\sum_{(i,j)\in E} \sum_{m=1}^{2} C_{ij} \times (x_{ij}^{s_1 d,m} + x_{ij}^{s_2 d,m} + x_{ij}^{s_3 d,m}) - \sum_{(i,j)\in E}(C_{ij} \times x_{ij}^{s_1 d,2} \times x_{ij}^{s_2 d,1} + C_{ij} \times x_{ij}^{s_2 d,2} \times x_{ij}^{s_3 d,1})$	$P_{11}\&P_{32}$, $P_{11}\&P_{12}$, $P_{11}\&P_{21}$, $P_{11}\&P_{22}$, $P_{11}\&P_{31}$, $P_{32}\&P_{12}$, $P_{32}\&P_{21}$, $P_{32}\&P_{22}$, $P_{32}\&P_{31}$, $P_{12}\&P_{22}$, $P_{12}\&P_{31}$, $P_{21}\&P_{22}$, $P_{21}\&P_{31}$
SC 3 Sub-SC 2	$\sum_{(i,j)\in E} \sum_{m=1}^{2} C_{ij} \times (x_{ij}^{s_1 d,m} + x_{ij}^{s_2 d,m} + x_{ij}^{s_3 d,m}) - \sum_{(i,j)\in E}(C_{ij} \times x_{ij}^{s_1 d,1} \times x_{ij}^{s_2 d,1} + C_{ij} \times x_{ij}^{s_1 d,2} \times x_{ij}^{s_3 d,1})$	$P_{22}\&P_{32}$, $P_{22}\&P_{11}$, $P_{22}\&P_{21}$, $P_{22}\&P_{12}$, $P_{22}\&P_{31}$, $P_{32}\&P_{11}$, $P_{32}\&P_{21}$, $P_{32}\&P_{12}$, $P_{32}\&P_{31}$, $P_{11}\&P_{12}$, $P_{11}\&P_{31}$, $P_{21}\&P_{12}$, $P_{21}\&P_{31}$
SC 3 Sub-SC 3	$\sum_{(i,j)\in E} \sum_{m=1}^{2} C_{ij} \times (x_{ij}^{s_1 d,m} + x_{ij}^{s_2 d,m} + x_{ij}^{s_3 d,m}) - \sum_{(i,j)\in E}(C_{ij} \times x_{ij}^{s_1 d,2} \times x_{ij}^{s_3 d,1} + C_{ij} \times x_{ij}^{s_2 d,2} \times x_{ij}^{s_3 d,2})$	$P_{11}\&P_{21}$, $P_{11}\&P_{12}$, $P_{11}\&P_{31}$, $P_{11}\&P_{22}$, $P_{11}\&P_{32}$, $P_{21}\&P_{12}$, $P_{21}\&P_{31}$, $P_{21}\&P_{22}$, $P_{21}\&P_{32}$, $P_{12}\&P_{22}$, $P_{12}\&P_{32}$, $P_{31}\&P_{22}$, $P_{31}\&P_{32}$
SC 4 Sub-SC 1	$\sum_{(i,j)\in E} \sum_{m=1}^{2} C_{ij} \times (x_{ij}^{s_1 d,m} + x_{ij}^{s_2 d,m} + x_{ij}^{s_3 d,m}) - \sum_{(i,j)\in E}(C_{ij} \times x_{ij}^{s_1 d,2} \times x_{ij}^{s_2 d,1})$	$P_{11}\&P_{12}$, $P_{11}\&P_{21}$, $P_{11}\&P_{22}$, $P_{11}\&P_{32}$, $P_{31}\&P_{12}$, $P_{31}\&P_{21}$, $P_{31}\&P_{22}$, $P_{31}\&P_{32}$, $P_{12}\&P_{32}$,, $P_{12}\&P_{22}$, $P_{21}\&P_{22}$, $P_{21}\&P_{32}$
SC 4 Sub-SC 2	$\sum_{(i,j)\in E} \sum_{m=1}^{2} C_{ij} \times (x_{ij}^{s_1 d,m} + x_{ij}^{s_2 d,m} + x_{ij}^{s_3 d,m}) - \sum_{(i,j)\in E}(C_{ij} \times x_{ij}^{s_1 d,1} \times x_{ij}^{s_3 d,1})$	$P_{11}\&P_{12}$, $P_{11}\&P_{21}$, $P_{11}\&P_{22}$, $P_{11}\&P_{32}$, $P_{31}\&P_{12}$, $P_{31}\&P_{21}$, $P_{31}\&P_{22}$, $P_{31}\&P_{32}$, $P_{12}\&P_{32}$, $P_{12}\&P_{22}$, $P_{21}\&P_{22}$, $P_{21}\&P_{32}$
SC 4 Sub-SC 3	$\sum_{(i,j)\in E} \sum_{m=1}^{2} C_{ij} \times (x_{ij}^{s_1 d,m} + x_{ij}^{s_2 d,m} + x_{ij}^{s_3 d,m}) - \sum_{(i,j)\in E}(C_{ij} \times x_{ij}^{s_2 d,2} \times x_{ij}^{s_3 d,2})$	$P_{11}\&P_{12}$, $P_{11}\&P_{21}$, $P_{11}\&P_{22}$, $P_{11}\&P_{32}$, $P_{31}\&P_{12}$, $P_{31}\&P_{21}$, $P_{31}\&P_{22}$, $P_{31}\&P_{32}$, $P_{12}\&P_{32}$, $P_{12}\&P_{22}$, $P_{21}\&P_{22}$, $P_{21}\&P_{32}$
SC 5 Sub-SC 1	$\sum_{(i,j)\in E} \sum_{m=1}^{2} C_{ij} \times (x_{ij}^{s_1 d,m} + x_{ij}^{s_2 d,m} + x_{ij}^{s_3 d,m}) - \sum_{(i,j)\in E}(C_{ij} \times x_{ij}^{s_1 d,2} \times x_{ij}^{s_2 d,1})$	$P_{11}\&P_{12}$, $P_{11}\&P_{21}$, $P_{11}\&P_{22}$, $P_{11}\&P_{31}$, $P_{11}\&P_{32}$, $P_{12}\&P_{22}$, $P_{12}\&P_{31}$, $P_{12}\&P_{32}$, $P_{21}\&P_{22}$, $P_{21}\&P_{31}$, $P_{21}\&P_{32}$, $P_{22}\&P_{31}$, $P_{22}\&P_{32}$, $P_{31}\&P_{32}$
SC 5 Sub-SC 2	$\sum_{(i,j)\in E} \sum_{m=1}^{2} C_{ij} \times (x_{ij}^{s_1 d,m} + x_{ij}^{s_2 d,m} + x_{ij}^{s_3 d,m}) - \sum_{(i,j)\in E}(C_{ij} \times x_{ij}^{s_1 d,2} \times x_{ij}^{s_3 d,1})$	$P_{11}\&P_{12}$, $P_{11}\&P_{21}$, $P_{11}\&P_{22}$, $P_{11}\&P_{31}$, $P_{11}\&P_{32}$, $P_{12}\&P_{21}$, $P_{12}\&P_{22}$, $P_{12}\&P_{32}$, $P_{21}\&P_{22}$, $P_{21}\&P_{31}$, $P_{21}\&P_{32}$, $P_{22}\&P_{31}$, $P_{22}\&P_{32}$, $P_{31}\&P_{32}$
SC 5 Sub-SC 3	$\sum_{(i,j)\in E} \sum_{m=1}^{2} C_{ij} \times (x_{ij}^{s_1 d,m} + x_{ij}^{s_2 d,m} + x_{ij}^{s_3 d,m}) - \sum_{(i,j)\in E}(C_{ij} \times x_{ij}^{s_2 d,2} \times x_{ij}^{s_3 d,1})$	$P_{11}\&P_{12}$, $P_{11}\&P_{21}$, $P_{11}\&P_{22}$, $P_{11}\&P_{31}$, $P_{11}\&P_{32}$, $P_{12}\&P_{21}$, $P_{12}\&P_{22}$, $P_{12}\&P_{31}$, $P_{12}\&P_{32}$, $P_{21}\&P_{22}$, $P_{21}\&P_{31}$, $P_{21}\&P_{32}$, $P_{22}\&P_{32}$, $P_{31}\&P_{32}$

Chapter 5

Mathematical model for $2SD$ scenarios by relaxing node degree constraints

An integer linear programming (ILP) model to determine network coding (NC) aware minimum cost routes, for employing 1+1 protection in MSCD scenarios with two sources ($2SD$), was addressed in chapter 2. In this chapter, we refer to the ILP model in chapter 2 as a conventional model. When the common destination's node degree is only two, NC cannot be employed, because failure of an adjacent link carrying original data restricts recovery operation at the destination. This chapter presents a mathematical model for optimum NC aware 1+1 protection routing in $2SD$ scenarios with destination's node degree ≥ 2. Numerical results observe that our presented model achieves 5.5% of resource saving, averaged over all possible $2SD$ scenarios with destination's degree = 2, compared to the conventional model, in our examined networks.

5.1 Introduction

In today's high speed backbone networks failure of a network component results in a loss of large amount of data and causes service interruptions. Instantaneous recovery from network failures is a key feature that the service providers must offer to the customers needing reliable communication.

1+1 protection provides instantaneous recovery from any single link failure in the network by sending copy of the same data through two disjoint paths [1]. Instantaneous recovery is achieved, because only the destination node switches to the backup path after a failure is detected. However, this instantaneous recovery benefit is obtained at the cost of at least double resources.

The fact that adaptation of the network coding (NC) technique [12] with routing reduces the resources required for dedicated protection techniques has already been addressed [13, 35, 38].

A mathematical model to determine NC aware optimum routes for 1+1 protection in MSCD scenarios with two sources ($2SD$), where the common destination's node degree ≥ 3, was presented in chapter 2 and [35]. We name the model in [35] as a conventional model[1]. However, the *common destination's node degree of two* restricts the use of NC with 1+1 protection, and it is explained in the following.

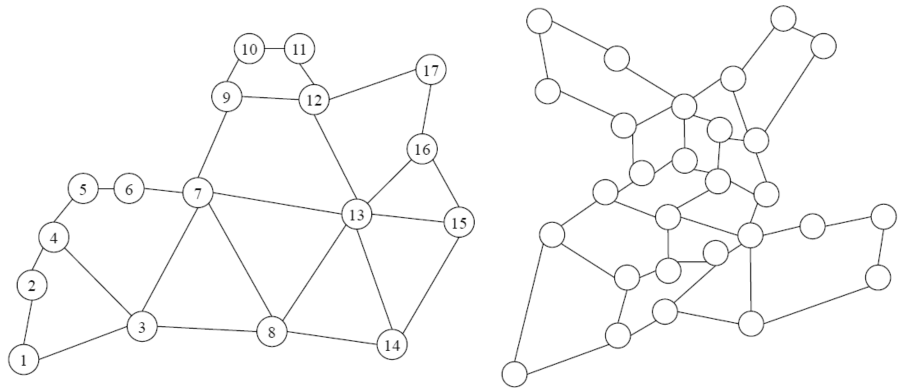
The Germany 17 network of Fig. 5.1(a), presented in [35, 66] and reproduced here, is used to explain why destination's node degree of two restricts the use of NC with 1+1 protection in any $2SD$ scenario. Let nodes 7 and 14 be the two source nodes and node 17 be the common destination node with degree two. For employing 1+1 protection, node 7 determines two disjoint paths as $7 \rightarrow 13 \rightarrow 16 \rightarrow 17$ and $7 \rightarrow 9 \rightarrow 12 \rightarrow 17$. For node 14, the two disjoint paths are $14 \rightarrow 15 \rightarrow 16 \rightarrow 17$ and $14 \rightarrow 13 \rightarrow 12 \rightarrow 17$.

Let the source data corresponding to node 7 be X_1 and that of for node 14 be X_2. Both nodes 12 and 16, adjacent nodes to common destination node 17, receive X_1 and X_2, and are capable of applying NC. Let node 12 apply NC and create network coded data as $X_1 \oplus X_2$ to save resources along the link $12 \rightarrow 17$. If link $16 \rightarrow 17$ fails, the destination node receives only network coded data $X_1 \oplus X_2$ along link $12 \rightarrow 17$, and it cannot perform the recovery operation due to the lack of any source data (X_1 or X_2).

When common destination node's degree is two, the conventional model cannot provide an NC aware routing solution for 1+1 protection in any $2SD$ scenario. This is because, in such a scenario if an adjacent link (carrying plain data) to the common destination is failed, recovery operation is not possible. Note that

[1] The work of [35] is extended in [66]. The extended work adopts the same routing approach as that of [35].

node degree of two is the minimum requirement for the implementation of 1+1 protection.



a) Germany 17 network. b) European 28 network.

Figure 5.1: Examined networks.

This chapter presents a mathematical model, for any $2SD$ scenario with destination's node degree ≥ 2, to determine NC aware disjoint routes for employing 1+1 protection. The presented model determines an optimum set of routes that minimizes the required network resources. In our presented model, an Integer Linear Programming (ILP) formulation is iteratively used. The cost obtained by our presented model is compared with that of the conventional model. Numerical results observe that our presented model achieves 5.5% of resource saving, averaged over all possible $2SD$ scenarios with destination's degree = 2, compared to the conventional model, in our examined networks.

5.2 Network coding (NC) aware route selection technique

Figure 5.2 illustrates how NC aware routes are selected for employing 1+1 protection in a $2SD$ scenario, where nodes S_1 and S_2 are the sources and D is the common destination with degree two. Source nodes S_1 and S_2 send their original data (X_1 and X_2, respectively) to a common intermediate destination node t. In the route selection process, NC is applied at the intermediate node E. Decoding is performed at node t. There are two disjoint paths between t and original destination node D. After the encoded data $(X_1 \oplus X_2)$ are decoded at t (if necessary), X_1 and X_2 are sent along the two disjoint paths to the destination D.

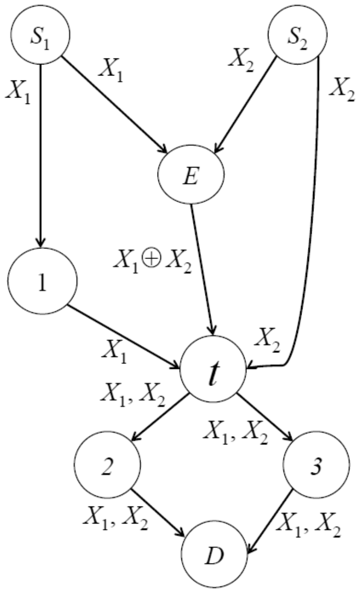


Figure 5.2: NC aware route selection technique with destination's node degree = 2.

Each node in the network is evaluated as t. The t which provides the minimum cost is selected as the desired intermediate destination, and corresponding routing is the desired solution.

5.3 Mathematical Model

5.3.1 Terminologies

The network is represented as an undirected graph $G(V, E)$, where V is the set of vertices (nodes) and E is the set of links. There are N nodes in $G(V, E)$. A link from node $i \in V$ to node $j \in V$ is denoted as $(i, j) \in E, i \neq j$. $x_{ij}^{sd,k}$ is the binary routing variable, where $k = \{1, 2\}$. If any link $(i, j) \in E$ belongs to the disjoint path number k between source node s and destination node d, then value of $x_{ij}^{sd,k}$

is 1, otherwise 0. c_{ij} is the cost of $(i, j) \in E$. z_{ij} is a binary variable used to select links $(i, j) \in E$ that are subject to NC. The network is bi-connected, i.e., it is ensured that in the network, for every possible source-destination pair, there exist two disjoint paths for applying 1+1 protection. Every node with degree ≥ 3 has the NC capability. Each node $t \in V$ is a candidate for intermediate destination node.

5.3.2 Presented mathematical model

The NC aware optimum routing solution for any $2SD$ scenario with destination's node degree ≥ 2 is described in the following. In the optimization problem, an integer linear programming (ILP) model is iteratively evaluated for all the nodes in the network.

The optimal routing problem is formulated by,

$$\min_{t \in V} f(t), \tag{5.1a}$$

where

$$\begin{aligned} f(t) &= \min \sum_{(i,j)\in E} \sum_{k=1}^{2} c_{ij} \times (x_{ij}^{s_1 t,k} + x_{ij}^{s_2 t,k} + 2 \times x_{ij}^{td,k}) \\ &\quad - \sum_{(i,j)\in E} (c_{ij} \times z_{ij}) \end{aligned} \tag{5.1b}$$

$$s.t. \quad x_{ij}^{s_1 t,1} + x_{ij}^{s_1 t,2} \leq 1, \ \forall (i,j) \in E \tag{5.1c}$$

$$x_{ij}^{s_2 t,1} + x_{ij}^{s_2 t,2} \leq 1, \ \forall (i,j) \in E \tag{5.1d}$$

$$x_{ij}^{s_1 t,1} + x_{ij}^{s_2 t,1} \leq 1, \ \forall (i,j) \in E \tag{5.1e}$$

$$x_{ij}^{s_1 t,1} + x_{ij}^{s_2 t,2} \leq 1, \ \forall (i,j) \in E \tag{5.1f}$$

$$x_{ij}^{s_1 t,2} + x_{ij}^{s_2 t,1} \leq 1, \ \forall (i,j) \in E \tag{5.1g}$$

$$x_{ij}^{td,1} + x_{ij}^{td,2} \leq 1, \ \forall (i,j) \in E \tag{5.1h}$$

$$\sum_{j\in V} (x_{ij}^{s_1 t,1} - x_{ji}^{s_1 t,1}) = 1, \ i = s_1 \tag{5.1i}$$

$$\sum_{j\in V} (x_{ij}^{s_1 t,2} - x_{ji}^{s_1 t,2}) = 1, \ i = s_1 \tag{5.1j}$$

$$\sum_{j\in V} (x_{ij}^{s_2 t,1} - x_{ji}^{s_2 t,1}) = 1, \ i = s_2 \tag{5.1k}$$

$$\sum_{j\in V}(x_{ij}^{s_2t,2} - x_{ji}^{s_2t,2}) = 1,\ i = s_2 \tag{5.1l}$$

$$\sum_{j\in V}(x_{ij}^{td,1} - x_{ji}^{td,1}) = 1,\ i = t \tag{5.1m}$$

$$\sum_{j\in V}(x_{ij}^{td,2} - x_{ji}^{td,2}) = 1,\ i = t \tag{5.1n}$$

$$\sum_{j\in V}(x_{ij}^{s_1t,1} - x_{ji}^{s_1t,1}) = 0,\ \forall i \in V, i \neq s_1, t \tag{5.1o}$$

$$\sum_{j\in V}(x_{ij}^{s_1t,2} - x_{ji}^{s_1t,2}) = 0,\ \forall i \in V, i \neq s_1, t \tag{5.1p}$$

$$\sum_{j\in V}(x_{ij}^{s_2t,1} - x_{ji}^{s_2t,1}) = 0,\ \forall i \in V, i \neq s_2, t \tag{5.1q}$$

$$\sum_{j\in V}(x_{ij}^{s_2t,2} - x_{ji}^{s_2t,2}) = 0,\ \forall i \in V, i \neq s_2, t \tag{5.1r}$$

$$\sum_{j\in V}(x_{ij}^{td,1} - x_{ji}^{td,1}) = 0,\ \forall i \in V, i \neq t, d \tag{5.1s}$$

$$\sum_{j\in V}(x_{ij}^{td,2} - x_{ji}^{td,2}) = 0,\ \forall i \in V, i \neq t, d \tag{5.1t}$$

$$x_{ij}^{sd,k} = \{0, 1\},\ \forall (i, j) \in E, s = s_1, s_2, t, k = 1, 2 \tag{5.1u}$$

$$z_{ij} = \{0, 1\},\ \forall (i, j) \in E \tag{5.1v}$$

$$z_{ij} \leq x_{ij}^{s_1t,2},\ \forall (i, j) \in E \tag{5.1w}$$

$$z_{ij} \leq x_{ij}^{s_2t,2},\ \forall (i, j) \in E \tag{5.1x}$$

$$z_{ij} \geq x_{ij}^{s_1t,2} + x_{ij}^{s_2t,2} - 1,\ \forall (i, j) \in E. \tag{5.1y}$$

Eq. (5.1a) selects the best $t \in V$ for which the 1+1 protection cost with NC is minimum. Eq. (5.1b) provides the optimum minimum cost of employing NC based 1+1 protection for each t in the network. Eqs. (5.1c)-(5.1h) specify the path disjoint constraints. Constraints (5.1c), (5.1d), and (5.1h) are the compulsory conditions that ensure 1+1 protection. The flow conservation constraints are specified by Eqs. (5.1i)-(5.1t). The flow conservation constraints at the source nodes are expressed by Eqs. (5.1i)-(5.1n), while Eqs. (5.1o)-(5.1t) are the flow conservation constraints at the intermediate nodes. The binary routing variable is described by Eq. (5.1u). Eqs. (5.1v)-(5.1y) describe the binary variable z_{ij}, which selects the links that are subject to NC.

5.4 Results

The performance measure is the amount of resource saving achieved by our presented model as compared to the same achieved by the conventional model. Resource saving is computed w.r.t. the cost of the regular 1+1 protection [1, 44] in which NC is not employed. We generated all possible combinations of $2SD$ scenarios in the Germany 17 and European 28 networks, as shown in Fig. 5.1, where common destination's node degree is two. Both the presented and the conventional models are evaluated in these generated scenarios, and the resource saving results are shown in Table 5.1. The commercial mathematical programming solver CPLEX®[46] is used to solve our presented model.

