KARADENİZ TECHNICAL UNIVERSITY
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COMPUTER ENGINEERING GRADUATE PROGRAM

TOPOLOGY-AWARE FAST RE-ROUTE ALGORITHMS

FOR FAULT TOLERANT NETWORKING

Ph. D. THESIS

Selçuk CEVHER, M.Sc.

JUNE 2016
TRABZON

KARADENİZ TECHNICAL UNIVERSITY

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

COMPUTER ENGINEERING GRADUATE PROGRAM

TOPOLOGY-AWARE FAST RE-ROUTE ALGORITHMS FOR FAULT TOLERANT NETWORKING

Selçuk CEVHER, M. Sc.

This thesis is accepted to give the degree of

DOCTOR OF PHILOSOPHY

By

The Graduate School of Natural and Applied Sciences at
Karadeniz Technical University

The Date of Submission : 20 /05 /2016
The Date of Examination : 20 /06 /2016

Thesis Supervisor : Assoc. Prof. Mustafa ULUTAŞ
Thesis Co-supervisor : Dr. İbrahim HÖKELEK

Trabzon 2016

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Computer Engineering Graduate Program

The thesis entitled:

TOPOLOGY-AWARE FAST RE-ROUTE ALGORITHMS FOR FAULT TOLERANT NETWORKING

Prepared by Selçuk Cevher

has been accepted as a thesis of

DOCTOR OF PHILOSOPHY

after his successful defense before the jury assigned by the Administrative Board of the Graduate School of Natural and Applied Sciences with the decision number 1654 dated May 24, 2016.

Examining Committee Members

Chair : Prof. Dr. Reha CİVANLAR ....

Member : Prof. Dr. Hakan Ali ÇIRPAN ....

Member : Prof. Dr. Cemal KÖSE ....

Member : Assoc. Prof. Dr. Mustafa ULUTAŞ ....

Member : Assoc. Prof. Dr. İsmail KAYA ....

Prof. Dr. Sadettin KORKMAZ
Director

FOREWORD

In this thesis, topology-aware fast re-route algorithms which provide fast recovery from network failures are proposed for fault tolerant networking. I would like to express my sincere gratitude to my advisor, Assoc. Prof. Mustafa Ulutaş for his guidance throughout the development of this thesis. I am thankful to him for his constructive comments and suggestions since I would not be able to concentrate on this research without his refreshing attitude. I am also deeply grateful to my co-advisor Dr. İbrahim Hökelek for his invaluable contributions to this thesis. He is the one who introduced me to the field of fault tolerant networking. I was fortunate to work with him for the past years in which he provided me valuable knowledge. Without his expertise in the field, dedication, and sacrifices, this work could not have been done. I also would like to thank Prof. Cemal KÖSE and Assoc. Prof. İsmail KAYA for their valuable comments on this thesis.

I am deeply grateful to my former advisor Prof. M. Ümit Uyar who introduced me to the field of network management. I express my special thanks to him for his encouragement and expertise during my graduate studies in the City University of New York. I would like to thank Mr. Sunil Samtani for his support during my internship studies at Telcordia Technologies, Inc., currently part of Ericsson, without which I would not be able to concentrate on this research.

I would like to express my special thanks to my elder Mr. Mus'ab Koçak. Without his wisdom and inspiration, I could not concentrate on this research during my graduate studies in the City University of New York. It is a great pleasure for me to mention that this work could not have been accomplished without the patience and support of my wife Elif. She always supported me in my hard times, and supplied me with the necessary stamina to complete this thesis. I am grateful to my little sons Yusuf and Yahya since they agreed to share their father with my graduate studies. I also would like to express my sincere thanks to my father Ömer, my mother Necla, my brothers Mahir and İbrahim, and my sister Azize. Without their support, this thesis could not have been accomplished. I dedicate this thesis to them.

Selçuk CEVHER
Trabzon 2016

STATEMENT OF ETHICS

I declare that all the parts of this PhD thesis entitled “Topology-Aware Fast Re-Route Algorithms for Fault Tolerant Networking” have been accomplished under the supervision of my advisor Assoc. Prof. Mustafa Ulutaş and co-advisor Dr. İbrahim Hökelek. I observed the highest ethical and scientific standards during the development of this thesis. I maintained the highest integrity at all times regarding the data gathering and experimentation. I avoided plagiarism and fully acknowledged the work of others to which I have referred in this thesis. I admit all the legal consequences if the falsehood of this declaration is proven. 20/06/2016

Selçuk CEVHER

TABLE OF CONTENTS

Page

FOREWORD III

STATEMENT OF ETHICS IV

TABLE OF CONTENTS V

SUMMARY VIII

ÖZET IX

LIST OF FIGURES XI

LIST OF TABLES XIII

LIST OF SYMBOLS XV

LIST OF ABBREVIATIONS XVII

1. INTRODUCTION 1

1.1. Research Objectives 6

1.2. Contributions 6

2. LITERATURE REVIEW 10

2.1. Topology-Aware Networking 10

2.2. Network Resilience Through Multi Topology Routing 11

2.3. Failure Recovery in Software Defined Networks 13

3. MULTI TOPOLOGY ROUTING BASED IPFRR MECHANISMS 14

3.1. Multiple Routing Configurations (MRC) 14

3.1.1. Overview 14

3.1.2. Example Operation 16

3.2. Maximally Redundant Trees (MRT) 19

3.2.1. Overview 19

3.2.2. Example Operation 20

3.3. Experimental Results 22

3.3.1. Performance Comparison of the MRC and the MRT 23

3.3.2. Performance Evaluation of the MRC Using Realistic ISP Topologies 26
4. AUTOMATED TOPOLOGICAL ANALYSIS OF MULTIPLE ROUTING CONFIGURATIONS 31
4.1. Topological Analysis of Multiple Routing Configurations 32
4.1.1. Our Workflow 32
4.1.2. Topological Characteristics In Our Workflow 33
4.1.3. Topology Generation Models In Our Workflow 35
4.2. Experimental Results 37
5. MULTIPLE ROUTING CONFIGURATIONS WITH NON-RANDOM NODE ISOLATION 46
5.1. Analysis of the Impact of Node Isolation Order 46
5.1.1. Example Operations Showing the Impact of Q_n 47
5.1.2. Graph Theoretical Analysis 49
5.2. Enhanced MRC Algorithm 54
5.2.1. Problem Statement 54
5.2.2. Algorithm 54
5.2.3. Complexity and Convergency Analysis 56
5.3. Experimental Results 57
5.3.1. Experimental Setup 57
5.3.2. Evaluation of Minimum VT Requirement 59
5.3.3. Evaluation of Partitioning Based Failures 65
5.3.4. Evaluation of Alternate Path Lengths 65
6. TOPOLOGY-AWARE MULTIPLE ROUTING CONFIGURATIONS 67
6.1. Topology-Aware MRC Algorithms 68
6.1.1. mMRC-1 68
6.1.1.1. Description 68
6.1.1.2. Convergence 72
6.1.1.3. Complexity 73
6.1.2. mMRC-2 74
6.1.2.1. Description 74

6.1.2.2. Convergence and Complexity 76
6.2. Experimental Results 76
6.2.1. Evaluation of Minimum VT Requirement 77
6.2.2. Effect of Isolation Order On Heterogeneity 83
6.2.3. Evaluation of Alternate Path Lengths 85
7. MULTI TOPOLOGY ROUTING BASED IPFRR FOR SOFTWARE DEFINED NETWORKS 86
7.1. Software Defined Networks 87
7.2. Architecture for MT-IPFRR in SDN 88
7.3. Experimental Results 92
8. DISCUSSION 96
9. CONCLUSIONS 99
10. RECOMMENDATIONS 102
11. REFERENCES 103
CURRICULUM VITAE