Table 5.1 shows that our presented model achieves 5.54% and 5.56% more resource saving respectively, in Germany 17 and European 28 networks, than that of the conventional model. Note that, this 5.5% resource saving is additional resource saving in these two evaluated network. This 5.5% resource saving add to the corresponding resource saving reported in Fig. 2.5 of chapter 2. The conventional model cannot provide an NC routing solution for 1+1 protection in a $2SD$ scenario with destination's node degree of two, and adopts the regular 1+1 protection as the solution. This is why no resource saving is achieved in the evaluated scenarios by the conventional model. Note that our presented model also includes the conventional model under the condition that $t = D$.

Table 5.1: Comparison of resource saving by presented and conventional momdels

	Germany 17	Europe 28
Number of nodes	17	28
Number of nodes with degree = 2	7	9
Resource saving by presented model	5.54%	5.56%
Resource saving by by conventional model	0%	0%

5.5 Summary

This chapter has formulated the problem of finding optimum network coding aware set of routes, in any MSCD scenario with two sources ($2SD$) where common destination's node degree is ≥ 2, as a mathematical model. In the presented model an ILP is evaluated iteratively for all the nodes in a network. The performance of the presented model w.r.t. a conventional model is thoroughly evaluated and analyzed in two different networks. Numerical results observed that our presented model achieves 5.5% resource saving than that of the conventional model in our examined networks.

Chapter 6

Mathematical model for coding aware instantaneous recovery with TS scenario

In chapter 6, we present a mathematical model for instantaneous recovery route design, by using the *TS scenario*, where traffic splitting and erasure correcting code are used. The presented mathematical model determines a set of $K + 1$ disjoint paths, K is the number of splitting, which minimizes the total cost of network resource utilization of traffic splitting based protection technique.

6.1 Introduction

Resilience is one of the key concerns in modern optical communication networks due to the huge concentration of capacity in optical fibres. Failure of one optical fibre may result in great data loss and massive service disruption to many customers. Thus, fast and efficient resilience techniques need to be provisioned to ensure service continuity in the event of network element failures. Moreover, it is desirable to keep the capital expenditures, i.e., the cost of network equipments as low as possible while choosing an appropriate resilience mechanism.

We consider the problem of protecting traffic demands between source-destination pairs against any single link failure. Our aim is to provide *instantaneous recovery* with 100% guarantee of data recovery upon occurrence of failure, while minimizing the total cost of network resource utilization. We are interested in the case of splitting the traffic demand into more than one sub-flow in order to achieve our stated goal.

In the conventional 1+1 protection two copies of the same data are sent on two disjoint paths, working and backup paths. Upon failure of the working path only the destination node switch over to the backup path which leads to the much desired instantaneous recovery feature, but double of the resources are needed.

The works in [20, 35] introduce two different models of network survivability against single link failure using network coding (NC). The technique presented in [35] uses the NC technique to reduce the total cost of network resource utilization for 1+1 protection in the MSCD scenario with two sources, while maintaining the 1+1 protection's instantaneous recovery feature. However, the encoding is achieved at the intermediate nodes, and, since traffic flows originate from different sources, flow synchronization [87] is necessary at the intermediate nodes before encoding is performed.

The work in [20] presents an NC-based instantaneous recovery technique against single link failure for single-source single-destination scenario. In [20] the traffic demand per communication round is split into two equal sub-flows, and sent along two disjoint paths, while the two sub-flows are encoded and sent simultaneously onto a third disjoint path which aids in the instantaneous recovery upon a failure.

Dividing the traffic demand between a source-destination pair into equal parts, and sending them onto disjoint paths reduce the bandwidth requirement on each utilized path. Therefore, the more we divide the traffic demand, the less bandwidth is required on each utilized path. However, increasing the number of disjoint paths incurs extra costs. There should be a tradeoff between the number of traffic splitting, K, and the total cost of all the disjoint paths. Note that no NC is employed when $K = 1$, which in this case is the conventional 1+1 protection technique. The study [20] considers the case of $K = 2$, and does not provide an optimum value of K that minimizes the total cost of network resource utilization.

Motivated by the issue mentioned above, this chapter presents an NC-aware instantaneous recovery technique that determines an optimal K and $K+1$ disjoint paths that minimize the total cost of network resource utilization.

6.2 Presented technique

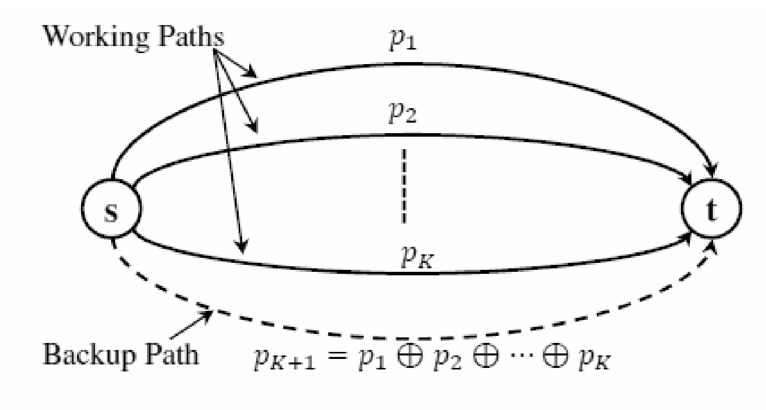


Figure 6.1: Network protection with optimal traffic splitting using NC.

The presented technique is illustrated in Figure 6.1. The traffic demand between the source s and destination t is split into K equal parts, and sent onto K disjoint paths. All the equally divided parts are network coded using the simple exclusive-OR (XOR) operation at the source node s, and sent simultaneously via a $K+1$th disjoint path to the destination.

The instantaneous recovery feature is achieved in the presented technique by the introduction of the $K+1$th disjoint path which carries the encoded data while K disjoint paths carry plain data. If a link fails, the destination node detects the failure, and forms the XOR of the encoded data on the $K+1$th path and the remaining plain data on $K-1$ working paths to instantaneously recover the lost data due to the failure.

The network $G(V, E)$ is represented as an undirected graph, where V is the set of vertices (nodes) and E is the set of directed edges (links). A link from node $i \in V$ to node $j \in V$ is denoted as $(i, j) \in E$. x_{ij}^k is the binary routing variable. If path k is routed through link $(i, j) \in E$, then $x_{ij}^k = 1$ or else $x_{ij}^k = 0$, and $M = \{1, 2, \cdots, K+1\}$. c_{ij} is the cost of using link $(i, j) \in E$ per unit of traffic demand, d_{st} is the traffic demand from source node s to destination node t, measured in bit/s. It is assumed that the network is at least bi-connected for each source-destination pair to facilitate the use of 1+1 path protection. The encoding is performed at the source, while the decoding is performed at the destination.

We present an integer linear programming (ILP) formulation to determine the minimum cost of the set of $K+1$ disjoint paths for each value of K. The value of K for which the total cost of the $K+1$ disjoint paths is minimum is the optimum traffic splitting number. Our objective function is as follows:

$$\min_{K \geq 1} \quad d_{st} \cdot \left[\frac{f(K+1)}{K} \right], \tag{6.1a}$$

where

$$f(K+1) = \min \sum_{k=1 \in M}^{K+1} \sum_{(i,j) \in E} c_{ij} x_{ij}^{k} \tag{6.1b}$$

subject to:

$$\sum_{j \in V} x_{ij}^{k} - \sum_{j \in V} x_{ji}^{k} = 1 \quad \text{if} \quad i = s, \forall k \in M \tag{6.1c}$$

$$\sum_{j \in V} x_{ij}^{k} - \sum_{j \in V} x_{ji}^{k} = 0 \quad \forall i \neq s, t \in V, \forall k \in M \tag{6.1d}$$

$$x_{ij}^{k} + x_{ij}^{k'} \leq 1 \quad \forall k, k'(k \neq k') \in M, \forall (i,j) \in E \tag{6.1e}$$

$$x_{ij}^{k} = \{0, 1\} \quad \forall k \in M, \forall (i,j) \in E \tag{6.1f}$$

The Equation (6.1a) is the objective function that selects the optimum value of K that minimizes the total cost of network resources, and the term $\left[\frac{f(K+1)}{K}\right]$ represents the cost of each individual path per-unit of traffic demand. Equation (6.1b) minimizes the total cost per-unit of traffic demand of the $K+1$ disjoint paths in the network. Equations (6.1c) and (6.1d) express flow conservation constraints at the source node and at intermediate nodes, respectively. The path disjoint constraint is shown in Equation (6.1e), which states that different paths do not share any common link. The binary routing variable is described by Equation (6.1f).

6.3 Results and discussion

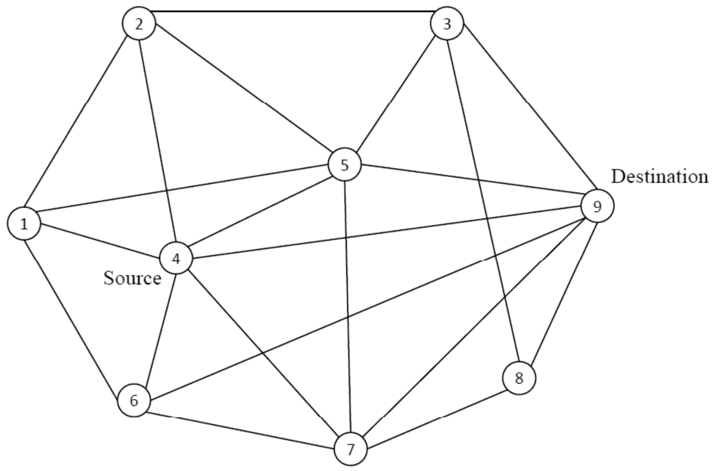


Figure 6.2: Network topology for evaluation.

In order to demonstrate the resource saving advantage of the presented technique, we select node 4 as the source, and node 9 as the destination in our examined network as shown in Figure 6.2. We assume a traffic demand ($d_{st} = 12$) between the selected source destination pair. The ILP problem is solved by using the open-source linear programming solver GLPK [88].

Figure 6.3 shows that the presented technique finds the optimum traffic splitting number $K = 3$ that minimizes the total cost of network resource utilization for the selected source destination pair. In Figure 6.3, the total cost of network resource utilization is normalized by that with $K = 1$. Note that, between the selected source destination pair at six link disjoint paths are possible. Thus maximum K for the selected pair is 5.

The presented technique achieves a network coding gain of 21% compared to the conventional 1+1 protection (for K=1) for the selected pair in our examined network. The result in Figure 6.3 also shows that, the total cost of network resource utilization first starts decreasing with the increasing of number of traffic splitting K, as it reaches the optimum traffic splitting value K, soon thereafter it begins to go upward again as the values of K go on increasing. Reason being that, with higher values of K, the disjoint paths between the source destination pair tend to be longer, resulting in higher cost of network resource usage. Thus, the higher the values of K result in higher total cost of network resource utilization.

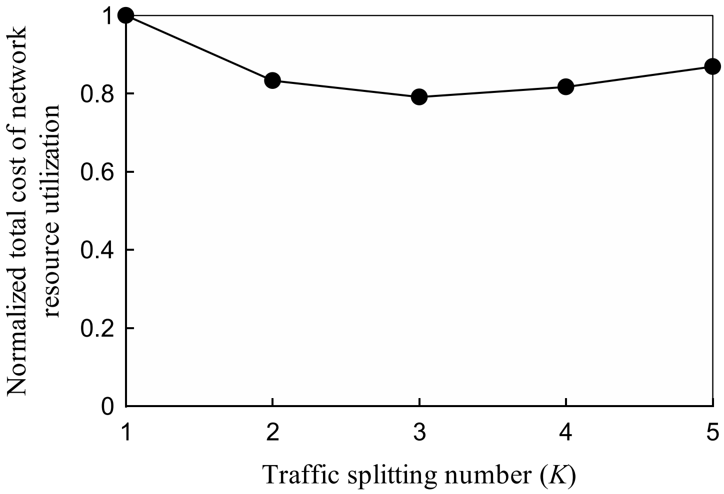


Figure 6.3: Total cost of network resource utilization for different values of K (for the selected source (node 4) and destination (node 9)) in our examined network.

6.4 Summary

This chapter presented an erasure correcting code aware instantaneous recovery technique with traffic splitting where a number of disjoint paths are employed between a source destination pair. In order to minimize the total cost of network resource utilization, a mathematical model is presented that determines an optimum value of traffic splitting $K, (K \geq 1)$ and $K+1$ disjoint paths. Numerical demonstration with a selected source destination pair has shown that the presented model effectively reduces the total cost of network resource utilization compared to the conventional 1+1 protection technique (for $K = 1$), while maintaining the much desired instantaneous recovery feature in the case of any single link failure.

Chapter 7

Differential Delay Aware Instantaneous Recovery Technique with TS scenario

This chapter presents a differential delay aware erasure correcting code based instantaneous recovery technique with traffic splitting for networks with the coding capability. In this technique traffic is split into K equal parts and forwarded independently through a set of K disjoint paths. P protection paths are employed in order to tackle any $t \geq 1$ failures on the $(K+P)$ disjoint paths, where $1 \leq t \leq P$. The protection paths carry encoded data, which are formed from the split parts by using an erasure correcting code at the source. A mathematical model is presented in order to determine $K+P$ disjoint paths. In order to recover from any t failure(s), all the remaining $(K+P-t)$ parts must be present at the destination. However, each disjoint path may experience a different delay, and the destination node has to buffer all the split parts, until the split part experiencing the longest delay reaches the destination. The amount of memory buffers depends on *the maximum allowable differential delay* Δ, the highest difference of delays of any two among the $(K+P)$ disjoint paths, supported at the destination. A large value of Δ increases the amount of required memory buffers and the service providers must consider Δ according to their budgets and service requirements. The effect of Δ on routing with traffic splitting is thoroughly investigated in various networks with respect to encoding/decoding costs, optical interface rates, buffering costs, the number of protection paths, the number of nodes in the network, and the number of minimum adjacent nodes. Simulation results observe that the number of required disjoint working paths (note that P is given), that carries the split data, decreases with the increase of encoding/decoding costs, optical interface rates, and buffering costs. It is also observed that the number of required disjoint working paths increases with the increase of the number of minimum adjacent nodes and Δ values.

7.1 Introduction

Nowadays the emerging high-performance applications require a huge amount of data transferred over the networks, in support of computation, storage, and visualization needs. These applications are typically pushing the bandwidth resources to the limits. Moreover, most of the high-performance applications are mission critical and require reliable end-to-end connections, where backup paths are needed for protection and restoration. Instantaneous recovery from failure by using minimum backup resources is a desirable feature that the service providers must offer to their customers needing reliable communication.

Multi-path provisioning can deliver many benefits and efficiency to network operators [51, 52]. However, with relatively high frequency of cable cuts, and tremendous amount of traffic loss due to a single failure, survivability in multi-path provisioning is of crucial concern [1, 68]. Protection, a proactive procedure in which spare capacity is reserved during connection establishment, allows recovery from such failures [69]. When a working path (i.e., a path carrying traffic during normal operation) fails, the connection is rerouted over to a backup path.

1+1 protection provides quick recovery from any single link failure in the network because only the destination node switching to the backup path is necessary after a failure is detected. In order to achieve this, the bandwidth resource requirements between every possible source and destination have to be doubled.

In order to minimize the resources required for protection from any single link failure, an erasure correcting code based instantaneous recovery technique with traffic splitting was addressed in literature [37]. In that addressed technique, traffic between a source destination pair is split into K equal parts, and sent along K disjoint paths independently. The $K + 1$th disjoint path is set to carry the encoded data formed from the K split parts.

In the protection technique presented in [37], in order to recover any lost split data, all the other split parts including the encoded data must be present at the destination altogether. Each disjoint path may experience a different propagation delay. For this reason extra memory buffers are required to store the split parts until all the parts arrive at the destination. In [37] the effect of propagation delay was not considered.

The destination node is equipped with either electrical or optical memory buffers. Electrical buffers are cost-efficient. Optical buffers, like optical delay lines, are expensive. The amount of memory buffers depends on the *maximum allowable differential delay* Δ, defined as the largest difference of delays of any two among the $(K+P)$ disjoint paths, supported at the destination, and must be taken into account along with the service requirements by the service providers to implement this technique. There must be a tradeoff between the allowed Δ and the costs incurred due to it.

This chapter presents a differential delay aware erasure correcting code based instantaneous recovery technique with traffic splitting for networks with the coding capability. In this technique traffic is split into K equal parts and forwarded independently through a set of K disjoint paths. This technique employs multiple protection paths, P, which carry encoded data formed from the split parts by using an erasure correcting code at the source, in order to tackle multiple failures. Each disjoint path experiences a different delay. The destination node has to buffer all the split parts, until it receives the split part experiencing the longest delay. The amount of memory buffers depends on Δ. We investigate the effect of Δ on routing with traffic splitting in various networks with respect to encoding/decoding costs, optical interface rates, buffering costs, the number of protection paths, the number of nodes in the network, and the number of minimum adjacent nodes. Simulation results observe that the number of required disjoint working paths (note that P is given), that carries the split data, decreases with the increase of encoding/decoding costs, optical interface rates, and buffering costs. It is also observed that the number of required disjoint working paths increases with the increase of the number of minimum adjacent nodes and Δ values.

The remainder of this chapter is as follows. Section 7.2 gives an overview of the related works and clarifies our contributions. The necessity of differential delay in the implementation of the presented instantaneous recovery technique is discussed in Section 7.3. The protection technique based on traffic splitting with multiple protection paths, in order to tackle multiple failures, is discussed in Section 7.4. Section 7.5 presents the terminologies and the mathematical model for determining an optimum set of disjoint routes supporting the assigned Δ. The effects of various parameters, mentioned earlier, on routing and the implementation issues of the presented technique are discussed in Section 7.6. Finally, Section 7.7 concludes the chapter.

7.2 Related works and contributions

7.2.1 Related works

The network coding (NC) technique [12] based protection for unicast and multicast traffic in optical networks were addressed in [15, 23, 24]. In the NC technique intermediate nodes are allowed to encode several incoming data [54].

A.E Kamal presented $1+N$ protection by using the p-cycle structure to handle single and multiple failures in optical mesh networks [15]. He also introduced $1+N$ protection technique that does not require a p-cycle [14]. Both the works [14, 15] presented integer linear programming (ILP) formulations in order to asses the cost of their respective protection techniques.

Manley *et al.* investigated the application of NC in the multicast protection in all-optical networks [23]. It also presented an ILP approach to find disjoint multicast protection tree. The design issues for NC in all-optical networks considering switching components, controllable optical delay lines, and all-optical XOR gates were also presented in [23].

Griffith *et al.* [24] presented an architecture employing 1+1 protection in optical burst switched (OBS) networks, and examined the design issues caused by propagation delays of the two disjoint paths across OBS network. Meusburger *et al.* suggested that rerouting of 1+1 optical channels, for long term planning of incremental networks, should be cost-efficient [57].

The simplest technique used for encoding data in recent optical networks, either at the intermediate nodes (NC technique [54]) or at the source node (erasure correcting codes) [58, 59], is the exclusive-OR (XOR) operation. The implementation of all-optical XOR devices have been addressed in several independent researches [23, 60, 61, 62, 63]. In these researches the operation speed of all-optical XOR gates were reported as 10.7 Gbps [60], 10 Gbps [61, 62], and 20 and 40 Gbps [63].

Traffic aggregation and buffering in optical networks were well addressed in literature [70, 71, 72, 73]. Yao *et al.* revealed that the use of electrical ingress buffering and traffic aggregation reduce packet-loss rate of optical packet-switched mesh networks [70]. Rhee *et al.* suggested that all-optical networking should not be a suitable option if cost and power consumption reduction is the main objective, but the use of passive-medium photonic switches with electronic memory buffers can introduce a substantial savings in power and cost [71]. All-optical signal processing and buffering were discussed in [72]. In order to ensure both efficient bandwidth utilization and scalability in all-optical wavelength-routed networks, a concept of traffic aggregation was addressed in [73].

The use of K disjoint path routing provisioning for more aggregate bandwidth has been studied in a number of different scenarios. Cidon *et al.* [74] highlighted the advantages of K disjoint path routing over single path routing considering the connection establishment time. [74] analyzed the performance of K disjoint path routing, without considering issues of path computation problems. Recently, differential delay problems have gained a lot of attention [75, 76, 77]. Ahuja *et al.* [75] studied the problem of minimizing the differential delay in the context of Ethernet over SONET (synchronous optical network). The algorithms presented in [75] select a path for a Virtually Concatenated Group (VCG) which has the minimum differential delay. Huang *et al.* [76] applied multi-path routing in SONET/SDH networks as a survivable routing approach with differential delay constraint. [76] presented heuristic algorithms based on K shortest path algorithms to decrease the blocking probability in SONET/SDH networks running on top of optical WDM. Srinivastava *et al.* [77] transformed the differential delay aware routing problem for ethernet over SONET/SDH as a flow problem. However, protection was not considered in [77]. Anderson *et al.* presented both exact and approximate algorithms for finding link disjoint path pairs satisfying wavelength continuity constraint in WDM networks [79].

1+1 protection techniques with erasure correcting code and NC were addressed in [20, 35], respectively. In [20] erasure correcting code based 1+1 protection technique was illustrated with the traffic split into two equal parts only, and it did not consider a suitable traffic splitting that minimizes cost. The work in [35] presented an NC based 1+1 protection technique that is applicable to scenarios where two source nodes have a common destination node, and this technique requires synchronization of flows from two different sources.