PhD. Thesis

SUMMARY

TOPOLOGY-AWARE FAST RE-ROUTE ALGORITHMS FOR FAULT TOLERANT NETWORKING

Selçuk CEVHER

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Computer Engineering Graduate Program
Advisor: Assoc. Prof. Mustafa ULUTAŞ
Co-Advisor: Dr. İbrahim HÖKELEK
2016, 109 Pages

Real-time services in communication networks require a fault tolerant data transmission to support their stringent quality of service requirements. The Internet Engineering Task Force has been working on standardizing IP Fast Re-Route (IPFRR) technologies with a full failure coverage which provide seamless forwarding of IP packets during network failures. Multi Topology Routing based IPFRR (MT-IPFRR) technologies use virtual topologies (VTs) to compute alternate routing tables to recover from failures. In this thesis, we compare the performances of the MT-IPFRR mechanisms, namely, Multiple Routing Configurations (MRC) and Maximally Redundant Trees (MRT). The results show that the alternate path lengths of the MRC are scalable with respect to the network size and density as opposed to the case for the MRT. We also provide an elaborate topological dependency analysis of the MRC. Using our automated topological analysis tool, we discover a significant correlation between the performance of the MRC and the heterogeneity level of a topology, namely, the tendency to have hub nodes, whose degree is much higher than the rest of the network. Inspired by our topological analysis results, we propose heuristic algorithms to reduce the number of VTs used by the MRC to decrease its operational complexity. Finally, we propose a new MT-IPFRR technique leading to self-recovering Software Defined Networks.

Keywords: IP fast re-route, Multi topology routing, Multiple routing configurations, Topological analysis, Software defined networks

Doktora Tezi

ÖZET

HATA TOLERANSLI HABERLEŞME İÇİN TOPOLOJİ FARKINDA HIZLI YENİDEN YÖNLENDİRME ALGORİTMALARI

Selçuk CEVHER

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Doç. Dr. Mustafa ULUTAŞ
İkinci Danışman: Dr. İbrahim HÖKELEK
2016, 109 Sayfa

Haberleşme ağlarındaki gerçek zamanlı servisler, katı servis kalitesi gereksinimlerinin karşılanabilmesi için hata toleranslı bir veri iletimini gerektirirler. İnternet Mühendisliği Görev Gücü (Internet Engineering Task Force), ağ hataları süresince IP paketlerinin kesintisiz iletimini sağlayan tam hata kapsamasına sahip IP Hızlı Yeniden Yönlendirme (IP Fast Re-Route - IPFRR) teknolojilerinin standartlaştırılması üzerinde çalışmaktadır. Çoklu Topoloji Yönlendirmesi tabanlı IPFRR (Multi Topology Routing based IPFRR -MT-IPFRR) teknolojileri, arızalanması muhtemel olan ağ bileşenlerinden sakınan alternatif yönlendirme tablolarının hesaplanması için sanal topolojiler (ST) kullanırlar. Bu tezde, MT-IPFRR mekanizmalarından olan Çoklu Yönlendirme Konfigürasyonlarının (Multiple Routing Configurations - MRC) ve Azami Fazlalık Ağaçlarının (Maximally Redundant Trees - MRT) performanslarını kıyaslamaktayız. Sonuçlar, MRC ’ye ait alternatif güzergah uzunlukları ağdaki düğüm sayısı ve link yoğunluğu arttıkça sadece çok az değişirken MRT’ye ait alternatif güzergah uzunluklarının ölçeklenebilir olmadığını göstermektedir. Bunun, IP ağlardaki düğümler arasındaki erişilebilirliği iyileştirmek için kullanılacak MT-IPFRR mekanizmasının seçimine kılavuzluk edecek önemli bir ölçeklenebilirlik sonucu olduğuna inanmaktayız. Ayrıca, MRC ’nin ayrıntılı bir topolojik bağımlılık analizini sunmaktayız. Otomatik topolojik analiz aracımızı kullanarak, MRC ’nin performansı ile topolojik karakteristikler arasında önemli bir bağımlılığın bulunduğunu tespit etmekteyiz. MRC, bir ağ topolojisinin içerdiği, derecesi ağdaki diğer düğümlerin derecelerinden çok daha büyük olan merkezi (hub) düğümlerin sayısının artması durumunda, tam alternatif güzergah kapsaması sağlayabilmesi için daha fazla sayıda ST oluşturmaya ihtiyaç duymaktadır. Topolojik analiz sonuçlarımızdan esinlenerek, MRC ’nin operasyonel karmaşıklığının iyileştirilmesi için MRC ’nin kullandığı ST sayısını azaltacak sezgisel algoritmalar önermekteyiz. Sayısal deneylerimiz, sezgisel algoritmalarımızın, topolojik karakteristiklerin sistematik analizinin etkinliğini doğrulayarak, MRC ’nin performansını belirgin bir şekilde iyileştirdiğini göstermektedir. Ayrıca, kendi kendini onaran hata toleranslı Yazılım Tabanlı Ağların oluşturulabilmesini hedefleyen yeni bir MT-IPFRR yöntemi önermekteyiz. Deneysel sonuçlarımız, yaklaşımımızın, tepkisel (reactive) onarım sürecine kıyasla hata onarım süresini önemli ölçüde azalttığını göstermektedir.

Anahtar Kelimeler: IP hızlı yeniden yönlendirme, Çoklu topoloji yönlendirmesi, Çoklu yönlendirme konfigürasyonları, Topolojik analiz, Yazılım tabanlı ağlar

LIST OF FIGURES

Page

Figure 1.1 VTs computed for an example physical topology 3

Figure 3.1 Initial structures of VTs 16

Figure 3.2 Structures of VTs after $R9$, $R6$ and $R3$ become isolated 17

Figure 3.3 Structures of VTs after $R2$, $R8$ and $R1$ become isolated 18

Figure 3.4 Structures of VTs after $R4$, $R7$ and $R5$ become isolated 18

Figure 3.5 Red and blue trees calculated for an example topology 20

Figure 3.6 DFS and low-point numbers for an example N2C topology 21

Figure 3.7 GADAG computed for the N2C topology in Fig. 3.6 21

Figure 3.8 Modified ADAG and BFS trees for red/blue routing tables 22

Figure 3.9 Success rates of the MRC for the construction of 3 VTs 26

Figure 3.10 Comparison of the alternate path lengths of the MRC and the MRT 27

Figure 3.11 Minimum VT requirement of the MRC and and its success rates for realistic ISP-level topologies 29

Figure 4.1 Workflow of our topological analysis 32

Figure 4.2 An example topology to analyze the topological characteristics 34

Figure 4.3 Analysis of ave_{minVT} of the MRC with respect to n (**BA:** m=2 **GLP:** m=2, p=0.2, β=0.6 **BA2:** m=2, p=0.1, q=0 **Waxman:** m=2, α=0.15, β=0.2) 39

Figure 4.4 Analysis of the topological characteristics with respect to n (**BA:** m=2 **GLP:** m=2, p=0.2, β=0.6 **BA2:** m=2, p=0.1, q=0 **Waxman:** m=2, α=0.15, β=0.2).. 40

Figure 4.5 Analysis of ave_{minVT} of the MRC with respect to m (**BA:** n=300 **GLP:** n=300, p=0.2, β=0.6 **BA2:** n=300, p=0.001, q=0 **Waxman:** n=300, α=0.15, β=0.2) 41

Figure 4.6 Analysis of the topological characteristics with respect to m (**BA:** n=300 **GLP:** n=300, p=0.2, β=0.6 **BA2:** n=300, p=0.001, q=0 **Waxman:** n=300, α=0.15, β=0.2)... 42

Figure 4.7 Example distributions for H and $minVT$ 44

Figure 5.1 Impact of the Q_n on the successful termination of the MRC 48

Figure 5.2 An example highly heterogeneous topology 50

Figure 5.3 The structures of the backbones of VT_1 and VT_2 after two and four iterations 51

Figure 5.4 Four possible cases for the isolation of u_0 within the first four iterations ... 52