7.2.2 Our contributions

Our research work, for the first time, presents a differential delay aware erasure correcting code based instantaneous recovery technique with traffic splitting for networks with the coding capability. Our motivation to present this technique is to develop a recovery technique that achieves instantaneous recovery from failure by using less resources than that of conventional 1+1 protection. The presented technique allows the usage of multiple protection paths, which carry encoded data formed by using an erasure correcting code, in order to tackle arbitrary multiple failures among the employed disjoint paths. Traffic splitting reduces bandwidth requirements on the disjoint paths employed in the protection technique. The more we split, the more reduction in required bandwidth per disjoint path we get. However, as the traffic splitting number increases, the total length of the employed disjoint paths also increases. In addition, the differential delays and the required amount of memory buffers also increase. There must be a trade-off between the bandwidth reduction achieved per path due to traffic splitting, and the total length of the employed disjoint paths. We thoroughly investigate the performance of our presented technique. The results show trends on how routing, i.e., the total number of required disjoint working paths changes (the number of protection paths is given) with respect to various parameters mentioned earlier.

7.3 Differential delay aware instantaneous recovery technique with traffic splitting

[1]

7.3.1 Instantaneous recovery by erasure correcting code

The instantaneous recovery technique based on traffic splitting, presented in [37], is re-produced in Fig. 7.1 for the sake of discussion about differential delay. In Fig. 7.1, source node s and destination node d are the edge nodes in a network. Each of the edge nodes in the network is equipped with memory buffers for storing data. In the recovery technique the traffic between s and d is split into K equal parts $(p_1, p_2, ..., p_K)$ and sent through K disjoint paths simultaneously. The split parts of the traffic are encoded at the source by simple exclusive-OR (XOR) operation as $p_{K+1} = p_1 \oplus p_2 \oplus ..... \oplus p_K$, and sent simultaneously onto a $K+1$th disjoint path.

Note that the encoding technique here is the erasure correcting code [59] but not the NC technique. This is because the encoding is performed only at the source node, while NC is performed at the intermediate nodes [54].

Instantaneous recovery is achieved in this technique by the introduction of the encoded data in the $K+1$th path as illustrated in Fig. 7.2. If any link carrying plain data fails, the destination node can detect the failure. Only the destination node needs to switch to the $K+1$th backup path. By using the plain data on the $K-1$ working paths and the encoded data on the $K+1$th protection path, destination node instantaneously recovers the lost data sent along the failed path. For example, let the disjoint path carrying split part p_2 fails. Destination node d detects this, performs switching over to the $K+1$ path, which is a backup path, and recover lost data by simple XOR operation as $p_2 = p_1 \oplus p_3 \oplus ..... \oplus p_K \oplus p_{K+1}$.

[1] We discuss this section by considering $P = 1$ protection path.

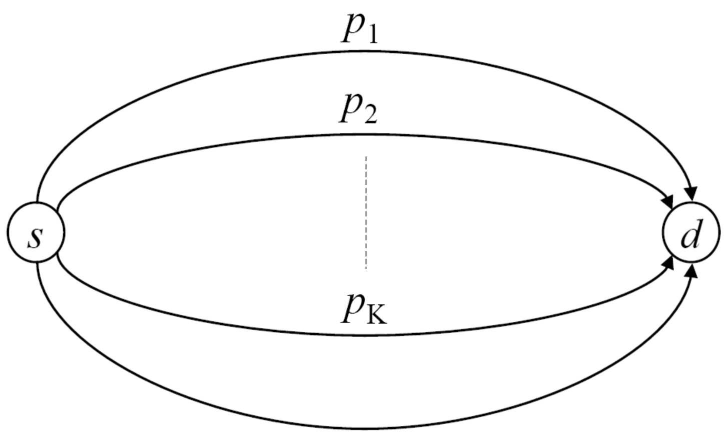


Figure 7.1: Network protection with traffic splitting using erasure correcting code at the source.

7.3.2 Differential Delay

The technique presented in [37] provides instantaneous recovery service from any single failure at a lower cost than the conventional 1+1 protection. However, as the traffic splitting number K increases, required disjoint paths tend to be longer in length. Therefore, each path requires a finite time to deliver the data sent through it from source node s to destination node d. We define this time as the propagation delay for each individual path. Each disjoint path has its own delay. The difference between any two among the employed $K + 1$ disjoint paths is defined as the differential delay.

7.3.3 Differential delay requires memory buffers

The destination node is responsible for reassembling the data stream as shown in Fig. 7.1. In order to recover the lost data p_2 in Fig. 7.2, all the other parts must reach at the destination node d before the recovery operation is performed. Since each path has a delay associated with it, the destination node has to buffer the parts arriving earlier up to the maximum differential delay among the $K + 1$ disjoint paths.

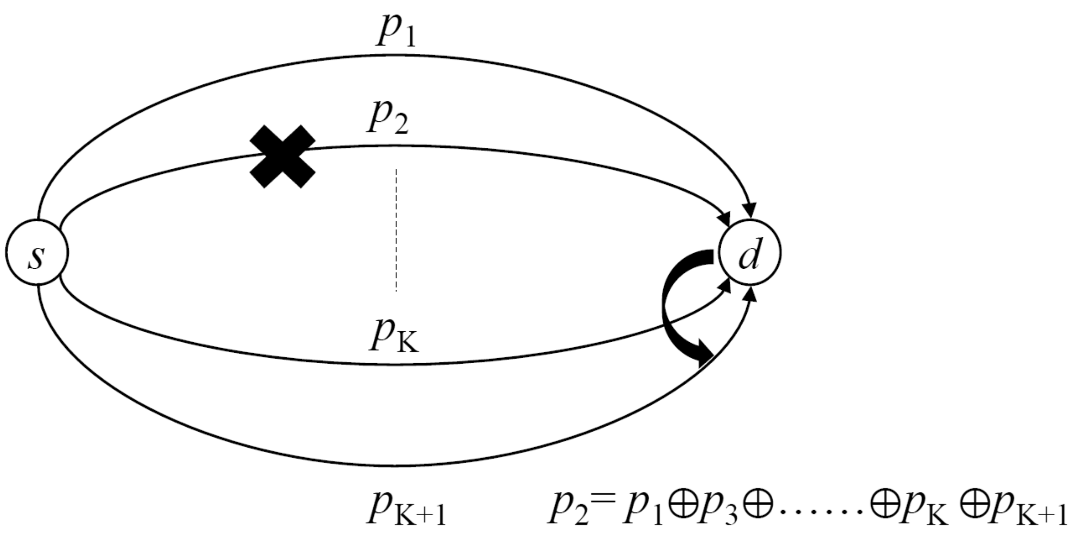


Figure 7.2: Recovery of failed data at the destination.

In order to buffer these split parts the destination node need to be equipped with additional memory buffers. It is obvious that the higher the differential delay among the $K+1$ disjoint paths, the higher is the amount of required memory buffers at the destination. Depending on the amount of memory at the destination node, there is an upper limit of the differential delay that can be supported per set of $K+1$ disjoint paths.

The differential delay is limited by the destination node's delay compensation capability. Network operations, administration, maintenance, provisioning, end-to-end quality of service (QoS) etc. services may be affected, if the differential delay is not properly taken into consideration in the routing [77].

7.3.4 Amount of required memory buffers

For example, assuming four paths P_1, P_2, P_3 and P_4 have an end-to-end propagation delays of 45, 47, 48, and 50ms. In order to reconstruct data at destination node, the destination node has to buffer the data from paths P_1, P_2, P_3 for 5, 3 and 2ms respectively until the data on P_4, with the largest differential delay, arrives. Thus, if each of the paths were transmitting data at 14Gb/s trunks, these three paths (P_1, P_2, and P_3) would require a total of 210 Mb memory buffers at the destination node, for it to be able to reconstruct the entire data. An increased line rate also increases this memory requirement.

The highest amount of memory buffers required for any disjoint path at the destination can be computed as $\mathcal{B} = R \times \Delta$, where $\mathcal{B}$ is the amount of memory buffers, Δ is the maximum allowable differential delay, and R is the data transmission rate. Thus, if we have $K + 1$ disjoint paths, the total amount of memory buffers required at the destination is computed as $\mathcal{B} = K \times R \times \Delta$.

7.4 Traffic splitting with multiple backup path for protection against multiple failure

In the presented erasure correcting code based instantaneous recovery technique with traffic splitting, multiple disjoint protection paths are introduced in order to tackle multiple link failures. In [17], [80] a technique with reduced capacity utilization was addressed to recover several disjoint connections (among different source destination pairs) from multiple failures. However, in this research we consider protecting multiple failures among several disjoint paths belonging to the same source destination pair.

Let us consider that K working paths carry plain data, and P protection paths carry encoded data. The protection paths are provided to tackle $t \geq 1$ failures. Any arbitrary t paths out of $(K + P)$ paths may fail.

The source node sends plain data along K working paths, and encoded data along P protection paths. The encoded data are generated by using a generator matrix $\mathbf{Z}$, known to both source and destination, which is illustrated in Fig. 7.3.

The generator matrix $\mathbf{Z}$ of Fig. 7.3 consists of an identity submatrix $\mathbf{I}_{K\times K}$ corresponding to the working paths, and a submatrix $\mathbf{E}_{K\times P}$ corresponding to the protection paths. $\mathbf{Z}$ has a rank of K. Each column of $\mathbf{I}_{K\times K}$ indicates that the working paths carry plain data only. If e_{ij} is 1, plain data on working path i is included in protection path j. Otherwise e_{ij} is zero. All the encoding (and decoding) operations are performed over the binary finite field $\mathbf{F}_2$. Each row of $\mathbf{Z}$ represents a codeword. The weight of a codeword in $\mathbf{Z}$ is the number of nonzero

$$\boldsymbol{Z} = \left[\begin{array}{ccccc|cccc} 1 & 0 & . & . & 0 & e_{11} & e_{12} & \cdot & e_{1p} \\ 0 & 1 & . & . & 0 & e_{21} & e_{22} & \cdot & e_{2p} \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ 0 & 0 & . & . & 1 & e_{k1} & e_{k2} & \cdot & e_{kp} \end{array}\right]$$

Identity matrix $I_{K\times K}$ Sub-matrix $E_{K\times P}$

Figure 7.3: Generator matrix.

elements. The minimum weight d_{min} of $\mathbf{Z}$ is denoted by the minimum weight over all rows in $\mathbf{Z}$ [17].

Among the $(K+P)$ disjoint paths, at most t arbitrary path(s) may fail. Data of K working paths are recovered if the following conditions are satisfied,

- $t \leq d_{min} - 1$, and
- The rank of the received matrix, which has a dimension of $K(K + P - t)$, is K.

In order to provide protection from any single ($t = 1$) failure, d_{min} should be equal to 2. For this purpose only one protection path ($P = 1$) is sufficient. In order to tackle $t \geq 2$ failures, d_{min} should be ≥ 3, and for this purpose at least $P \geq d_{min} + 1$ protection paths must be employed.

In order to satisfy the above recovery conditions we must ensure that, first, all the K plain data must be present in the $(K + P - t)$ received combinations, and, second, encoded data on each of the protection path must be different. The well known Hamming codes, or narrow-sense BCH (Bose-Chaudhuri) codes can be used to create a proper generator matrix that satisfies the above two recovery conditions [17], [81], [82].

Let us consider the scenario, shown in Fig. 7.4, where seven disjoint paths are available between source node s and destination node d. Let $K = 4$ be the traffic splitting number. Three paths are available to use as protection paths.

The generator matrix, known to both s and d, for this situation is given in the following,

$$\mathbf{Z} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

This $\mathbf{Z}$ has $d_{min} = 3$. Thus it can recover from any two failures occurring in any of the seven disjoint paths. According to $\mathbf{Z}$, source node s divides its data into four equal parts x_1, x_2, x_3 and x_4, and sends them through four disjoint paths. The three protection paths have different data combinations. Following $\mathbf{Z}$, the three encoded data are created as $x_1 \oplus x_2 \oplus x_3$, $x_2 \oplus x_3 \oplus x_4$, and $x_1 \oplus x_3 \oplus x_4$, and sent through the three available backup paths.

Suppose failures happened on the two paths carrying x_1 and x_3. Then destination receives five different data combinations, x_2, x_4, $x_1 \oplus x_2 \oplus x_3$, $x_2 \oplus x_3 \oplus x_4$, and $x_1 \oplus x_3 \oplus x_4$, on the five active disjoint paths. The destination node can recover the lost two split data parts, x_1 and x_3, from these five received parts as follows: it recovers x_3 by performing two XOR operations as $x_3 = x_2 \oplus x_4 \oplus (x_2 \oplus x_3 \oplus x_4)$. In a similar way x_1 can be recovered as $x_1 = x_3 \oplus x_4 \oplus (x_1 \oplus x_3 \oplus x_4)$.

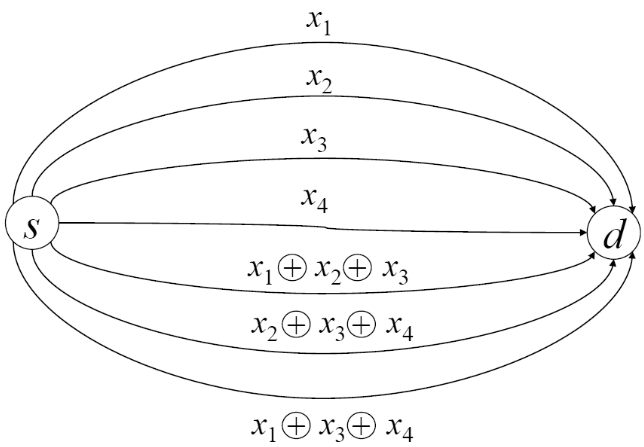


Figure 7.4: K is 4. We employ $P = 3$ backup paths for protection against any $t = 2$ failures occurring on any of the $4 + 3 = 7$ disjoint paths.

7.5 Mathematical Preliminaries and Formulation

7.5.1 Terminologies

The network $G(V, E)$ is represented as an undirected graph, V is the set of vertices (nodes) and E is the set of directed edges (links). A link from node $i \in V$ to node $j \in V$ is denoted as $(i, j) \in E$. x_{ij}^k is the binary routing variable. If (i, j) belongs to disjoint path number $k \in M$, then x_{ij}^k is 1, otherwise 0, where $M = \{1, 2, \cdots, K, K + 1, \cdots, K + P\}, P \geq 1$. Here P implies the number of protection path(s) used. c_{ij} is the cost of using (i, j) by unit bandwidth. The link cost is a given parameter, which is set by the network service providers. D_{sd} is the traffic demand from source s to destination d. It is assumed that the network is at least two-connected to facilitate the use of 1+1 path protection. Encoding is performed at the source, while decoding is performed at the destination upon t failure(s). The split parts are encoded and decoded by using XOR operations. The split parts are encoded and decoded by using XOR operations.

An optical interface is of fixed rate. Let B_I be the rate of an optical interface in bit/second. In order to set up paths in optical networks we assume conventional rules for wavelength continuity[1].

For $t = 1$ failure recovery, $K - 1$ bitwise XOR operations are required for encoding on the only protection path ($P = 1$). For $t \geq 2$ failures recovery with $K \geq P$, at most $K - 2$ bitwise XOR operations are required for encoding on each of the $P \geq 2$ protection paths. Let Ψ be the cost of one XOR operation. Let $\mathcal{X}$[2] be the total number of required XOR operations over all the protection paths,

[1]The objective of the presented technique is to determine a set of disjoint routes that minimizes the cost of protection of bandwidth request between any source destination pair in networks with the coding capability. We avoid specially optical network specific constraint wavelength continuity, as we consider a general problem.

[2]We provide an expression for the number of encoding operations that meets the recovery conditions.

which is defined by,

$$\mathcal{X} = \begin{cases} (K-1)\times P, & \text{if } K \geq 1, P = 1 \\ (K-2)\times P, & \text{if } K \geq P \geq 2 \\ (K-1) + (K-2)\times(P-1), & \text{if } K \leq P, P \geq 2 \end{cases} \tag{7.1}$$

At the destination, the data processing technique may be different than that of the source. However, for simplicity we assume that the same number of XOR operations are required for recovery operation at the destination.

A traffic splitting number, for which the total cost of provisioned $(K + P)$ disjoint paths, are minimum is defined as the suitable traffic splitting number, K_S. The K_S indicates the number of required disjoint working paths. Therefore, a change in K_S also causes a change in the routing. Let $\mathcal{K}$ be the minimum node degree between a source destination pair. The range of K_S for or any source destination pair is given by $1 \leq K_S \leq \mathcal{K} - P$.

The path delay associated with a disjoint path is defined as follows. Let δ_{ij} be the delay of an edge $(i, j) \in E$, then the path delay $\delta_{\mathcal{P}}$ of a path $\mathcal{P}$ is defined as:

$$\delta_{\mathcal{P}} = \sum_{(i,j)\in\mathcal{P}} \delta_{ij}.$$

The differential delay between two paths $\mathcal{P}$ and $\mathcal{P}'$ can be defined as follows:

$$|\delta_{\mathcal{P}} - \delta_{\mathcal{P}'}| \leq \Delta,$$

where Δ is the maximum allowable differential delay supported by the destination node.

The maximum allowable differential delay Δ is a given parameter. We introduce a decision variable y in our formulation. y represents the longest propagation delay among the disjoint paths, and facilitates the determination of differential delay for each path. Let C_B be the buffer cost per unit delay for each path $k \in M$.

If multiple services are aggregated, the same optical channel can be shared to reduce the cost. In this case the most strict differential delay constraint determined by service requirements should be considered.

Table 7.1 below lists the symbols that are frequently used in this chapter.

Table 7.1: List of symbols and descriptions

Symbol	Description
K	Traffic splitting number
P	Number of protection paths
K_S	The suitable splitting number
B_I	Optical interface rate
t	number of failure(s)
Δ	Maximum allowable differential delay
Ψ	Encoding/Decoding costs
C_B	Buffering costs
N	The total number of nodes
m	Number of minimum adjacent nodes
D_{sd}	Traffic demand between source s and destination d
c_{ij}	Cost of a link (i, j)
y	A variable related to path delay
δ_{ij}	Delay of a link (i, j)
$\mathcal{K}$	the minimum node degree between the source and destination
$\delta_{\mathcal{P}}$	Delay of a path
x_{ij}^k	Routing variable
y	the longest propagation delay among the disjoint paths
$\mathcal{X}$	Required number of XOR operations

7.5.2 Mathematical formulation

We present two mathematical models, which are a traffic volume charging model and an interface-oriented charging model. In the traffic volume charging model, the total cost is affected by the actual total bandwidth usage. In the interface-oriented charging model, it is affected by the number of optical interfaces used.

7.5.2.1 Traffic volume charging model

For the traffic volume charging model, we present an optimization problem by formulating an integer linear programming (ILP) one. The optimization problem determines determines the best set of $K_S + P$ disjoint paths that meets the maximum allowable differential delay (Δ) supported at the destination.

The optimization problem is expressed as follows:

$$\min_{K\geq 1} \quad \frac{D_{sd}}{K} \times f(K+P) + 2 \times \Psi \times \mathfrak{X} \tag{7.2a}$$

where

$$f(K+P) = \min \sum_{k\in M}^{K+P} \sum_{(i,j)\in E} c_{ij}x_{ij}^k + C_B \times \sum_{k\in M}^{K+P} \left(y - \sum_{(i,j)\in E} \delta_{ij} x_{ij}^k \right) \tag{7.2b}$$

subject to:

$$\sum_{j\in V} x_{ij}^k - \sum_{j\in V} x_{ji}^k = 1 \quad \text{if} \quad i = s, \forall k \in M \tag{7.2c}$$

$$\sum_{j\in V} x_{ij}^k - \sum_{j\in V} x_{ji}^k = 0 \quad \forall i \neq s, t \in V, \forall k \in M \tag{7.2d}$$

$$\sum_{k\in M} x_{ij}^k \leq 1 \quad \forall (i,j) \in E \tag{7.2e}$$

$$\sum_{(i,j)\in E} \delta_{ij} x_{ij}^k - \sum_{(i,j)\in E} \delta_{ij} x_{ij}^{k'} \leq \Delta \quad \forall k, k'(k \neq k') \in M \tag{7.2f}$$

$$\sum_{(i,j)\in E} \delta_{ij} x_{ij}^k \leq y \quad \forall k \in M \tag{7.2g}$$

$$x_{ij}^k = \{0, 1\} \quad \forall k \in M, \forall (i,j) \in E. \tag{7.2h}$$

The objective function in Eq. (7.2a) chooses a suitable value of $K = K_S$ for the set of $(K + P)$ disjoint paths that meets the maximum end-to-end delay required at the destination node. The first term of Eq. (7.2a) determines the total bandwidth usage cost, and the second term provides the cost for encoding/decoding. The second part is multiplied by 2, because we assume that the same number of XOR operations are required at the source and destination. Eq. (7.2b) is the objective function of the ILP problem. The first part of Eq. (7.2b) indicates the total cost of sending each unit of traffic demand through the set of $K + P$ disjoint paths in the network. The second part of Eq. (7.2b) indicates the total buffering cost at the destination by computing the buffering time needed for the data on each of the disjoint paths. Eqs. (7.2c) and (7.2d) express the flow conservation constraints at the source node and at intermediate nodes, respectively. Eq. (7.2f) guarantees that the differential delay between any two out of $K+P$ paths in the solution is not greater than Δ supported at the destination node. y ($\leq \Delta$) is a positive nonzero decision variable as defined by Eq. (7.2g). Note that y is the longest propagation delay among the $(K+P)$ paths. The path disjoint constraint is given by Eq. (7.2e), which states that different paths do not share any common link. The binary routing variable is described by Eq. (7.2h).