Figure 5.5 Iteration distribution for the node degrees 64

Figure 6.1 Tie breaking mechanisms of mMRC-1 and mMRC-2 70

Figure 6.2 Example VTs constructed by mMRC-1 72

Figure 6.3 Example VTs constructed by mMRC-2 75

Figure 6.4 $minVT$ distribution for $G^{GLP}_{m=2}$ subset of P_4 79

Figure 6.5 The correlation between ave_H and IP values 80

Figure 6.6 Distributions of isolated node degrees and heterogeneity of VTs 84

Figure 7.1 SDN architecture incorporating MT-IPFRR 88

Figure 7.2 State machine to perform MT-IPFRR in SDN 89

Figure 7.3 An example scenario for the restoration method 91

Figure 7.4 Example virtual topologies 91

Figure 7.5 Experimental setup 92

Figure 7.6 Traffic captured at the controller during the failure recovery process 93

LIST OF TABLES

Page

Table 1.1 Tasks to be performed by MT-IPFRR without a human in the loop 4

Table 3.1 Comparison results for the MRC and the MRT 25

Table 3.2 BRITE parameters used to generate realistic ISP-level topologies 27

Table 3.3 RAPL values of the MRC for realistic ISP-level topologies 28

Table 4.1 Parameters used to generate our topology pool 37

Table 4.2 Topological analysis of ave_{minVT} of the MRC with respect to β parameter of GLP for $n = 300$ 43

Table 4.3 Comparison of partitioning based failures of the MRC ($n = 200$, $m = 2$) ... 45

Table 5.1 Different topology pools used in our experiments 57

Table 5.2 ave_{minVT} and its confidence intervals with 95% confidence with respect to n ($m = 2$) 60

Table 5.3 ave_{minVT} and its confidence intervals with 95% confidence with respect to m ($n = 300$) 62

Table 5.4 Evaluation of minimum VT requirement using real networks 64

Table 5.5 Comparison of the number of partitioning based failures for the GLP topologies ($n = 200$, $m = 2$) 65

Table 5.6 Alternate path lengths for the GLP topologies 66

Table 6.1 Notation 68

Table 6.2 Operation of mMRC-1 on an example network 72

Table 6.3 Operation of mMRC-2 on an example network 75

Table 6.4 Average minimum VT requirement and its confidence intervals with 95% confidence for topology pools with varying m 78

Table 6.5 Average minimum VT requirement and its confidence intervals with 95% confidence for topology pools with varying n 81

Table 6.6 The comparison of minimum VT requirement using real networks 83

Table 6.7 RAPL values computed using P_3 85

Table 7.1 States versus actions 89

Table 7.2 Performance comparison of MT-IPFRR with the reactive recovery 94

LIST OF SYMBOLS

w_r	Weight of a restricted link
V	Set of nodes
Q_n	Node isolation queue
n	Number of nodes
$minVT$	Minimum VT requirement
m	Number of links per new node
I_i	Set of isolated nodes
d	Node degree distribution
d_{max}	Maximum node degree
σ_d	Standard deviation
μ_d	Mean node degree
δ	Network density
ϕ_u	Number of actual links interconnecting u 's neighbors
ω_u	Number of all possible links interconnecting u 's neighbors
$P(x,y)$	Probability function for the link existence between the nodes x and y
α	Model-specific constant in Waxman model
β	Model-specific constant in Waxman and GLP models
D	Euclidean distance in Waxman model
L	Maximum distance among nodes in Waxman model
p	Probability of adding a new link in BA2 and GLP models
q	Probability of rewiring in BA2 model
ave_H	Average heterogeneity value
ave_C	Average centralization value
ave_{CC}	Average clustering coefficient value
ave_{minVT}	Average minimum VT requirement
N_{N2C}	Number of non-2-connected topologies
F_{par}	Number of partitioning based failures in MRC
λ	Number of nodes on outer cycle

k_{opt}	Optimum cardinality for the set of VTs
E	Set of edges
$N(l)$	Remote end-point of the link l
$D(v)$	Degree of node v
$A_{E(u)}$	Array of neighbouring links of node u
$\mathscr{O}$	Asymptotic upper bound
$G^{M}_{n=k}$	Set of topologies with $n = k$ generated using model M
$G^{M}_{m=j}$	Set of topologies with $m = j$ generated using model M
t	Appropriate percentage point for Student's t-distribution
G	Graph of the physical topology
k	Pre-configured number of VTs to be constructed
A_V	Array of ordered nodes
$E(u)$	Set of neighbouring links of node u
A_{VT}	Array of ordered VTs
$\mathbb{N}(u)$	Set of neighbouring nodes of node u
$D_{max}(\mathbb{N}(u))$	Maximum neighbour degree within the neighbourhood of node u
$B(VT_i)$	Backbone of VT_i
$D_{B(VT_i)}(u)$	Degree of node u in $B(VT_i)$
$\Delta H(B(VT_i)\backslash u)$	Change in the heterogeneity level of $B(VT_i)$ if u is isolated in VT_i
Ω	Failure recovery time
ρ	Capture time of the first datagram
γ	Capture time of the last datagram

LIST OF ABBREVIATIONS

ADAG	Almost Directed Acyclic Graph
APL	Alternate Path Length
ARP	Address Resolution Protocol
AT&T	American Telephone and Telegraph Company
BA	Barabasi Albert
BA2	Barabasi Albert 2
BFD	Bidirectional Forwarding Detection
BFS	Breadth First Search
BRITE	Boston University Representative Internet Topology Generator
C	Centralization
CC	Clustering Coefficient
DFN	Deutsches Forschungsnetz
DFS	Depth First Search
DSCP	Differentiated Services Code Point
FIB	Forwarding Information Base
GADAG	Generalized Almost Directed Acyclic Graph
GLP	Generalized Linear Preference
H	Heterogeneity
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPFRR	Internet Protocol Fast Re-Route
IPTV	Internet Protocol Television
IS-IS	Intermediate System to Intermediate System
ISP	Internet Service Provider
LFA	Loop Free Alternates
LL	Lower Limit
LLDP	Link Layer Discovery Protocol
LOS	Loss Of Signal

MRC	Multiple Routing Configurations
MRT	Maximally Redundant Trees
MTR	Multi Topology Routing
MT-IPFRR	Multi Topology Routing based IPFRR
N2C	Non-2-connected
NP	Nondeterministic Polynomial
OSPF	Open Shortest Path First
QoS	Quality of Service
RAPL	Relative Alternate Path Length
RTGWG	Routing Area Working Group
SDN	Software Defined Network
SPF	Shortest Path First
SR	Success Rate
TAPL	Total Alternate Path Length
UDP	User Datagram Protocol
UL	Upper Limit
VoIP	Voice over IP
VT	Virtual Topology
2C	2-connected

1. INTRODUCTION

The availability of communication networks has become crucial with the recent deployments of next generation services such as Voice over IP (VoIP), Internet Protocol Television (IPTV), and mobile services. Internet Service Providers (ISPs) face new challenges to meet Quality of Service (QoS) requirements of these real-time services since IP networks may experience link and node failures. In order to seamlessly protect real-time services in IP networks, IP Fast Re-Route (IPFRR) technologies, which provide loop-free fast convergence, have been an active research area since the re-convergence of the existing routing protocols is a slow process [1]. The frequent topological changes in operational networks caused by network failures or network planning and maintenance activities may have a significant impact on the performance of the networking algorithms. Therefore, evaluating the IPFRR algorithms under varying topological conditions prior to their deployments is crucial in designing robust networks. Such an elaborate topological analysis becomes feasible if the tasks of generating diverse topologies and analyzing the impact of their characteristics on the network performance can be automated. Topological analysis is also widely used for social [2,3] and biological networks [4,5].