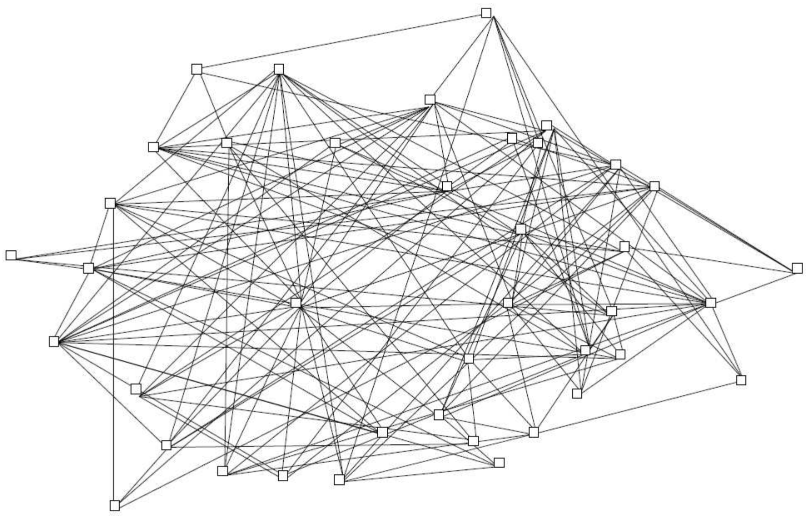


Figure 7.5: Examined network topology

7.5.2.2 Interface-oriented charging model

In the interface-oriented charging model, we redefine the objective function in Eq. (7.2a) by incorporating optical interface rate B_I as follows:

$$\min_{K\geq 1} \left\lceil \frac{D_{sd}}{B_I \times K} \right\rceil B_I \times f(K+P) + 2 \times \Psi \times \mathfrak{X}. \tag{7.3}$$

In Eq. (7.3) $\left\lceil \frac{D_{sd}}{B_I \times K} \right\rceil$ determines the number of optical interfaces required per disjoint path. Note that $\lceil x \rceil$ is the smallest integer greater than or equal to x. The term $B_I \times f(K+P)$ determines the total path cost considering each path uses a interface of rate B_I. The second part of Eq. (7.3) provides the encoding/decoding cost. The constraints in the interface-oriented charging model are the same as those in the traffic volume charging model.

In the presented optimization problem, Eqs. (7.2a) and (7.3) take into account key factors including bandwidth usage, encoding, buffering, and decoding costs, that contribute to the implementation cost of the presented technique.

7.6 Performance Evaluation and Discussion

For the presented differential delay aware instantaneous recovery technique with traffic splitting, we investigate how the routing, i.e., the suitable number of required working paths (K_S) (since P is a given parameter) changes with varying maximum allowable differential delay Δ, encoding/decoding costs Ψ, buffering costs C_B, optical interface rates B_I, the number of protection paths P, the total number of nodes N, and the number of minimum adjacent nodes m. The presented technique is extensively evaluated in several random networks. In our evaluation we always use the traffic volume charging model, unless otherwise specified. In order to solve the presented ILP problem, a commercial linear programming solver, CPLEX® [46], is used.

Note that the number of possible traffic splitting depends on the smaller node degree between source and destination nodes. That is, if the source node has connection to six neighbor nodes and the destination node has connection to four nodes, then the possible splitting number in the presented technique, considering one protection path, is equal to at most three.

The random network topologies for evaluation are generated based on the Waxman's probability model by using the BRITE internet topology generator [49]. The Waxmans probability model for interconnecting the nodes of the topology is given by:

$$P(u,v) = \alpha e^{-d_e/(\beta L)},$$

where $0 < \alpha, \beta \leq 1$, d_e is the Euclidean distance from node u to node v, and L is the maximum distance between any two nodes.

7.6.1 Effects of Δ, Ψ, C_B, B_I, and P on routing

The random network topology, as shown in Fig. 7.5, is used to evaluate the effects of various values of Δ, Ψ, C_B, B_I, and P on $K_S + P$. A source destination pair, between which the highest number of disjoint paths exists, is selected for evaluation. The link costs are considered uniform. The delay for each link is set proportional to the corresponding link's length.

Table 7.2: Normalized total cost for $(K + P)$ disjoint paths (w.r.t. the cost of K=1) for various Δ values in traffic volume charging model.

K	$\Delta \geq 6$	$\Delta = 5$	$\Delta = 4$	$\Delta = 3$
1	1	1	1	1
2	0.75	0.75	0.75	0.75
3	0.67	0.67	0.67	0.67
4	0.63	0.63	0.63	**0.63**
5	0.6	0.6	0.6	0.65
6	**0.58**	**0.58**	**0.58**	
7	0.61	0.61	0.61	
8	0.63	0.63	0.63	
9	0.64	0.64	0.64	
10	0.65	0.65	0.65	
11	0.66	0.66	0.68	
12	0.67	0.67	0.77	
13	0.67	0.69		
14	0.68	0.71		
15	0.7	0.73		

$^{\perp}$ For $\Delta \leq 2$ there is no feasible solution.

7.6.1.1 Δ versus K_S

The effect of various values of Δ on the total cost for $K_S + P$ paths is presented in Table 7.2, which illustrates that as Δ value decreases, K_S also decreases. If Δ has a low value ($\Delta \leq 2$ in our observations) no solution is possible, because the minimum differential delay between any two disjoint paths is higher than that of the specified Δ.

Here we assume that Δ is a variable parameter, which is set by the service providers, and the cost of encoding/decoding Ψ is set to 1. We also assume that only one protection path ($P = 1$) is employed in the model. The buffering cost C_B is set to 0.1.

For a particular value of Δ the traffic splitting number K, for which the normalized total cost is minimum, is selected as K_S. In the table for each Δ value the total path cost, for any K, is normalized with respect to the cost corresponding to $K = 1$. Note that $K = 1$ means no splitting, and thus the cost corresponds to the conventional 1+1 protection technique. Throughout this chapter the total path cost in any circumstance is normalized with the related total path cost with $K = 1$.

In Tables 7.1, 7.2, 7.3, 7.4, and 7.5, an empty entry for any particular K value implies that there is no feasible solution for that particular traffic splitting number. A bold entry indicates the minimum value in each column. The K value corresponding to a bold entry indicates the suitable traffic splitting number, K_S.

7.6.1.2 Ψ versus K_S

The effect of various Ψ values on the total cost for $K_S + P$ paths is presented in Table 7.3, which illustrates that the K_S decreases with the increase of the encoding/decoding costs Ψ. The data presented in Table 7.3 is interpreted in the same way as described in the previous subsection.

Here we assume that Δ, C_B, and P are set to 10, 0.1, and 1, respectively. For simplicity, we assume that the same number of encoding and decoding operations are required.

Table 7.3: Normalized total cost for $(K+P)$ paths (w.r.t. the cost of K=1) for various Ψ values in traffic volume charging model.

K	$\Psi = 1$	$\Psi = 5$	$\Psi = 8$	$\Psi = 11$	$\Psi = 17$	$\Psi = 35$	$\Psi = 101$
1	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**
2	0.75	0.76	0.77	0.78	0.79	**0.81**	1.02
3	0.67	0.69	0.71	0.72	**0.73**	0.84	1.17
4	0.63	0.67	0.69	**0.70**	0.75	0.89	1.38
5	0.61	0.66	**0.67**	0.71	0.77	0.95	1.61
6	**0.60**	**0.65**	0.68	0.72	0.80	1.02	1.85
7	0.62	0.68	0.73	0.77	0.86	1.13	2.12
8	0.64	0.71	0.77	0.82	0.92	1.24	2.39
9	0.66	0.74	0.80	0.86	0.98	1.34	2.66
10	0.67	0.76	0.83	0.90	1.03	1.44	2.92
11	0.68	0.78	0.86	0.93	1.08	1.53	3.18
12	0.69	0.80	0.89	0.97	1.13	1.63	3.44
13	0.70	0.82	0.91	1.00	1.18	1.72	3.70
14	0.71	0.84	0.94	1.04	1.23	1.82	3.96
15	0.74	0.88	0.98	1.09	1.30	1.93	4.24

7.6.1.3 Dependency of K_S on multiple protection paths ($P \geq 1$)

The effect of the number of protection paths, P, on the total cost for $K_S + P$ paths is presented in Table 7.4, which illustrates that for $P = 1, 2, 3, 4, 5, 6, 7$ and 8, K_Ss are $6, 5, 10, 10, 11, 10, 9$, and 10, respectively. Thus as P increases, the value of $K_S + P$ also increases.

It is assumed that Δ is set to 10 and the cost of encoding/decoding Ψ is set to 1. We also assume that up to eight protection paths are allowed, i.e., $P = 1, 2, 3, 4, 5, 6, 7, 8$.

7.6.1.4 Effects of buffering cost C_B on K_S

Table 7.5 illustrates that as buffering cost increases, K_S decreases. We assume that Δ is set to 10, buffering cost C_B is set to 0.1, and the cost of encoding/decoding Ψ is set to 1. We also assume that only one protection path ($P = 1$)

Table 7.4: Number of protection paths, P, versus K in traffic volume charging model.

	Normalized cost for the number of protection paths, P							
K	P=1	P=2	P=3	P=4	P=5	P=6	P=7	P=8
1	1	1	1	1	1	1	1	1
2	0.751	0.669	0.628	0.610	0.587	0.607	0.594	0.572
3	0.671	0.561	0.511	0.468	0.475	0.481	0.453	0.432
4	0.632	0.513	0.449	0.434	0.423	0.413	0.385	0.367
5	0.617	**0.481**	0.437	0.413	0.388	0.374	0.349	0.323
6	**0.603**	0.487	0.432	0.394	0.366	0.353	0.320	0.298
7	0.628	0.496	0.424	0.382	0.355	0.333	0.303	0.281
8	0.652	0.497	0.421	0.379	0.342	0.322	0.292	**0.270**
9	0.664	0.500	0.423	0.371	0.336	0.316	**0.286**	
10	0.677	0.510	**0.420**	**0.369**	0.333	**0.312**		
11	0.697	0.511	0.421	0.370	**0.333**			
12	0.703	0.517	0.425	0.372				
13	0.717	0.525	0.432					
14	0.733	0.537						
15	0.753							

is employed in the model.

7.6.1.5 Dependency of K_S on $\frac{D_{sd}}{B_I}$

We examine the interface-oriented charging model. The effect of various $\frac{D_{sd}}{B_I}$ values on the total cost for K_S+P paths is presented in Table 7.6 and the number of required interfaces for each $\frac{D_{sd}}{B_I}$ and K are presented in Table 7.7. Table 7.6 illustrates that a low value of $\frac{D_{sd}}{B_I}$ suppress traffic splitting. For example, let us assume that D_{sd}= 5 Gbps and B_I=2.5 Gbps, and thus $\frac{D_{sd}}{B_I}$=2. According to Table 7.6 and Table 7.7, K_S is 2, and three optical interfaces (two for split traffic and one for encoded traffic) are required. In this situation, an increase in the traffic splitting number causes the split traffic volume to become less than the optical interface rate, and leads to an inefficient use of the optical interfaces. This

Table 7.5: Buffering cost C_B at destination versus K in traffic volume charging model.

	Normalized cost of $(K+P)$ paths			
K	C_B=0.1	C_B=0.3	C_B=0.4	C_B=0.5
1	1.000	1.000	1.000	1.000
2	0.751	0.754	0.756	0.757
3	0.671	0.678	0.683	0.686
4	0.632	0.646	0.666	**0.660**
5	0.617	0.650	**0.659**	0.682
6	**0.603**	**0.642**	0.662	0.680
7	0.628	0.668	0.690	0.709
8	0.652	0.707	0.733	0.751
9	0.664	0.714	0.740	0.764
10	0.677	0.730	0.758	0.778
11	0.697	0.750	0.773	0.795
12	0.703	0.777	0.815	0.849
13	0.717	0.792	0.826	0.858
14	0.733	0.807	0.845	0.872
15	0.753	0.825	0.862	0.888

increases the total required number of interfaces, and the value of the objective function defined by Eq. (7.3) in the interface-oriented charging model.

We consider various traffic demand values between the selected source-destination pair. The optical interface traffic rate, B_I, is a given parameter, which can be set to any of the 2.5/10/40/100 Gbps rates, and these rates conform to currently available devices. It is assumed that Δ, C_B, Ψ, and P are set to 10, 0.1, 1, and 1, respectively.

7.6.2 Topology dependency of K_S

We evaluate the traffic volume charging model in several random network topologies to investigate the effects of two topological parameters, the number of nodes in the network, N, and the number of minimum adjacency nodes, m, on K_S.

Table 7.6: Normalized total cost for $(K+P)$ paths (w.r.t. the cost of K=1) for various $\frac{d_{st}}{B_I}$ values in interface-oriented charging model.

K	$\frac{d_{sd}}{B_I}$=80	$\frac{d_{sd}}{B_I}$=20	$\frac{d_{sd}}{B_I}$=5	$\frac{d_{sd}}{B_I}=2$
1	1.00	1.00	1.00	1.00
2	0.75	0.75	0.90	**0.75**
3	0.68	0.70	0.80	1.00
4	0.63	0.63	1.00	1.25
5	**0.60**	**0.60**	**0.60**	1.50
6	0.61	0.70	0.70	1.75
7	0.64	0.64	0.85	2.13
8	0.63	0.75	1.00	2.50
9	0.65	0.86	1.15	2.88
10	0.65	0.65	1.30	3.25
11	0.73	0.73	1.45	3.63
12	0.70	0.80	1.60	4.00
13	0.77	0.88	1.75	4.38
14	0.71	0.95	1.90	4.75
15	0.79	1.05	2.10	5.25

We generate 10 random network topologies for every m and N combination. We determine the maximum K_S in each of the generated random network by selecting source and destination pairs with the highest number of disjoint paths. We obtain the same maximum K_S for each of the 10 random topologies generated with the same m and N but different random seed values.

7.6.2.1 Effects of number of nodes, N, on K_S

The dependency of K_S, with a given Δ, with respect to the number of nodes in the network is presented in Fig. 7.6. In this case the number of minimum adjacent nodes is kept fixed at $m = 2$.

With sufficient allowed $\Delta = 10$, K_S increases with the number of nodes N in the network, but this increase is marginal. One point to note, when $\Delta = 1$, no splitting, i.e. the conventional 1+1 protection is the optimal solution as in the

Table 7.7: Number of required optical interfaces at different $\frac{d_{st}}{B_I}$ ratios versus K in interface-oriented charging model.

K	$\frac{d_{sd}}{B_I}$=80	$\frac{d_{sd}}{B_I}$=20	$\frac{d_{sd}}{B_I}$=5	$\frac{d_{sd}}{B_I}=2$
1	160	40	10	4
2	120	30	9	3
3	108	28	8	4
4	100	25	10	5
5	96	24	6	6
6	98	28	7	7
7	96	24	8	8
8	90	27	9	9
9	90	30	10	10
10	88	22	11	11
11	96	24	12	12
12	91	26	13	13
13	98	28	14	14
14	90	30	15	15
15	96	32	16	16

20, 30, and 40 nodes networks in Fig 7.6.

7.6.2.2 Effect of number of minimum adjacent nodes, m

The dependency of K_S, with a given Δ, with respect to the number of minimum adjacent nodes m in the network is presented in Fig. 7.7. In this case the number of nodes in the network is kept fixed at $N = 40$.

Figure 7.7 shows that K_S increases with m in the network. Because more number of adjacent nodes per node increases the opportunity for diverse disjoint multi-path routing.

7.6.3 Operation of presented technique

We explain the path setup operation, encoding, synchronization, and merging (decoding) operations of the presented technique.

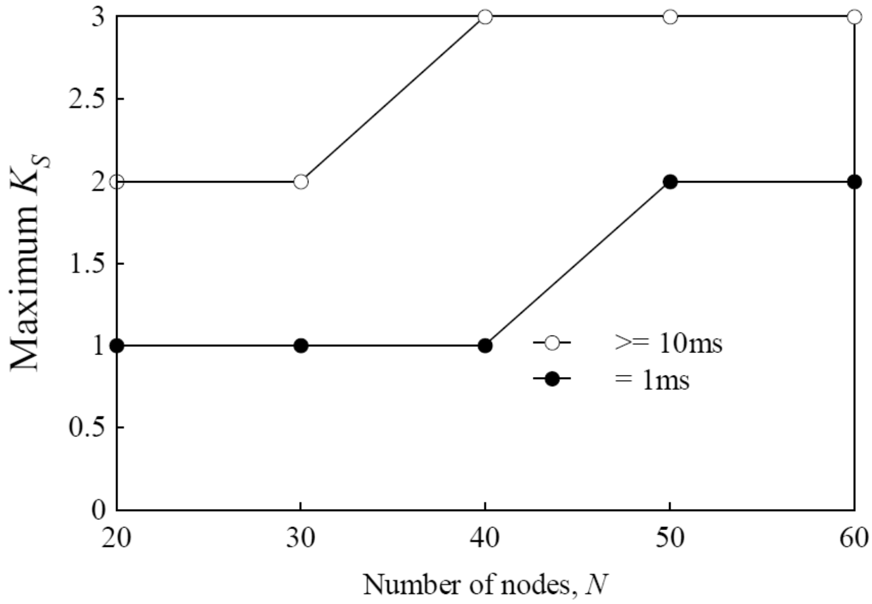


Figure 7.6: Dependency of K_S for different Δ values w.r.t. varying number of nodes N in the network.

7.6.3.1 Path setup

Figure 7.8 illustrates a path flow operation of the presented technique, where we refer to a path computation element (PCE)-based network control/management model [83, 84, 85]. A network management system (NMS) receives a request from the source node. The request includes the requested bandwidth and number of desired protection paths for a particular destination. The NMS sends the request to the PCE. The PCE determines an optimum set of disjoint paths, by computing the presented optimization problem, that minimizes the cost of protection of traffic between the intended source and destination pair. The PCE replies to the NMS. The reply includes all the path information, delay parameters, and the suitable traffic splitting number. The NMS has all the information to set up the paths.

There are two approaches to set up the paths. One is the centralized approach, and the other is the signalling approach. In the centralized approach the NMS only sends the reply message including the suitable traffic splitting number

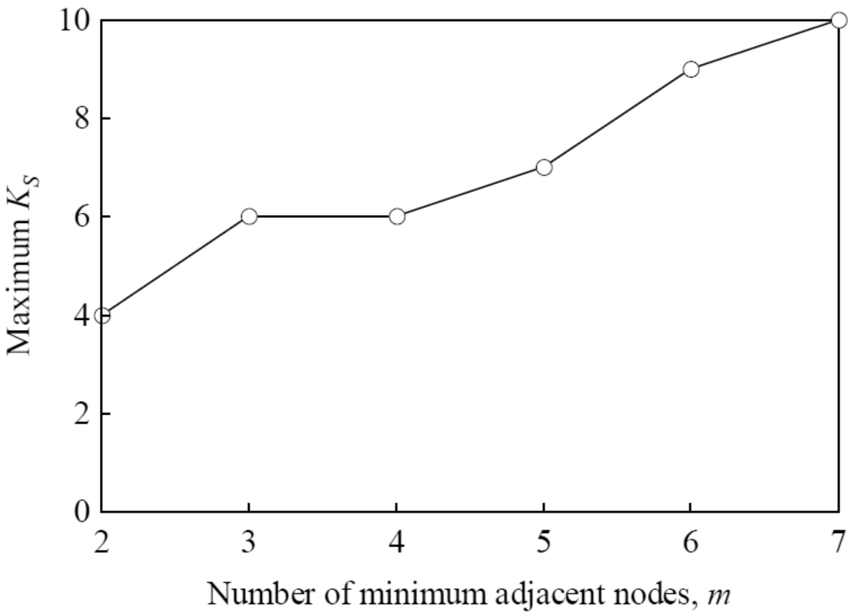


Figure 7.7: Dependency of K_S for varying number of minimum adjacent nodes m in the network.

to the source node for the purpose of splitting and encoding. The NMS directly configures/switches all the nodes along the disjoint paths, including the destination. In the signalling approach, the NMS sends the reply message, based on the results from the PCE, to the source node. The source node sends signalling messages to all the nodes along the disjoint paths, and configures the paths.

7.6.3.2 Additional functionalities at source and destination nodes

Table 7.8 summarizes the additional functionalities required at the source/destination node to support our presented technique. A schematic diagram of each node in the network is shown in Fig. 7.9. At the input line card of the source, traffic is split into multiple parts, encoded data to be sent along the protection path(s) is(are) created from these split parts by using XOR operations, and synchronization markers are inserted in each of the split parts including the encoded part periodically. These synchronization markers are used for the alignment of split parts before merging at the destination. The individual delay for each path is known. Based on this information, on the input line card of the destination node

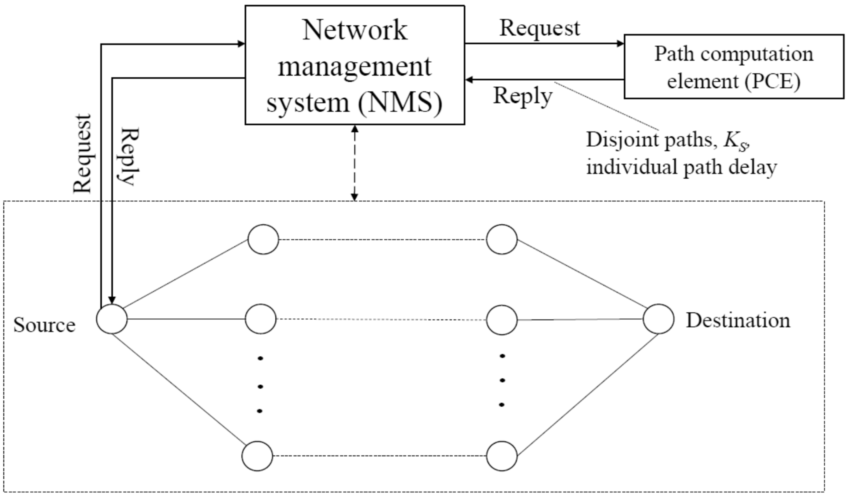


Figure 7.8: Path setup operation.

necessary delay buffers are employed. After all the split and encoded parts reach the input line card of the destination, they are sent to the output line card, where the split parts are synchronized and aligned by comparing the markers. After the synchronization is completed, merging (in case of failure decoding) of all the split parts, by using the XOR operations, is performed on the output line card of the destination.

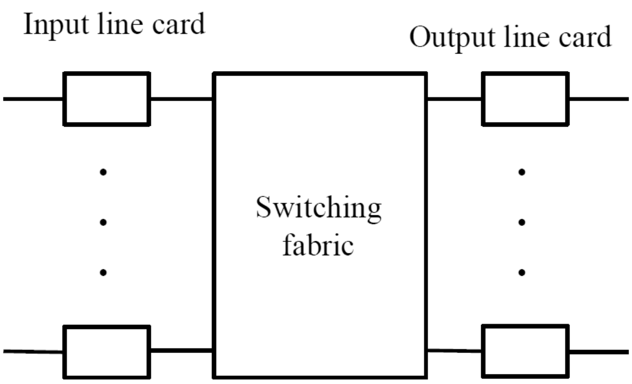


Figure 7.9: Basic switching fabric for source/destination node.

Table 7.8: Additional functionalities required for the presented technique

	Input line card	Output line card
Source node	Splitting, encoding, insert synchronization marker	†
Destination node	Delay buffer for each path	Synchronize split parts, merging

†No additional functionality.

7.6.4 Discussions

7.6.4.1 Encoding/decoding and buffering cost

The differential delay problem can be solved at the edge node by using electrical buffers, which are efficient in saving cost and power [71]. However, in optical networks in order to use electrical buffers we must use optical-electrical-optical signal conversions. With large number of wavelengths present, by using optical delay lines as optical buffers, we avoid the signal conversion [71]. Network service providers in practice can determine the buffering cost C_B according to their budget and service requirements. Similarly, the encoding/decoding costs Ψ can be determined by the type of XOR circuit used. Electrical XOR circuits are less expensive than optical XOR circuits.

Another issue, that network service providers have to consider, is the size of memory buffers used. Optical buffer is made of optical fibers, and is generally much larger than a electrical RAM chip of comparable capacity.

7.6.4.2 Limitation of the presented technique

The achievable data rate in optical fibers has reached the petabit/sec rate [86]. Optical transmission systems with 400 Gbps to 1 Tbps are being developed. Regardless of the path capacity, the spectrum resource in an optical fiber is roughly constant. In such high bit rate networks if we split traffic at the source, the use of more optical paths rather increases the cost of protection. If the given traffic is carried by a single high bit rate path and protected by another path, the resource usage and therefore cost will be minimized. Thus the presented technique is not advantageous for networks with leading-edge technologies.

7.6.4.3 Multiple connections among different source destination pairs

Assume that multiple connections exist among different source destination pairs, and the links in the network has enough capacity to serve all these requests (link capacity constraint is not considered). Under such assumptions, a set of disjoint routes for each connection can be independently determined by the optimization problem presented in this chapter. However, if the links in the network have finite capacities, then routing for one connection will affect/influence the routing of the remaining connections. Thus under limited capacity constraints, routing problems to allocate connections among different source and destination pairs are no more independent. In this situation a combined optimization problem for all the connections with link capacity constraints need to be considered.

7.7 Summary

This chapter presented a differential delay aware erasure correcting code based instantaneous recovery technique with traffic splitting. In this technique traffic is split into K equal parts and forwarded independently through a set of K disjoint paths. Multiple protection paths, P, provisioning is allowed in this technique in order to tackle multiple failures. A mathematical model was presented in order to determine $K+P$ disjoint paths. For the recovery of data on the failed path(s), all the split parts except the failed part(s) must be present at the destination. Each disjoint path may experience a different delay, and the recovery operation demands memory buffers for storing the split parts arriving earlier. The maximum allowable differential delay Δ determines the amount of memory buffers required for each disjoint path. In order to implement the presented technique the service providers must consider the memory buffers budgets incurred due to Δ. The effect of Δ on routing is thoroughly investigated in various random networks with respect to encoding/decoding costs, buffering costs, optical interface rates, the number of protection paths, the number of nodes in the network, and the number of minimum adjacent nodes. Simulation results observed that the K_S decreases with the increase of encoding/decoding costs, optical interface rates, and buffering costs. It is also observed that the K_S increases with the increase of the number of adjacent nodes and Δ values, and the number $K_S + P$ increases with the increase of P values. The path setup, encoding, synchronization, and merging (decoding) operations were discussed in order to support the presented technique.

Chapter 8

A Heuristic Routing Algorithm with Erasure Correcting Code Based Instantaneous Recovery Technique

This chapter presents a heuristic routing algorithm to design routes for all possible source destination pairs by provisioning erasure correcting code based instantaneous recovery technique with optimal traffic splitting (TS scenario), which was addressed for a source destination pair in chapter 6. We consider a static routing problem in networks having the coding capability. When the links in a network have finite capacities, assigning routing for all possible source destination pair by using this instantaneous recovery technique are mutually dependent, and this issue was not addressed. For the route designing purpose, one need to check routing for exponential number of traffic splitting number combinations. If the number of combinations to be considered, which equals the multiplication of each individual maximum possible traffic splitting numbers of all pairs considered, becomes extremely large obtaining a routing solution within a practical time is not possible. In order to achieve a routing solution within a practical time, the presented heuristic algorithm gives highest priority to the pair either with the largest cost or with the largest resource saving effect. For all source destination pairs the total path costs of implementing erasure correcting code based instantaneous recovery technique, and conventional 1+1 protection technique are computed. Almost 20% resource saving w.r.t. 1+1 protection is achieved in our examined networks.

8.1 Introduction

Multi-path provisioning can deliver many benefits and efficiency to network operators [51, 52]. However, with relatively high frequency of fiber cuts, and tremendous amount of traffic loss due to a single failure, survivability in multi-path provisioning is of crucial concern [1, 68]. Protection, a proactive procedure in which spare capacity is reserved during connection establishment, allows recovery from such failures. When a working path (i.e., a path carrying traffic during normal operation) fails, the connection is rerouted over to a backup path.

1+1 protection provides instantaneous recovery from any single link failure in the network because only the destination node switching to the backup path is necessary after a failure is detected. In order to achieve this, the bandwidth resource requirements between every possible source and destination have to be doubled.

In order to reduce the overall resources required for protection, several coding based recovery techniques have been addressed [14, 15, 20, 23, 24, 35, 37]. The network coding (NC) technique [12] based protection for unicast and multicast traffic in optical networks is addressed in literature [15, 23, 24]. In the NC technique intermediate nodes are allowed to encode several incoming data [54].

A.E Kamal presented $1+N$ protection by using the protection cycle (p-cycle) structure to handle single and multiple failures in optical mesh networks [15]. A.E. Kamal *et al.* also introduced $1+N$ protection technique that does not require a p-cycle [14]. Both the works [14, 15] presented integer linear programming (ILP) formulations in order to asses the cost of their respective protection techniques.

Manley *et al.* investigated the application of NC in the multicast protection in all-optical networks [23]. It also presented an ILP approach to find disjoint multicast protection tree. The design issues for NC in all-optical networks considering switching components, controllable optical delay lines, and all-optical XOR gates were also presented in [23]. Griffith *et al.* [24] presented an architecture for employing 1+1 protection in optical burst switched (OBS) networks, and examined the design issues caused by propagation delays of the two disjoint paths across the OBS networks. Meusburger *et al.* suggested that rerouting of 1+1 optical channels, for long term planning of incremental networks, should be cost-efficient [57].

The simplest technique used for encoding data in recent optical networks, either at the intermediate nodes (NC technique [54]) or at the source node (erasure correcting codes [58, 59]), is the exclusive-OR (XOR) operation. The implementation of all-optical XOR devices have been addressed in several independent researches [23, 60, 61, 62, 63]. In those researches the operation speed of all-optical XOR gates were reported as 10.7 Gbps [60], 10 Gbps [61, 62], and 20 and 40 Gbps [63].

1+1 protection techniques with erasure correcting code and NC were addressed in [20, 35], respectively. In [20] erasure correcting code based protection technique was illustrated with the traffic split into two equal parts only. [55] reported that traffic splitting based protection technique is most resource efficient and maintains instantaneous recovery. However, optimal traffic splitting was not addressed in [20] and [55]. Diversity coding based protection technique was presented in [56]. The work in [35] presented an NC based 1+1 protection technique that is applicable to scenarios where two source nodes have a common destination node, and this technique requires synchronization of flows from two different sources.

In order to minimize the resources required for protection from any single link failure, an erasure correcting code based instantaneous recovery technique with optimal traffic splitting was addressed in literature [37]. In that addressed technique, the traffic between a source destination pair, p, is split into $k(p)$ equal parts, where $k(p)$ is the traffic splitting number for pair p, and sent along $k(p)$ disjoint paths independently. The $k(p) + 1$th disjoint path is set to carry the encoded data formed from the $k(p)$ split parts. A traffic splitting number $k(p)$, for which the total cost of provisioned $k(p) + 1$ disjoint paths for pair p is minimum, is selected as the optimal traffic splitting number. Note that for any source destination pair p, $1 \leq k(p) \leq K_{max}(p)$, where $K_{max}(p)$ equals the smaller node degree between the two nodes in pair p minus one. The instantaneous recovery technique in [37] was addressed only for a single source destination pair, where link capacity constraints were not considered.

When the links in a network have finite capacities, assigning routing for all possible source destination pair by using the erasure correcting code based instantaneous recovery technique [37] are mutually dependent, and this issue was not addressed in [37]. The mutual dependency is explained in the following.

Let us assume that in a network, there are three source destination pairs, and each has a $K_{max}(p)$ of 2. Under finite link capacity constraints, we have to check the overall routing cost for all possible traffic split number (up to $K_{max}(p)$) combinations for these three pairs at the same time. We set traffic split number for the three pairs, respectively, as {1,1,1}, {1,1,2}, {1,2,1}, {1,2,2}, {2,1,1}, {2,1,2}, {2,2,1}, and {2,2,2}. The combination which gives the minimum total path cost is selected, and the corresponding routing is the desired solution. In general one need to check $\prod_{p \in P} K_{max}(p)$, where P is the set of all possible pairs, total cost combinations. When the number of pairs is large, the number of combinations to be considered becomes extremely large, and obtaining a routing solution for all possible source destination pairs within a practical time is not possible.

This chapter presents a heuristic routing algorithm in order to achieve a routing solution for all $p \in P$ within a practical time, where erasure correcting code based instantaneous recovery technique is employed. We consider a static routing problem in networks having the coding capability. In the presented algorithm, the highest priority is given to the pair either with the largest cost first policy, or with the largest resource saving effect first policy. For each pair, an iterative ILP model is used to determine a suitable traffic splitting number and the overall routing path cost of implementing erasure correcting code based instantaneous recovery technique. After routing is assigned, a pair is removed from P, the network residual link capacities are updated, and we repeat the whole procedure described earlier, until P is empty.

The remainder of this chapter is as follows. The erasure correcting code based instantaneous recovery technique with optimal traffic splitting is discussed in Section 8.2. Section 8.3 presents the terminologies and the mathematical model for determining a suitable $k(p)$ and $k(p)+1$ disjoint paths for pair p. The presented algorithm is presented in Section 8.4. The evaluation results are discussed in Section 8.5. Finally, Section 8.6 concludes the chapter.

8.2 Erasure correcting code based instantaneous recovery technique

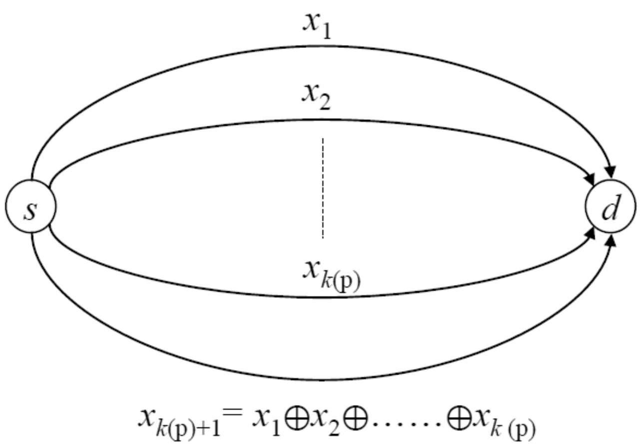


Figure 8.1: Network protection with optimal traffic splitting using erasure correcting code.

The erasure correcting code based instantaneous recovery technique with optimal traffic splitting, presented in [37], is re-produced in Fig. 8.1. Source node s and destination node d in Fig. 8.1 are the edge nodes in a network. Each of the edge nodes in the network is equipped with memory buffers for storing data. In the recovery technique the traffic between a pair p, consisting of nodes s and d, is split into $k(p)$ equal parts $(x_1, x_2, ..., x_{k(p)})$ and sent through $k(p)$ disjoint paths simultaneously. The split parts of the traffic are encoded at the source by simple exclusive-OR (XOR) operation as $x_{k(p)+1} = x_1 \oplus x_2 \oplus ..... \oplus x_{k(p)}$, and sent simultaneously onto the $k(p) + 1$th disjoint path.

Note that the encoding technique here is the erasure correcting code [59] but not the NC technique. This is because the encoding is performed only at the source node, while NC is performed at the intermediate nodes [54].

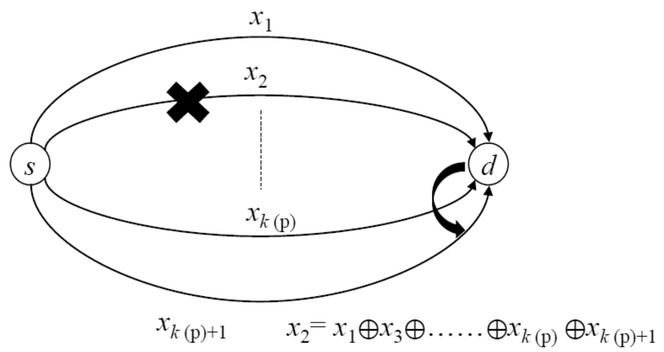


Figure 8.2: Recovery of failed data at the destination.

Instantaneous recovery is achieved in this technique by the introduction of the encoded data in the $k(p) + 1$th path as illustrated in Fig. 8.2. If any link carrying plain data fails, the destination node can detect the failure. Only the destination node needs to switch to the $k(p) + 1$th backup path. By using the plain data on the $k(p) - 1$ working paths and the encoded data on the $k(p) +$ 1th protection path, destination node instantaneously recovers the lost data sent along the failed path. For example, let the disjoint path carrying split part x_2 fails. Destination node d detects this, performs switching over to the $k(p) + 1$ path, which is a backup path, and recover lost data by simple XOR operation as $x_2 = x_1 \oplus x_3 \oplus ..... \oplus x_{k(p)} \oplus x_{k(p)+1}$.

8.3 Mathematical Preliminaries and Formulation

The mathematical model presented in this section is used to determine a suitable traffic splitting number $k(p)$ and $k(p) + 1$ disjoint paths between a source destination pair p.

8.3.1 Terminologies

The network $G(V, E)$ is represented as an undirected graph, where V is the set of vertices (nodes) and E is the set of edges (links). A link from node $i \in V$ to node $j \in V$ is denoted as $(i,j) \in E, i \neq j$. x_{ij}^m is the binary routing variable. If a link $(i,j) \in E$ belongs to disjoint path number $m \in M$, then x_{ij}^m is 1, otherwise 0, where $M = \{1, 2, \cdots k(p)+1\}$. c_{ij} is the cost of link $(i,j) \in E$ per unit bandwidth. Let d_{sd} be the traffic demand from the source s to the destination d of pair p. It is assumed that the network is at least two-connected to facilitate the use of 1+1 protection technique. Encoding is done at the source, while decoding is done at the destination upon a failure occurs. Let the residual capacity of each link is κ_{ij}. P be the set of all possible source destination pairs.

8.3.2 Mathematical formulation

For a source destination pair we present an optimization problem by formulating an integer linear programming (ILP) one, where link capacity constraints are considered, and the total cost is affected by the actual total bandwidth usage.

The optimization problem is expressed as follows:

$$\min_{k(p)\geq 1} \quad d_{sd} \cdot \left[\frac{f(k(p)+1)}{k(p)} \right], \tag{8.1a}$$

where

$$f(k(p)+1) = \quad \min \sum_{m \in M} \sum_{(i,j) \in E} c_{ij} x_{ij}^m \tag{8.1b}$$

subject to:

$$\sum_{j \in V} x_{ij}^m - \sum_{j \in V} x_{ji}^m = 1, \text{ if } i = s, \forall m \in M \tag{8.1c}$$

$$\sum_{j \in V} x_{ij}^m - \sum_{j \in V} x_{ji}^m = 0, \; \forall i \neq s, t \in V, \forall m \in M \tag{8.1d}$$

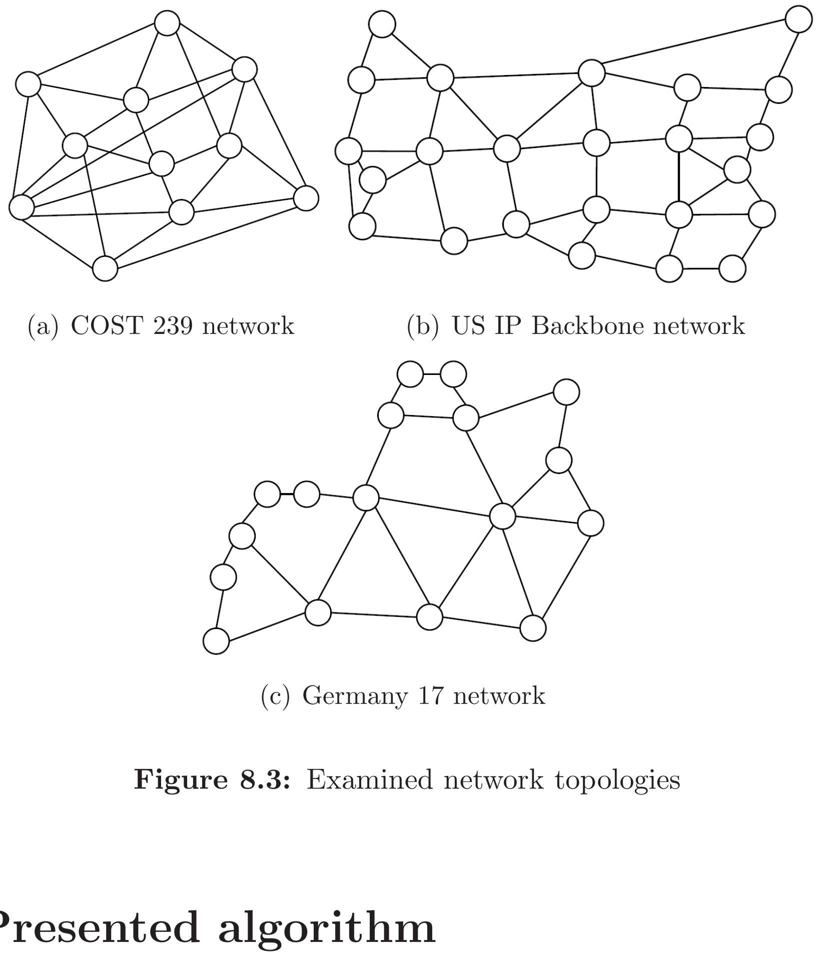
$$x_{ij}^m + x_{ij}^{m'} \leq 1, \forall m, m' (m \neq m') \in M, \forall (i,j) \in E \tag{8.1e}$$

$$\frac{d_{sd}}{k(p)} \cdot x_{ij}^m \leq \kappa_{ij}, \forall (i,j) \in E \tag{8.1f}$$

$$x_{ij}^m = \{0, 1\}, \forall m \in M, \forall (i,j) \in E. \tag{8.1g}$$

The objective function in Eq. (8.1a) chooses a suitable value of $k(p)$, for which the total cost of $k(p)$+1 disjoint paths for pair p is minimum. Eq. (8.1b) minimizes the total cost of sending each unit of traffic demand through the set of $k(p)$+1 disjoint paths in the network. Eqs. (8.1c) and (8.1d) express the flow conservation constraints at the source node and at intermediate nodes, respectively. The path disjoint constraint is shown in Eq. (8.1e), which states that different paths do not share any common link. Eq. (8.1f) describes the link capacity constraint, which states that the used capacity in each link must not exceed the residual capacity. The binary routing variable is described by Eq. (8.1g).

Note that, throughout the rest of the chapter we use the term 'iterative ILP model' to refer to this optimization problem. In the optimization process an ILP model, expressed by Eqs.(8.1b)-(8.1g), is solved iteratively by fixing $k(p)$ to different integer values between $[1, K_{max}(p)]$ and then it is checked whether Eq. (8.1a) is satisfied.



(a) COST 239 network (b) US IP Backbone network

(c) Germany 17 network

Figure 8.3: Examined network topologies

8.4 Presented algorithm

We present a heuristic routing algorithm where, at first, priorities are assigned to all the source destination pairs according to either policy 1 or policy 2 (to be described later in this section), then a pair with the highest priority is selected and routing is assigned to it[1]. After routing is assigned, for the remaining pairs we update the residual link capacities, and re-assign priorities based on updated link capacities.

8.4.1 Algorithm description

The algorithm is as follows,

- Step 1: Compute and assign priorities to all the source destination pair $p \in P$ according to either policy 1 or policy 2. In both policies, the iterative ILP model is used.
- Step 2: Select a pair with the highest priority. By using the results of iterative ILP model obtained in Step 1, routing is assigned for the selected pair.
- Step 3: Remove p from P and update the residual capacities of the links used in Step 2.
- Step 4: If updated P is not empty, go to Step 1. Otherwise, go to Step 5.
- Step 5: The algorithm stops.

8.4.2 Policies to select a pair first

We have two policies to assign priorities to all the pairs in Step 1 of our presented algorithm. The two policies are described in the following.

8.4.2.1 Policy 1 - Largest cost first policy

The highest priority is given to the source destination pair with the largest cost. By using the iterative ILP model we compute the cost of implementing erasure correcting code based instantaneous recovery technique by considering the residual link capacities. The pair with the largest cost is selected and routing is

[1]Since assigning routing for all $p \in P$ at the same time by considering link capacities is not possible in a practical time, we assign routing to one pair at a time.

assigned to it. We select a pair with the largest cost because this demand has the most impact on routing and residual link capacities. After routing is assigned for a pair, it is removed from P (in Step 3), and the residual link capacities are updated. We repeat the whole procedure for the remaining $p \in P$, until $P = \emptyset$.

8.4.2.2 Policy 2 - Largest resource saving effect first policy

Prior to describe policy 2, we define a metric named "resource saving effect". For a pair p the resource saving effect, $\mathcal{R}(p)$, is defined as the difference of total cost for suitable $k(p) + 1$ paths and the total cost for two disjoint paths used for conventional 1+1 protection.

In policy 2, highest priority is given to the pair with the largest $\mathcal{R}(p)$. For all $p \in P$, by using the iterative ILP model, determine the cost of $k(p) + 1$ paths for erasure correcting code based instantaneous recovery technique, and $k(p) + 1 = 2$ paths for conventional 1+1 protection technique, and compute $\mathcal{R}(p)$. The pair with the largest $\mathcal{R}(p)$ is selected, and routing is assigned for the selected pair in step 2. Then it is removed from P and residual link capacities are updated in Step 3 of our presented algorithm. We repeat the whole procedure for the remaining $p \in P$, until $P = \emptyset$.

8.4.3 Algorithm complexity

The presented algorithm uses an iterative ILP model to assign routing for each considered pair. A complexity measure for an ILP model was presented in [16]. We follow a similar measure to derive the complexity of the iterative ILP model presented in this chapter. The ILP has $|E|$ variables, and approximately $(3|E| + \frac{|E|(|E|-1)}{2})$ constraints, where $|E|$ is the number of links. Thus the complexity of our presented ILP is dominated by $O(|E|^3)$. For a pair p, the ILP model has to run $K_{max}(p)$ times, where $K_{max}(p)$ is the maximum possible traffic split number for the considered pair. Thus for a pair the iterative ILP model has an overall complexity of $O(|E|^3)K_{max}(p)$.

In our presented algorithm, whether the largest cost first policy or the largest resource saving effect first policy is used, the iterative ILP approach is used $\frac{|P|(|P|+1)}{2}$ times, where $|P|$ is the number of all possible source destination pairs.

This is because in Step 1 of the presented algorithm, in order to compute largest cost (policy 1) or $\mathcal{R}(p)$ (policy 2) for the updated residual capacities, the iterative ILP approach is used for all the pairs in the updated P again. Note that the $O()$ notation disregards any constants [65]. Thus, with either policy, our presented algorithm has a complexity of the order of $O(|P|^2|E|^3\mathcal{K})$, where $\mathcal{K}$ is the maximum value of $K_{max}(p)$ over P.

8.5 Performance Evaluation and Discussion

We evaluate the performance of the presented algorithm, with both largest cost first and largest resource saving effect first policies, by comparing the total cost of all the routes designed by using our presented algorithm to the total cost of all the paths when conventional 1+1 protection technique is used. Note that the total path cost in all the cases include all the disjoint paths corresponding to all possible source destination pair. We evaluate our presented algorithm in the three networks shown in Fig. 8.3. The three networks are known as COST 239 network, US IP Backbone network, and Germany 17 network [67].

In the three networks link capacities are kept to a fixed value of 200 traffic unit. The traffic demands for all possible source destination pairs are randomly assigned in the range of 10 to 50 traffic units. The iterative ILP in Step 2 of the presented algorithm is solved by using the CPLEX® [46] mathematical programming solver.

When the conventional 1+1 protection technique is employed, highest priority is given to the source destination pair with the largest traffic demand. We prepare a priority list of all the pairs $p \in P$ according to the decreasing traffic demands. The pair at the top of priority list is selected. Routing is assigned to the selected pair by using the iterative ILP model with the constraint that only two disjoint paths between a pair is allowed. Once fixed, the priority order in the list is not changed. After routing is assigned to a pair, it is removed from the top of the list, residual link capacities are updated, and the following pair in the list becomes the highest priority pair to which routing is assigned next.

The normalized total cost for all the paths designed by using the erasure correcting code based instantaneous recovery technique and the conventional 1+1

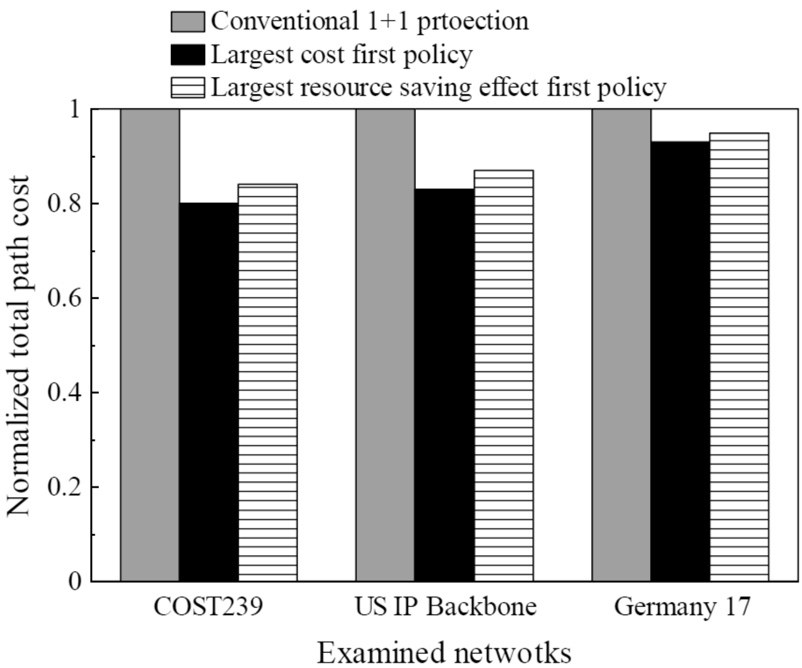


Figure 8.4: Comparison of total path costs, normalized w.r.t. 1+1 protection cost.

protection technique, in the three evaluated networks, are presented in Fig. 8.4. Note that the total costs are normalized with respect to the total conventional 1+1 protection cost.

From the results of Fig. 8.4 it is observed that, the largest cost first policy outperforms largest resource saving effect first policy in our examined networks. Figure 8.4 illustrates that almost 20% resource saving is achieved in the COST 239 network by using traffic splitting based routing with the largest cost first policy. In the COST 239 network the least node degree is four, and all the source destination pair in this network can avail the resource saving advantage of the erasure correcting code based instantaneous recovery technique. In the US IP Backbone and Germany 17 networks, there are a number of pairs to which erasure correcting code based instantaneous recovery technique is not applicable, because only two disjoint paths exist for those pairs in the network. Thus some pairs cannot obtain the benefit of resource saving due to traffic splitting in these two networks. This is the reason of lower resource saving in the US IP backbone and Germany 17 networks.

8.6 Summary

This chapter presented a heuristic routing algorithm to design routes for all possible source destination pairs, in a practical time, by provisioning erasure correcting code based instantaneous recovery technique with optimal traffic splitting. Our presented heuristic algorithm gives the highest priority to the pair either with the largest cost or with the largest resource saving effect. For all source destination pairs the total path costs of implementing erasure coding based instantaneous recovery technique with both policies, and the conventional 1+1 protection technique are computed. Almost 20% resource saving w.r.t. 1+1 protection was achieved in our examined networks. Evaluation results observed that the largest cost first policy outperforms the largest resource saving effect first policy in our examined networks.

Chapter 9

Conclusions and future work

9.1 Conclusions

This thesis has proposed two reliable route design schemes for the MSCD and TS scenarios with the same objective of achieving instantaneous recovery at the expense of reduced backup resources. Both MSCD and TS scenarios (presented in previous works) have different requirements regarding node connectivity and encoding technique. This is why two different routing schemes for two different scenarios are proposed.

In the first scheme with *MSCD scenario* the route design problem, where an optimum set of network coding (NC) aware routes that minimizes the cost of 1+1 protection in any scenario with $k \geq 2$ sources and a common destination, is formulated as an integer linear programming (ILP) formulation (chapter 2). An ILP model is not practically solvable for large scale networks due to the requirement of large amount of computer memory (RAM) and CPU processing power. In order to achieve a near optimal routing solution in large scale networks, a heuristic algorithm is presented (chapter 3). A mathematical model framework of 13 ILP models was presented in chapter 4 in order to design optimum routes in $3SD$ scenarios. A heuristic routing algorithm is presented in order to design NC bases 1+1 protection routes for all possible source destination pair in the network. In chapter 5, an iterative ILP model is presented for optimum route design in $2SD$ scenarios, where the common destination node's degree is ≥ 2. In

Appendix A, an ILP with unequal traffic demand and link capacity constraints is presented.

Our first proposed routing scheme with MSCD scenario meets all the requirements stated in section 1.4 of chapter 1. It achieves instantaneous receiver initiated recovery (req. 1), it requires less backup resources than the conventional 1+1 protection (req. 2), simple bitwise exclusive OR (XOR) operations are used for encoding and decoding (req. 3), and it is applicable in scenarios where one or both source and destination node is connected to only two neighbor nodes (req. 4).

In the second proposed scheme with *TS scenario*, traffic is split at the source, encoded data is created using XOR operations on the split parts, and all the split parts including the encoded one are sent to the destinationvia disjoint paths. An ILP model is presented in chapter 6 which determines a suitable traffic splitting number and a set of disjoint routes that minimizes the cost of protection. Each disjoint path experiences a delay, and the destination node has to buffer all the split parts arriving earlier until the last part arrives. After this recovery operation is performed, if necessary. Differential delay (between any two employed disjoint paths) aware routing with traffic splitting is discussed in chapter 7. The path setup, encoding, synchronization, and merging operations to support this scheme was also discussed. When the links in the network have finite capacities, routing with splitting for one pair affects the routing of other pairs. In chapter 8, a heuristic routing algorithm is presented in order to design traffic splitting based protection routes for all possible source destination pair under the link capacity constraints.

The second proposed routing scheme meets the first three requirements mentioned in chapter 1. However, it is not applicable if one or both source and destination node is connected to only two neighbor nodes. In this case, 1+1 protection without splitting is the only solution.

In order to evaluate the effectiveness and applicability, the performance of our two proposed route design schemes were evaluated in terms of the total resource saving with respect to the conventional 1+1 protection technique. Table 9.1 below summarizes the evaluated resource saving results obtained by using our proposed schemes. In the evaluation protection routes for all possible source destination pairs were designed by the two proposed schemes, respective, in the three evaluated networks.

Table 9.1: Comparison of two proposed schemes

	Resource saving %	
Examined networks	Scheme with MSCD scenario	Scheme with TS scenario
COST 239	15%	20%
Germany 17	7.1%	7%
US IP Backbone	11 %	17%

From the evaluation results of Table 9.1, we conclude that our proposed routing scheme with *MSCD scenario* is suitable for sparse networks (for example Germany 17 network), while the proposed scheme with *TS scenario* is advantageous for dense networks (i.e. COST 239 network). The routing scheme with the *TS scenario* achieves a higher resource saving than that of the scheme with the *MSCD scenario*. However, in sparse networks opportunity of splitting is less, and even does not exist for some pair. In this situation the routing scheme with the *MSCD scenario* can be applied in order to design coding aware resource efficient routes.

9.2 Future work

One important concern is "Should we use coding technique for reliable communication network design?" It is necessary to have an idea on how much recourse saving is possible if coding technique is used for reliable route design for communication networks. The amount of resource saving will create interest for the investors and researchers in communication community to use coding in network design. In this thesis, we address the coding based route design problem to find the amount of resource saving achieved by using coding technique with protection techniques.

For coding based survivable network design the following issues should be investigated.

- Which layer?
  - At lower layer, hardwired coding technique. Hardware cost is high, and increases network node size.
  - At higher layer, software based coding technique. Easy to control coding operations, less expensive than hardwired solution.
- Synchronization: Bit level, frame level, packet level, or segment level.
- Protocol to manage synchronization and coding

In this research we considered that the traffic demand is exactly known. However, it is difficult to measure the exact traffic demands, and traffic demands fluctuate in present high speed networks. It would be interesting to design our proposed schemes for uncertain traffic demands. To tackle the issue of uncertain demands we have to re-design our routing scheme, where the objective will be to minimize the worst case network congestion ratio (maximum link utilization over all links).

Appendix A

ILP for $2SD$ scenarios with unequal traffic demands

A.1 Effect of unequal traffic demand on NC

The effect of unequal traffic demands on resource saving using the NC technique is described with reference to Fig. A.1. Source nodes A and B have data X_1 and X_2 respectively to send to the common destination node D. Let ω_{AD} and ω_{BD} are denoted as the traffic demands between nodes A and D, and between nodes B and D, respectively. ω_{AD} and ω_{BD} either have the same value or have different values. Let β_{ij} denote the bandwidth demand, and γ_{ij} indicates the bandwidth saving due to the NC technique on any link $(i, j) \in E$ in the network. In Fig. A.1, paths $P_{A1} = A - D$ and $P_{B1} = B - D$ carry plain data X_1 and X_2, while $P_{A2} = A - C - D$ and $P_{B2} = B - C - D$ experience the NC effect. $C - D$ is the common link and C is the *NC node*. Depending on the values of ω_{AD} and ω_{BD}, link $C - D$ carries one of the following three combinations of data.

1. If $\omega_{AD} = \omega_{BD}$, it carries X_0, $X_1 \oplus X_2$, and $\beta_{CD} = \omega_{AD} = \omega_{BD}$. Note that, X_0 means no data.

2. $\omega_{AD} > \omega_{BD}$, it carries X_1, $X_1 \oplus X_2$, and $\beta_{CD} = \omega_{AD}$.

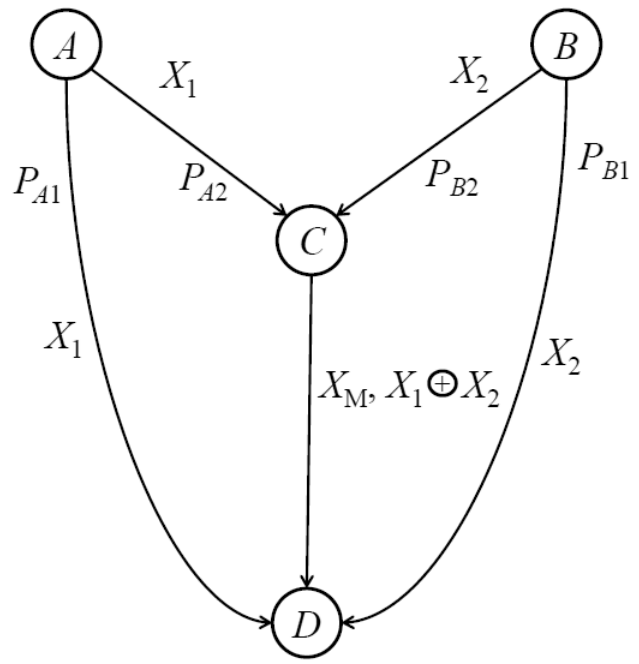


Figure A.1: Effect of unequal traffic demands on network coding along the link $C-D$. The value of X_M is determined by the bandwidth demands of X_1 and X_2.

3. $\omega_{AD} < \omega_{BD}$, it carries X_2, $X_1 \oplus X_2$, and $\beta_{CD} = \omega_{BD}$.

The β_{CD} depends on the values of ω_{AD} and ω_{BD}. When both demands are equal, X_1 and X_2 are compressed as $X_1 \oplus X_2$, and $\beta_{CD} = \omega_{AD} = \omega_{BD}$. If source A has larger traffic demand than that of source B, i.e., $\omega_{AD} > \omega_{BD}, \omega_{BD}$ amount of X_1 data is XORed with X_2, requiring ω_{BD} bandwidth. The rest of the X_1 data is sent as it is through link $C-D$, which requires a bandwidth of $\omega_{AD} - \omega_{BD}$. Thus, we obtain $\beta_{CD} = \omega_{BD} + \omega_{AD} - \omega_{BD} = \omega_{AD}$, which is equal to the bandwidth corresponding to higher demand, and $\gamma_{CD} = \omega_{BD}$. In a similar way, for the case of $\omega_{AD} < \omega_{BD}$, we obtain $\beta_{CD} = \omega_{BD}$ and $\gamma_{CD} = \omega_{AD}$.

A.2 Mathematical model for unequal traffic demands with link capacity constraints

In this model unequal traffic demands and link capacities are included as constraints. $T = \{\omega_{sd}\}$ is the traffic matrix, where ω_{sd} is the traffic demand between source node s and destination node d. Let κ_{ij} be the capacity of $(i, j) \in E$ and σ be the set of the link capacities, i.e., $\kappa_{ij} \in \sigma$.

A new model includes the effect of unequal traffic demands in the objective function. The link capacity constraints are also included. We introduce two binary variables h_1 and h_2 to determine the smaller traffic demand between the two. The ILP model includes five binary variables. The third one is the binary routing variable $x_{ij}^{sd,k}$. The fourth and the fifth binary variables z_{ij}^1 and z_{ij}^2 are used to determine proper NC effect depending on the routing variable, h_1 and h_2. The input parameters to this ILP model are source and destination nodes s_1, s_2, $d \in V$, number of nodes in the network $\mathcal{N}$, the cost of each link c_{ij}, capacity of each link κ_{ij}, and demands of each possible source-destination pair ω_{sd}, where $i, j, s, d \in V$.

The model, belonging to the ILP class, with unequal traffic demands and link capacity constraints is as follows:

$$\min \sum_{(i,j)\in E} \sum_{k=1}^{2} c_{ij} \times (\omega_{s_1 d} \times x_{ij}^{s_1 d,k} + \omega_{s_2 d} \times x_{ij}^{s_2 d,k}) - \sum_{(i,j)\in E} (c_{ij} \times \omega_{s_1 d} \times z_{ij}^1) - \sum_{(i,j)\in E} (c_{ij} \times \omega_{s_2 d} \times z_{ij}^2) \tag{A.1}$$

$$s.t. \quad x_{ij}^{s_1 d,1} + x_{ij}^{s_1 d,2} \leq 1, \forall (i, j) \in E \tag{A.2}$$

$$x_{ij}^{s_2 d,1} + x_{ij}^{s_2 d,2} \leq 1, \forall (i, j) \in E \tag{A.3}$$

$$x_{ij}^{s_1 d,1} + x_{ij}^{s_2 d,1} \leq 1, \forall (i, j) \in E \tag{A.4}$$

$$x_{ij}^{s_1 d,1} + x_{ij}^{s_2 d,2} \leq 1, \forall (i, j) \in E \tag{A.5}$$

$$x_{ij}^{s_1 d,2} + x_{ij}^{s_2 d,1} \leq 1, \forall (i, j) \in E \tag{A.6}$$

$$\sum_{j\in V}(x_{ij}^{s_1d,1} - x_{ji}^{s_1d,1}) = 1,\ i = s_1 \tag{A.7}$$

$$\sum_{j\in V}(x_{ij}^{s_1d,2} - x_{ji}^{s_1d,2}) = 1,\ i = s_1 \tag{A.8}$$

$$\sum_{j\in V}(x_{ij}^{s_2d,1} - x_{ji}^{s_2d,1}) = 1,\ i = s_2 \tag{A.9}$$

$$\sum_{j\in V}(x_{ij}^{s_2d,2} - x_{ji}^{s_2d,2}) = 1,\ i = s_2 \tag{A.10}$$

$$\sum_{j\in V}(x_{ij}^{s_1d,1} - x_{ji}^{s_1d,1}) = 0,\ \forall i \in V, i \neq s_1, d \tag{A.11}$$

$$\sum_{j\in V}(x_{ij}^{s_1d,2} - x_{ji}^{s_1d,2}) = 0,\ \forall i \in V, i \neq s_1, d \tag{A.12}$$

$$\sum_{j\in V}(x_{ij}^{s_2d,1} - x_{ji}^{s_2d,1}) = 0,\ \forall i \in V, i \neq s_2, d \tag{A.13}$$

$$\sum_{j\in V}(x_{ij}^{s_2d,2} - x_{ji}^{s_2d,2}) = 0,\ \forall i \in V, i \neq s_2, d \tag{A.14}$$

$$h_2 - h_1 = 1,\ \text{for}\ \omega_{s_1d} \geq \omega_{s_2d} \tag{A.15}$$

$$h_1 - h_2 = 1,\ \text{for}\ \omega_{s_1d} < \omega_{s_2d} \tag{A.16}$$

$$\omega_{s_1d} \times x_{ij}^{s_1d,1} \leq \kappa_{ij},\ \forall (i,j) \in \sigma \tag{A.17}$$

$$\omega_{s_1d} \times x_{ij}^{s_1d,2} \leq \kappa_{ij},\ \forall (i,j) \in \sigma \tag{A.18}$$

$$\omega_{s_2d} \times x_{ij}^{s_2d,1} \leq \kappa_{ij},\ \forall (i,j) \in \sigma \tag{A.19}$$

$$\omega_{s_2d} \times x_{ij}^{s_2d,2} \leq \kappa_{ij},\ \forall (i,j) \in \sigma \tag{A.20}$$

$$x_{ij}^{sd,k} = \{0,1\},\ \forall (i,j) \in E,$$
$$s = s_1, s_2, k = 1, 2 \tag{A.21}$$

$$h_1 = \{0,1\}, \tag{A.22}$$

$$h_2 = \{0,1\}, \tag{A.23}$$

$$z_{ij}^1 = \{0,1\},\ \forall (i,j) \in E \tag{A.24}$$

$$z_{ij}^1 \leq x_{ij}^{s_1d,2},\ \forall (i,j) \in E \tag{A.25}$$

$$z_{ij}^1 \leq x_{ij}^{s_2d,2},\ \forall (i,j) \in E \tag{A.26}$$

$$z_{ij}^1 \leq h_1,\ \forall (i,j) \in E \tag{A.27}$$

$$z_{ij}^1 \geq x_{ij}^{s_1d,2} + x_{ij}^{s_2d,2} + h_1 - 2,\ \forall (i,j) \in E \tag{A.28}$$

$$z_{ij}^2 = \{0,1\},\ \forall (i,j) \in E \tag{A.29}$$

$$z_{ij}^2 \leq x_{ij}^{s_1d,2}, \forall(i,j) \in E \tag{A.30}$$

$$z_{ij}^2 \leq x_{ij}^{s_2d,2}, \forall(i,j) \in E \tag{A.31}$$

$$z_{ij}^2 \leq h_2, \forall(i,j) \in E \tag{A.32}$$

$$z_{ij}^2 \geq x_{ij}^{s_1d,2} + x_{ij}^{s_2d,2} + h_2 - 2, \forall(i,j) \in E. \tag{A.33}$$

The objective function in Eq. (A.1) is modified properly to get the optimum minimum cost of employing NC based 1+1 protection in any scenario with two sources and a common destination with unequal traffic demands. The second and the third terms in Eq. (A.1) are responsible for providing the NC effect with the help of two binary decision variables z_{ij}^1 and z_{ij}^2. Eqs. (A.15) and (A.16) are the constraints that identifies the higher and lower between the two traffic demands and set the binary variables h_1 and h_2, respectively in Eqs. (A.22)-(A.23), accordingly. Eq. (A.21) describes the binary routing variable. Eqs. (A.17)-(A.20) describe the link capacity constraints, which tell that the total traffic on any link belonging to the employed paths should not exceed the specified capacity. Eqs. (A.2) and (A.6) specify the path disjoint constraints while the flow conservation constraints are specified by the Eqs. (A.7)-(A.14), these constraints are the same in both the ILP models presented in this paper. Eqs. (A.24)-(A.33) describe the binary variables to determine the proper links to employ NC. Eq. (A.28) describes that, if $\omega_{s_1d} < \omega_{s_2d}$ then z_{ij}^1 is 1, otherwise 0. Eq. (A.33) describes that, if $\omega_{s_1d} \geq \omega_{s_2d}$ then z_{ij}^2 is 1, otherwise 0.

Appendix B

Discussion on theoretical complexity of the considered network coding based 1+1 protection route design problem

Technique to prove a new problem is NP-Complete

Given a new problem $\mathcal{X}$, the basic strategy for proving it is NP-Complete is as follows [89].

1. Prove that $\mathcal{X}$ is an NP problem.

2. Choose a problem $\mathcal{Y}$ that is known to be NP-Complete.

3. Prove that $\mathcal{Y}$ is polynomial time reducible to X, i.e., $\mathcal{Y} \leq_P \mathcal{X}$.

Problem description when k is a variable

Our considered network coding [12] based 1+1 protection route design problem and the Steiner tree problem, when k is a variable , are defined in the NP form in the following.

Network coding based 1+1 protection route design problem (NCR)

Instance:

- A graph $G = (V, E)$, where V is the set of vertices, and E is the set of edges.
- An edge-cost function: $\mathcal{C}$: $E \to Z^+$, where Z^+ is the set of nonnegative integers.
- A set T of k sources $\{v_1, v_2, \cdots, v_k\} \in V$.
- A destination $d \in V$
- An integer: h.

Question

Is there any set F of edges forming network coding based protection routes for T and d with $\mathcal{C}(F) \leq h$?

Requirement:

Provide a "YES" or "NO" answer to the above mentioned question for any instance of NCR.

Note that, T and d constitute multiple sources and a common destination (MSCD) scenario with k sources, respectively, denoted by kSD.

Steiner tree problem (ST)

Instance:

- A graph $G' = (V', E')$, where V' is the set of vertices, and E' is the set of edges.
- An edge-cost function: $\mathcal{C}'$: $E' \to Z^+$, where Z^+ is the set of nonnegative integers.
- A set T' of k' terminals $\{v'_1, v'_2, \cdots, v'_{k'}\} \in V'$
- An integer: h'

Question
Is there any subtree F', that spans all the k' terminals $\in T'$, with $\mathcal{C}'(F') \leq h'$?
Requirement:
Provide a "YES" or "NO" answer to the above mentioned question for any instance of ST.
Lemma 1
The NCR problem, when k is a variable, in an NP-Complete problem.
Discussion - how ST and NCR are related
A feasible routing solution of our considered NCR problem in any kSD scenario includes a combination of a Steiner tree, on which a common destination, d, and k sources exist, and multiple disjoint paths from the k sources to the common destination. For example, consider an MSCD scenario with $k = 3$ sources, as shown in Fig. B.1 below.

A Steiner tree (rooted at the common destination, indicated by dotted lines in Fig. B.1), that connects the common destination d to all the given k sources $\in T$, serves as the backup protection paths for the given sources. Network coding (NC) can be applied at some intermediate node of this tree. k disjoint paths (indicated by solid lines in Fig. B.1) from all the given sources to the common destination, which must also be disjoint from the Steiner tree mentioned earlier, serve as working paths.

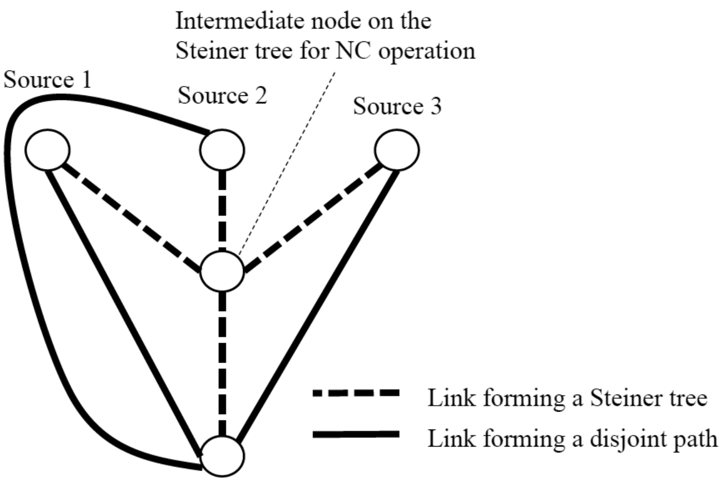


Figure B.1: A solution of the considered routing problem includes a Steiner tree finding approach

Proof of Lemma 1

Polynomial time reduction of any instance of ST into an instance of NCR

Let $X = (G', T', \mathcal{C}', h')$ be any given instance of ST. We define a function f that reduces any given instance X of ST into an instance $Y = (G, T, d, \mathcal{C}, h)$ of NCR in polynomial time. In other words, $Y = f(X)$.

Description of function f

Input

Any instance $X = (G', T', \mathcal{C}', h')$ of ST.

Processing

- Let $V = V'$, $d = v'_{k'}$, $T = T' \setminus \{v'_{k'}\}$, $h = h'$, $k = k' - 1$, and $\mathcal{C}$=$\mathcal{C}'$.
- Add zero-cost edges (v'_i, d) for $i = 1, 2, \cdots, k' - 1$.
- Thus $E = E' \cup \{(v'_1, d), (v'_2, d), \cdots, (v'_{k'-1}, d)\}$.

Output

An instance $Y = (G, T, d, \mathcal{C}, h)$ of NCR.

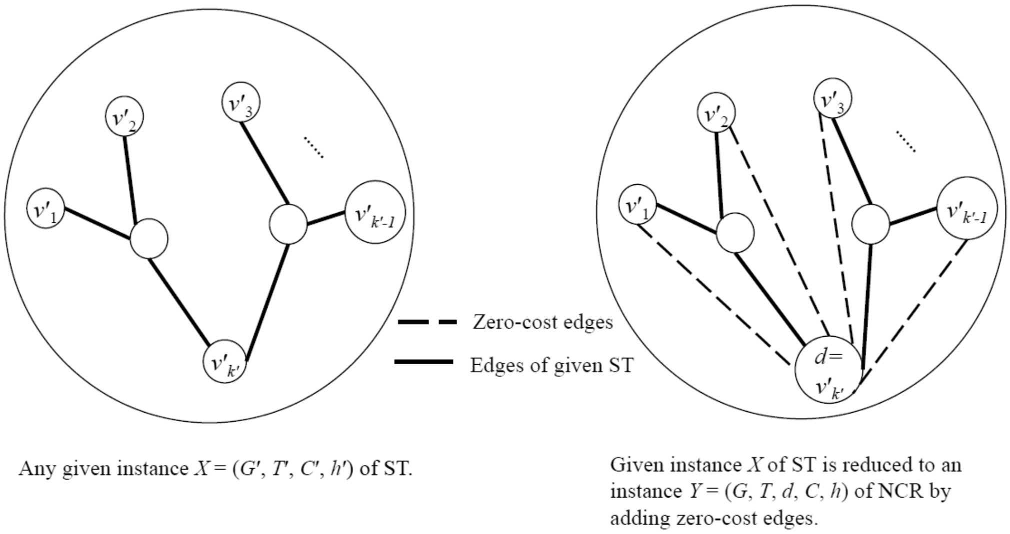


Figure B.2: Illustration of a function to reduce any given instance X of ST into an instance Y of NCR in polynomial time.

Thus, using function f, we can reduce any given instance X of ST into an instance Y of NCR in polynomial time, which is described in Fig. B.2.

$$X \leq_P Y \tag{B.1}$$

Note that, X is feasible if and only if Y is feasible,

$$X \in ST \Longleftrightarrow Y \in NCR \tag{B.2}$$

In order to complete the proof of Lemma 1, we have to prove that Eq. B.2 is true in both ways. We split the proof of Eq. B.2 into two parts, one for each implication.

- ST $\Rightarrow$ NCR
  According to our described function f any given instance of ST is transformed into an instance of NCR by adding zero-cost edges. Thus both instances of ST and NCR have the same cost. We also set $h = h'$ as defined in f. Therefore, a feasible solution exists for the transformed instance of NCR (with cost $\leq h$) if the given instance of ST has a cost $\leq h'$.

- ST $\Leftarrow$ NCR

  Suppose there exists an instance of NCR with cost w, where $w = a+b$ is the total cost of primary (a) and NC based backup paths (b). NCR is feasible if $w \leq h$. $F \subseteq E$ includes all the edges that belong to an instance of NCR. Let $F_1 \subset F$ be a set of edges that belong to the primary disjoint working paths, and $F_2 \subset F$ be a set of edges belonging to the NC based backup paths. By removing the edges of F_1 we can transform any given instance of NCR into an instance of ST with cost b. By definition a is nonnegative. Thus, if $w = a + b \leq h$ is satisfied, $b \leq h$ is also satisfied. Therefore, a feasible solution exists for the transformed instance of ST (with cost $\leq h'$) if the given instance of NCR has a cost $\leq h$.

From Eq. (B.2) we have the following corollary.

Corollary

If NCR has a polynomial time algorithm, then ST also has a polynomial time algorithm.

However, the problem of finding an instance X of ST among a subset of k vertices, where k is a variable, is known to be NP-Complete [90]. Since ST is an NP-Complete problem, the NCR problem is also an NP-Complete problem. (Lemma 1 is proved) □

NCR problem includes zero-cost edges, and for the polynomial time transformation of any instance of ST into an instance of NCR we added zero-cost edges. In our original NCR (O-NCR) problem that is discussed in chapters 2 and 3, zero-cost edges do not exist. We can transform any given instance of ST into an instance of O-NCR in polynomial time, where Bhandari's edge-disjoint shortest pair algorithm [44] is used iteratively to find k disjoint paths among k sources and the common destination. The transformation from NCR to O-NCR is straightforward. Therefore, our discussion mainly focuses on the transformation ST $\leq_P$ NCR.

Network coding based minimum cost route design problem (Min-NCR)

Instance:

- A graph $G = (V, E)$, where V is the set of vertices, and E is the set of edges.
- An edge-cost function: $\mathcal{C}$:$E \rightarrow Z^+$, where Z^+ is the set of nonnegative integers.
- A set T of k sources $\{v_1, v_2, \cdots, v_k\} \in V$.
- A destination $d \in V$

Requirement:

- Determine a set F of edges, forming network coding based 1+1 protection routes for T and d, that minimizes the $\mathcal{C}(F)$ of protection.

Note:

- Min-NCR does not require a "YES" or "NO" answer.

Theorem 1: The Min-NCR problem, when k is a variable, is an NP-Hard problem.
Proof:
According to Lemma 1, NCR is an NP-Complete decision problem. However, Min-NCR is not a decision problem. It is a search or optimization problem. Therefore, Min-NCR is an NP-hard problem.

Problem description when k is fixed

Our considered network coding based 1+1 protection route design problem and the Steiner tree problem, when k is fixed, are defined in the following.

Network coding based 1+1 protection route design problem with fixed k (NCR1)

Instance:

- A graph $G = (V, E)$, where V is the set of vertices, and E is the set of edges.
- An edge-cost function: $\mathcal{C}$: $E \rightarrow Z^+$, where Z^+ is the set of nonnegative integers.
- A set T of k fixed sources $\in V$.
- A destination $d \in V$

Requirement:

- Determine a set F of edges, forming network coding based 1+1 protection routes for T and d, that minimizes the $\mathcal{C}(F)$ of protection.

Steiner tree problem with fixed k (ST1)

Instance:

- A graph $G' = (V', E')$, where V' is the set of vertices, and E' is the set of edges.
- An edge-cost function: $\mathcal{C}'$: $E \rightarrow Z^+$, where Z^+ is the set of nonnegative integers.
- A set T' of k' fixed terminals $\in V'$

Requirement:

- Determine a subtree F' spanning T whose $\mathcal{C}(F)$ is minimum.

Discussion:

Any instance X of ST1 is solvable in polynomial time with fixed number of terminals [91]. According to our previous discussion, any instance X of ST1 is polynomial time reducible to an instance Y of NCR1. However, from this we can not say that NCR1 is also a polynomial time problem.

References

[1] D. Zhou and S. Subramaniam , **"Survivability in Optical Networks,"** *IEEE Network*, Vol. 14, pp. 16-23, Nov./Dec., 2000. 1, 64, 69, 78, 110

[2] S. Ramamurthy, L. Sahasrabuddhe, R. Mukherjee, **"Survivable WDM Mesh Networks,"** *Journal of Lightwave Technology*, Vol. 21, No. 4, April 2003, pp. 870-883. 1

[3] M. Kodialam and T.V. Lakshman, **"Dynamic Routing of Bandwidth Guaranteed Multicasts with Failure Backup,"** *10th IEEE International Conference on Network Protocols (ICNP'02)*, November 12-15, 2002, Paris, France. 2, 7

[4] M. Kodialam and T.V. Lakshman, **"Dynamic Routing of Bandwidth-Guaranteed Tunnels Using Aggregated Network Resource Usage Information,"** *IEEE/ACM Transactions on Networking*, Vol. 11, No. 3, June 2003, pp. 399-410. 2, 7

[5] O. Hauser, M.S. Kodialam, T.V. Laxman, and W.C. Lau, **"Maintaining Information to Optimize Restorable Dynamic Routing with Shared Backup,"** *United States Patent*, Patent No.: US 6993979 B2, Date: January 31, 2006. 2, 7

[6] M. Clouqueur and W.D. Grover, **"Availability Analysis of Span-Restorable Mesh Networks,"** *IEEE Journal on Selected Areas in Communications*, Vol. 20, No. 4, May 2002, pp. 810-821. 2

[7] K. Kar, M. Kodialam and T.V. Lakshman, **"Routing Restorable Bandwidth Guaranteed Connections Using Maximum 2-Route Flows,"** *IEEE/ACM Transactions on Networking*, Vol. 11, No. 5, October 2003, pp. 772-781. 8

[8] Q. Guo, P.-H. Ho, H.-F. Yu, J. Tapolcai, H. Mouftah, **"Spare capacity reprovisioning for high availability shared backup path protection connections,"** *Computer Communications* 33 (2010) pp. 603-611. 8

[9] P.-H. Ho, J. Tapolcai, H.T. Mouftah, and C.-H. Yeh, **"Linear Formulation for Path Shared Protection,"** *in proc. IEEE International Conference on Communications 2004*, 20-24 June 2004, Paris, France, pp. 1622-1627. 8

[10] T. Stidsen, B. Petersen, S. Spoorendonk, M. Zachariasen, and K. B. Rasmussen, **"Optimal Routing with Failure-Independent Path Protection,"** *NETWORKS*, Vol. 55(2), 2010, pp. 125-137. 8

[11] A. Kodian and W.D. Grover, **"Failure-Independent Path-Protecting p-Cycles: Efficient and Simple Fully Preconnected Optical-Path Protection,"** *Journal of Lightwave Technology*, Vol. 23, No. 10, October 2005, pp. 3241-3259. 8

[12] R. Ahlswede, N. Cai, S.-Y.R. Li, and R.W. Yeung, **"Network Information Flow,"** *IEEE Transactions on Information Theory*, Vol. 46, No. 4, pp. 1204-1216, July 2000. 3, 16, 64, 80, 110, 134

[13] M. Belzner, F. Alexander, and H. Haunstein, **"Performance of Network Coding in Transport Networks with Traffic Protection,"** *2009 ITG Symposium on Photonic Networks*, 4-5 May 2009. 3, 4, 6, 9, 26, 48, 64

[14] A.E. Kamal and O. Al-Kofahi, **"Efficient and Agile 1+N Protection,"** *IEEE Transactions on Communications*, Vol. 59, No. 1, pp. 169-180, January 2011 6, 9, 16, 48, 80, 110

[15] A.E. Kamal, **"1+N Network Protection for Mesh Networks: Network Coding-Based Protection Using p-Cycles,"** *IEEE/ACM Transactions on Networking*, Vol. 18, No. 1, pp. 67-80, February 2010. 6, 9, 48, 80, 110

[16] A.E. Kamal, A. Ramamoorthy, L. Long, and S. Li, **"Overlay Protection Against Link Failures Using Network Coding,"** *IEEE/ACM Transactions on Networking*, Vol. 19, No. 4, August 2011,pp. 1071-1084. 6, 48, 118

[17] S.A. Aly, A.E. Kamal, and O. Al-Kofahi, **"Network protection codes: Providing self-healing in autonomic networks using network coding,"** *Computer Networks* 56(2012), pp. 99-111. 6, 48, 86, 87

[18] I.B. Barla, F. Rambach, D.A. Schupke, and M. Thakur, **"Network Coding for Protection against Multiple Link Failures in Multi-Domain Networks,"** *IEEE International Conference on Communications (ICC)*, 23-27 May 2010. 6, 48

[19] I.B. Barla, F. Rambach, D.A. Schupke, and G. Carle, **"Efficient Protection in Single-Domain Networks using Network Coding,"** *IEEE Global Telecommunications Conference (GLOBECOM 2010)*, 6-10 Dec. 2010. 6, 7, 48

[20] S.E. Rouayheb, A. Sprintson, and C. Georghiades, **"Robust Network Codes for Unicast Connections: A Case Study,"** *IEEE/ACM Transactions on Networking*, Vol. 19, No. 3, June 2011, pp. 644-656. 5, 6, 7, 9, 48, 72, 82, 110, 111

[21] J.J. Shi and J.P. Fonseka, **"Analysis and design of survivable telecommunications networks,"** *IEE Proc. Commun.*, Vol 144, No 5, October 1997. 6

[22] Q. She, X. Huang, and J.P. Jue, **"How Reliable Can Two-Path Protection be?,"** *IEEE/ACM Transactions on Networking*, Vol. 18, No. 3, pp. 922-933, June 2010. 7

[23] E.D. Manley, J.S. Deogun, L. Xu, and D.R. Alexander, **"All-Optical Network Coding,"** *IEEE/OSA Journal of Optical Communications and Networking*, Vol. 2, No. 4, April 2010, pp. 175-191. 7, 48, 80, 81, 110, 111

[24] D. Griffith and S. Lee, **"A 1 + 1 Protection Architecture for Optical Burst Switched Networks,"** *IEEE Journal on Selected Areas in Communications*, Vol. 21, No. 9, November 2003, pp. 1384-1398. 7, 80, 110, 111

[25] D.S. Lun, N. Rantakar, M. Medard, R. Koetter, D.R. Karger, T. Ho, E. Ahmed, and F. Zhao, **"Minimum-Cost Multicast over Coded Packet Networks,"** *IEEE Transactions on Information Theory*, Vol. 52, No. 6, June 2006, pp. 2608-2623. 7, 48

[26] S.S.W. Lee, A. Chen, and Po-Kai Tseng, **"Optimal routing and bandwidth provisioning for survivable IPTV multicasting using network coding,",** *IEEE 2011 Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, 9-12 Jan. 2011, pp. 771 - 775 . 7

[27] B. Guo, H. Li, C. Zhou, and Y. Cheng, **" Analysis of General Network Coding Conditions and Design of a Free-Ride-Oriented Routing Metric,",** *IEEE Transactions on Vehicular Technology*, Vol. 60, No. 4, May 2011. 7

[28] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft, **Xors in the air: practical wireless network coding,** *in Proc. ACM SIGCOMM06*, 2006. 7, 48

[29] S. Sengupta, S. Rayanchu, and S. Banerjee, **"Network Coding-Aware Routing in Wireless Networks,"** *IEEE/ACM Transactions on Networking*, Vol. 18, No. 4, pp. 1158-1170, August 2010. 7, 48

[30] K. Fan, X. Wei, and D. Long, **"A Load-Balanced Route Selection for Network Coding in Wireless Mesh Networks,"** *in the proc. of IEEE International Conference on Communications (ICC 2009)*. 7

[31] X. Zhang and H. Yu, **"Network Coding Aware Joint Routing for Multiple Unicast Flows in Wireless Multihop Networks,"** *4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2008)*, 12-14 Oct. 2008. 7

[32] S. Shabdanov, C. Rosenberg, P. Mitran, **"Joint routing, scheduling, and network coding for wireless multihop networks,"** *in the proceedings of International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, 9-13 May, 2011, pp. 33 - 40. 7

[33] C. Xiong and X. Li, **"Minimmum Cost Optimization of Multicast Wireless Networks with Network Coding,"** *44th Annual Conference on Information Sciences and Systems (CISS)*, 17-19 March 2010, Princeton, NJ. 7, 48

[34] O. Al-Kofahi and A.E. Kamal, **"Network Coding-Based Protection of Many-to-One Wireless Flows,"** *IEEE JSAC* Vol. 27, No. 5, pp. 797-813, 2009. 7

[35] A.H.A. Muktadir, A.A. Jose, and E. Oki, **"An Optimum Mathematical Programming Model for Network-Coding Based Routing with 1+1 Path Protection,"** *in the proceedings of World Telecommunications Congress 2012 (WTC 2012)*, 5-6 March, 2012, Miyazaki, Japan. 64, 72, 82, 110, 111

[36] A.H.A. Muktadir and E. Oki, **"A mathematical model for routing in 1+1 protection with network coding for instantaneous recovery,",** *IEICE Communications Express* 2012, Vol.1, No.6, pp. 228-233. xi, 26, 27

[37] A.A. Jose, A.H.A. Muktadir, and E. Oki, **"Network Coding Aware Instantaneous Recovery Scheme Based On Optimal Traffic Splitting,"** *IEICE Communications Express 2012*, Vol.1, No.1, pp. 28-32. 48, 78, 79, 83, 84, 110, 111, 112, 113

[38] H. Overby, G. Biczk, P. Babarczi, and J. Tapolcai, **"Cost Comparison of 1+1 Path Protection Schemes: A Case for Coding,"** in *Proc. IEEE ICC 2012*, pp. 3067-3072, 10-25 June, 2012. 64

[39] P. Babarczi, J. Tapolcai, J., P.H. Ho, and M. Medard, **"Optimal Dedicated Protection Approach to Shared Risk Link Group Failures using Network Coding,"** *in Proc. IEEE ICC 2012*, pp. 3084-3088, 10-25 June, 2012.

[40] A.H.A. Muktadir and E. Oki, **"Network Coding Based Cost-Efficient 1+1 Protection in Scenarios with Two Sources and Different Destinations for Instantaneous Recovery,"** *in Proc. 10th International Conference on the Optical Internet (COIN 2012)*, May 29-31, 2012, Yokohoma, Japan.

[41] E. Oki, **Linear Programming and Algorithms for Communication Networks: A Practical Guide to Network Design, Control, and Management**, *CRC Press*, Boca Raton, 2012. 23

[42] S. Homer and A.l. Selman, **Computability and Complexity Theory (Texts in Computer Science)** *2nd Ed., Springer* 2011.

[43] R. Bhandari, **"Optimal Physical Divesity Algorithms and Survivable Networks,"** *in proc. of the 2nd IEEE Symposium on Computers and Communications (ISCC '97)*, pp. 433-441, 1997. 31

[44] R. Bhandari, **Survivable Networks: Algorithms for Diverse Routing**, *New York: Springer*, 1999. 19, 26, 31, 35, 69, 138

[45] S. Harris and J. Ross, **Beginning Algorithms,** *Wiley Publishing, Inc.* 2006. 35, 119

[46] CPLEX, [Online] **http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/,** *2013.* 24, 38, 59, 69, 94, 119

[47] Guidelines for estimating CPLEX memory requirements based on problem size, **[Online] http://www-01.ibm.com/support/docview.wss?uid=swg21399933.** 27

[48] A.L. Barabasi and R. Albert, **Emergence of Scaling in Random Networks,** *Science*, pp. 509-512, October 1999. 38

[49] BRITE (Boston University Representative Internet Topology Generator), **http://www.cs.bu.edu/brite/.** *visited* April 2014. 38, 94

[50] R.M. Karp, **"Reducibility Among Combinatorial Problems,"** *Complexity of Computer Computations*, New York: Plenum. pp. 85103, 1972.

[51] Zhu, H. Zang, and B. Mukherjee, **"Exploiting the Benefits of Virtual Concatenation Technique to the Optical Transport Networks,"** *in Proc. Optical Fiber Communication (OFC) 03,* Anaheim, CA, Mar. 2003. 78, 110

[52] J. Kuri, M. Gagnaire, and N. Puech, **"On the Resource Efficiency of Virtual Concatenation in SONET/SDH Mesh Transport Networks Bearing Protected Scheduled Connections,"** *J. Lightw. Technol.*, Vol. 23, pp. 104-112, May 2002. 78, 110

[53] C. Ou, L. Sahasabuddle, K. Zhu, C. Martel and B. Mukherjee, **"Survivable Virtual Concatenation for Data Over SONET/SDH in Optical Transport Networks,"** *IEEE/ACM Trans. Netw.*, vol. 14 , pp. 218-231, Feb. 2006. 78, 110

[54] T. Matsuda, T. Noguchi, and T. Takine, **"Survey of Network Coding and Its Applications,"** *IEICE Trans. Commun.*, vol. E94-B, no. 3, pp. 698-717, Mar. 2011. 5, 80, 81, 83, 110, 111, 113

[55] P. Babarczi, G. Biczok, H. verby, J. Tapolcai, and P. Soproni, **"Realization Strategies of Dedicated Path Protection: A Bandwidth Cost Perspective,"** *Comp. Netw.*, vol. 57, no. 9, pp. 1974-1990, Jun. 2013. 111

[56] S.N. Avci and E. Ayanoglu, **"Optimal Algorithms for Near-Hitless Network Restoration via Diversity Coding,"** *IEEE Trans. Commun.*, vol. 61, no. 9, pp. 3878-3893, Sep. 2013. 111

[57] C. Meusburger, and D.C. Schupke, **"Rerouting of 1+1 Optical Channels: Cost Benefits for Incrementally Planned Networks,"** *in Proc. OSA/OFC/NFOEC Conf.*, 2009. 81, 111

[58] M.G. Luby, M. Mitzenmacher, M.A. Shokrollahi, and D.A. Spielman, **"Efficient Erasure Correcting Codes,"** *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 569-584, Feb. 2001. 81, 111

[59] T. Matsuda and T. Takine, **"Multicast communications with Reed Solomon/network joint coding in wireless multihop networks,"** *J. Commun.*, vol.4, no.11, pp.856864, Dec. 2009. 81, 83, 111, 113

[60] D. Hisano, A. Maruta, and K. Kitayama, **"Demonstration of All-Optical Network Coding by using SOA-MZI based XOR Gates,"** *Proc. OFC/NFOEC Conf.* 2013. 81, 111

[61] Y. An, F.D. Ros, and C. Peucheret, **"All-Optical Network Coding for DPSK signals,"** *Proc. OFC/NFOEC Conf.*, 2013. 81, 111

[62] J.H. Kim, Y.M. Jhon, Y.T. Byun, S. Lee, D.H. Woo, and S.H. Kim, **"All-Optical XOR Gate Using Semiconductor Optical Amplifiers Without Additional Input Beam,"** *IEEE Photon. Technol. Lett.*, vol. 12, no. 10, pp. 1436-1438, Oct. 2002. 81, 111

[63] Q. Wang, G. Zhu, H. Chen, J. Jaques, J. Leuthold, A.B. Piccirilli, and N.K. Dutta, **"Study of All-optical XOR Using Mach-Zehnder Interferometer and Differential Scheme,"** *IEEE J. Quantum Electron.*, vol. 40, no. 6, pp. 703-710, Jun. 2004. 81, 111

[64] S. Harris and J. Ross, **Beginning Algorithms**, *Wiley Publishing, Inc.* 2006. 35, 119

[65] S. Harris and J. Ross, **Beginning Algorithms**, *Wiley Publishing, Inc.* 2006. 35, 119

[66] A.H.A. Muktadir and E. Oki, **"Optimum Route Design in 1+1 Protection with Network Coding for Instantaneous Recovery,"** *IEICE Trans. Commun.*, vol. E97-B, no. 1, pp. 87-104, Jan. 2014. 48, 50, 55, 56, 64

[67] [online], http://sndlib.zib.de, 2013. 119

[68] C. Ou, L. Sahasabuddle, K. Zhu, C. Martel and B. Mukherjee, **"Survivable Virtual Concatenation for Data Over SONET/SDH in Optical Transport Networks,"** *IEEE/ACM Trans. Netw.*, vol. 14 , pp. 218-231, Feb. 2006. 78, 110

[69] B. Mukherjee, **Optical WDM Networks**, *Springer*, 2006. 78

[70] S. Yao, F. Xue, B. Mukherjee, S.J.B. Yoo, and S. Dixit, **"Electrical Ingress Buffering and Traffic Aggregation for Optical Packet Switching and Their Effect on TCP-Level Performance in Optical Mesh Networks,"** *IEEE Commun. Mag.*, pp. 66-72, Sep. 2002. 81

[71] J.K. Rhee, C. Lee, J. Kim, Y. Won, J.S. Choi, and J. Choi, **"Power and Cost Reduction by Hybrid Optical Packet Switching**

with Shared Memory Buffering," *IEEE Commun. Mag.*, pp. 102-110, May 2011. 81, 105

[72] H.J.S. Dorren, M.T. Hill, Y. Liu, N. Calabretta, A. Srivatsa, F. M. Huijskens, H. de Waardt, and G. D. Khoe, **"Optical Packet Switching and Buffering by Using All-Optical Signal Processing Methods,"** *J. Lightw. Technol.*, vol. 21, no. 1, pp. 2-12, Jan. 2003. 81

[73] N. Bouabdallah, H. Perros, and G. Pujolle, **"A Cost-Effective Traffic Aggregation Scheme in All-Optical Networks,"** *in Proc. IEEE Global Telecommun. Conf. (GLOBECOM 2005)*, 28 Nov. - 2 Dec., 2005, St. Louis, MO, USA. 81

[74] I. Cidon, R. Rom, and Y. Shavitt, **"Analysis of Multi-path Routing,"** *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 885-896, Dec. 1999. 81

[75] S. Ahuja, T. Korkmarz, and M. Krunz, **"Minimizing the Differential Delay for Virtually Concatenated Ethernet over SONET Systems,"** *in Proc. ICCCN Conf.*, pp. 205-210, 2004. 81

[76] S. Huang, B. Mukherjee, and C. Martel, **"Survivable Multipath Provisioning for High-Capacity Backbone with Differential Delay Aware Constraint in Telecom Mesh Networks,"** *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 657-669, Jun. 2011. 81

[77] A. Srivastava, S. Achaya, M. Alicherry, B. Gupta, and P. Risbood, **"Differential Delay Aware Routing for Ethernet over**

SONET/SDH," *in Proc. IEEE INFOCOM Conf.*, pp. 1117-1127, Miami, FL, 2005. 81, 82, 85

[78] J. Mocerino, **"Carrier Class Ethernet Service Delivery Migrating SONET to IP and Triple Plays Offerings,"** *in Proc. Optical Fiber Communications (OFC) 06,* Anaheim, CA, Mar. 2006.

[79] R. Andersen, F. Chung, A. Sen, and G. Xue, **"On Disjoint Path Pairs with Wavelength Continuity Constraint in WDM Networks,"** *Proc. 23rd Annu. Joint Conf. of the IEEE Comput. and Commun. Societies, INFOCOM 2004,* pp. 524-535, 7-11 Mar., 2004. 82

[80] S. A. Aly and A. E. Kamal, **"Network Protection Codes Against Link Failures Using Network Coding,"** *IEEE Global Telecommun Conf. (GLOBECOM 2008)*, 30 Nov. - 4 Dec 2008, New Orleans, LA, USA. 86

[81] W. C. Huffman and V. Pless, **Fundamentals of Error-Correcting Codes**, *Cambridge University Press*, Cambridge, 2003. 87

[82] E. Fujiwara, **Code design for dependable systems : theory and practical applications**, *Wiley-Interscience*, 2006. 87

[83] E. Oki, T. Takeda, JL. Le Roux, and A. Farrel, **"Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering,"** *RFC 5623*, Sep. 2009. 102

[84] Y. Lee, JL. Le Roux, D. King, and E. Oki, **"Path Computation Element Communication Protocol (PCEP) Requirements and Pro-**

tocol Extensions in Support of Global Concurrent Optimization," *RFC 5557*, Jul. 2009. 102

[85] A. Farrel, J.-P. Vasseur,and J. Ash, **"A Path Computation Element (PCE)-Based Architecture,"** *RFC 4655*, Aug. 2006. 102

[86] NEC and Corning achieve petabit optical transmission, [online], http://optics.org/news/4/1/29, 22 Jan., 2013. 105

[87] T. Biermann, M. Draxler, and H. Karl, **Flow Synchronization for Network Coding,** *Journal of Communications*, vol. 4, no. 11, Dec. 2009. 72

[88] The GLPK (GNU Linear Programming Kit) package [Online], http://www.gnu.org/software/glpk/, 2012. 75

[89] J. Kleinberg and E. Tardos, **Algorithm Design,** *Pearson/Addison-Wesley*, 2006. 133

[90] M.R. Garey and D.S. Johnson, **COMPUTERS AND INTRACTABILITY: A Guide to the Theory of NP-Completeness,** *Macmillan Higher Education*, 1979. 138

[91] B. Kimelfeld and Y. Sagiv, **"New algorithms for computing Steiner trees for a fixed number of terminals,"** unpublished manuscript, available online, http://researcher.ibm.com/researcher/files/us-kimelfeld/papers-steiner06.pdf. 140

Publications

List of Publications related to the dissertation

Journal Papers

1. A.H.A. Muktadir and E. Oki, "Optimum Route Design in 1+1 Protection with Network Coding for Instantaneous Recovery, IEICE Trans. Commun., Vol. E97-B, No.01, pp. 87-104, Jan. 2014. (related to Chapters 2, 3)

2. A.H.A. Muktadir, P.V. Phong, and E. Oki, "A mathematical model for network coding aware optimum routing in 1+1 protection for instantaneous recovery with relaxing destinations node degree constraint, IEICE Commun. Express, Vol. 2, No. 12, pp. 512-517, Dec. 2013. (letter) (related to Chapter 5)

3. A.H.A. Muktadir and E. Oki, "A Mathematical Model for Routing in 1+1 Protection with Network Coding for Instantaneous Recovery," IEICE Commun. Express, Vol. 1, No. 6, pp. 228-233, Nov. 2012. (letter) (related to Chapter 2)

International Conference Papers

1. A.H.A. Muktadir and E. Oki, "A Heuristic Routing Algorithm for Network Coding Aware 1+1 Protection Route Design for Instantaneous Recovery," The IEEE 15th International Conference on High Performance Switch-

ing and Routing (HPSR 2014), pp. 84-89, Vancouver, British Columbia, Canada 1-4 Jul., 2014. (related to Chapter 4)

2. A.H.A. Muktadir and E. Oki, "A Heuristic Routing Algorithm with Erasure Correcting Code Based Instantaneous Recovery Technique," The IEEE International Conference on Communications (ICC 2014), pp. 1149-1153, Sydney, Australia, 10 - 14 Jun., 2014. (related to Chapter 8)

3. A.H.A. Muktadir, A.A. Jose, and E. Oki, "Mathematical Programming Model for Network-Coding Based Routing with 1+1 Path Protection, World Telecommunications Congress 2012, Mar. 2012. (related to Chapter 2)

National Conference Papers

1. A.H.A. Muktadir and E. Oki, "Routing with Traffic Splitting based on Erasure Correcting Code for Instantaneous Recovery," IEICE Tech. Rep., vol. 113, no. 393, PN2013-57, pp. 123-128, Jan. 2014. (related to Chapter 8)

List of Other Publications

Journal Papers

1. S. Tsunoda, A.H.A. Muktadir, and E. Oki, "Load-Balanced Non-Split Shortest-Path-Based Routing with Hose Model and Its Demonstration," IEICE Trans. Commun., Vol. E96-B, No. 05, May 2013.

2. A.A. Jose, A.H.A. Muktadir, and E. Oki, "Network Coding Aware Instantaneous Recovery Scheme Based on Optimal Traffic Splitting, IEICE Communications Express, vol. 1, no. 1, pp. 28-32, Jun. 2012. (letter)

International Conference Papers

1. E. Oki and A.H.A. Muktadir, "A Routing Algorithm for 1+1 Protection Considering Network Coding Effect with Polynomial-Time Computation, International Workshop on Optical Networking (iWON 2013), Atlanta, USA, Dec. 9, 2013. (Invited Paper)

2. P.V. Phong, A.H.A. Muktadir, and E. Oki, "A Mathematical Model for Network Coding Aware Optimal Routing in 1+1 Protection for Destination's Node Degree ≥ 2," The 18th OptoElectronics and Communications Conference / Photonics in Switching 2013 - OECC/PS, Kyoto, Japan, 30 Jun. - 4 Jul., 2013.

3. A.A. Jose, A.H.A. Muktadir, and E. Oki, "Network Coding Aware Instantaneous Recovery Scheme With Differential Delay Constraints Based on Optimal Traffic Splitting, The 7th Triangle Symposium on Advanced ICT (TriSAI 2012), Sep. 2012.

4. A.H.A. Muktadir and E. Oki, "Network Coding Based Cost-Efficient 1+1 Protection in Scenarios with Two Sources and Different Destinations for Instantaneous Recovery, 10th International Conference on the Optical Internet (COIN 2012), May 2012.

5. A.A. Jose, A.H.A. Muktadir, and E. Oki, "Instantaneous Recovery Scheme with Optimal Traffic Splitting Using Network Coding, 10th International Conference on the Optical Internet (COIN 2012), May 2012.

6. S. Tsunoda, A.H.A. Muktadir, and E. Oki, "Performance Evaluation of Non-Split S-OSPF with Hose Model," 10th International Conference on the Optical Internet (COIN 2012), May 2012.

7. S. Tsunoda, A.H.A. Muktadir, and E. Oki, "Implementation of Load-Balanced Non-Split Shortest-Path-Based Routing with Hose Model," The 6th Triangle Symposium on Advanced ICT (TriSAI 2011), Aug. 2011.

8. S. Tsunoda, A.H.A. Muktadir, and E. Oki, “Load-Balanced Shortest-Path-Based Routing Without Traffic Splitting in Hose Model, IEEE International Conference on Communications (ICC 2011), 5-9 Jun., 2011, Kyoto, Japan.

9. A.H.A. Muktadir, M.A. Rahman and S.M.H.S. Iqbal, “Analytical Comparison of AODV and DSDV Routing Protocols Over MANET”, 9th International Conference on Computer and Information Technology (ICCIT) 2006, Bangladesh.

10. M.A. Rahman, A.H.A. Muktadir, “The Impact of Data Send Rate, Node Velocity and Transmission Range on QoS Parameters of OLSR and DYMO MANET Routing Protocols”, 10th International Conference on Computer and Information Technology (ICCIT) 2007, Bangladesh.

11. A.H.A. Muktadir, M.A. Rahman and M.K. Hasan, “Energy Consumption Study Of OLSR And DYMO MANET Routing Protocols In Military And Rescue Operation, International Conference on Electronics, Computer and Communication, 27-29 June, 2008, University of Rajshahi, Bangladesh.