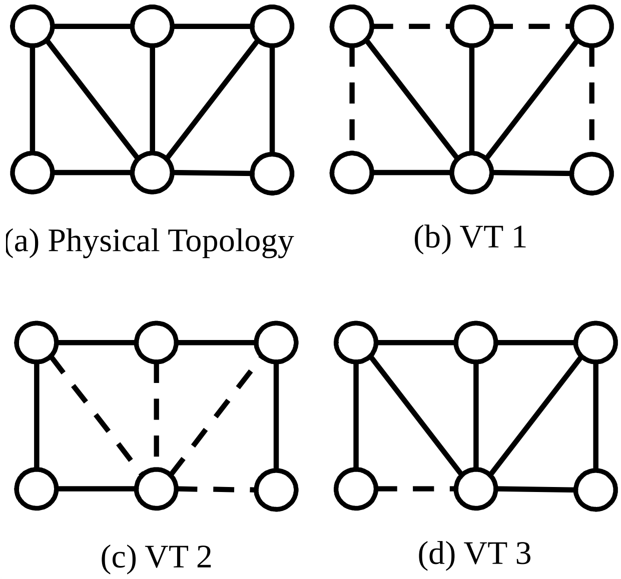
The Routing Area Working Group (RTGWG) of the Internet Engineering Task Force (IETF) has been working on standardizing IPFRR technologies to extend link-state routing protocols by providing fast recovery from link/node failures. IETF RTGWG has already standardized the IPFRR framework [6] and Loop Free Alternates (LFA) [7]. The LFA technology locally updates the routing tables at routers where the failure is detected. Since no explicit signaling among routers is needed and the LFAs are pre-computed, the fast re-routing of real-time traffic can be achieved. ISPs are planned to deploy this technology due to its simplicity and having no interoperability issue. However, the LFA technology does not provide full alternate path coverage for single link failures in operational networks. Multiple Routing Configurations (MRC) [8] is a widely studied IPFRR technology that has been proposed in the literature to provide full coverage. Maximally Redundant Trees (MRT) is another technique with full coverage which is currently heavily investigated within

IETF [9–11]. Not-via [12] and multi-hop repair paths [13–16] are other technologies that have been proposed in the literature.

Multi Topology Routing (MTR) uses virtual topologies (VTs) to construct alternate routing tables, and has been standardized by IETF as extensions to Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS) [17, 18]. The flexibility of having multiple routing tables allows not only to adopt different routing paths for different classes of traffic, but also to evenly distribute the link loads on the network. This redundancy in routing tables can also be used to achieve fast re-routing if VTs are constructed in such a way that an anticipated failed component is not used to carry any traffic in at least one of the VTs.

MTR based IPFRR (MT-IPFRR) is introduced in [19, 20] as a credible alternative to other recovery mechanisms since it is more suitable for traffic engineering applications, and does not require the knowledge of whether the failed component is a link or node. MT-IPFRR uses VTs to construct alternate routing tables to be used to recover from anticipated failures. Each VT has the same network graph as the physical topology, but has different link weights. Certain links in VTs are assigned a very high weight to guarantee that each anticipated failed component is not used to forward any transit traffic in at least one of the VTs. Upon the detection of each failure, the disrupted traffic can be re-routed towards its destination according to a VT where the failed component is by-passed. MT-IPFRR introduces the overhead of computing Shortest Path First (SPF) trees for multiple VTs, and maintaining multiple forwarding states in routers [19–22]. When the number of VTs gets higher, the time required to compute alternate routing tables, the look-up time for alternate next-hops, and the amount of required Forwarding Information Base (FIB) storage increase.

In MT-IPFRR, the VT construction procedure has to be standardized. Both the MRC and the MRT techniques can be used to construct VTs, and hence corresponding alternate routes. Once the VTs are constructed, fast re-routing will be realized in the same way to be defined by the IETF for both mechanisms. Fig. 1.1 shows an example physical topology and 3 VTs on it. If one of the links in the physical network fails, MT-IPFRR can immediately re-route the traffic initially going through the failed link to an alternate path using a pre-computed VT, in which the anticipated failed links represented by a dashed line are not used to carry any traffic. A router which detects a failure marks the IP header of



(a) Physical Topology

(b) VT 1

(c) VT 2

(d) VT 3

Figure 1.1. VTs computed for an example physical topology

each re-routed packet with the identity of the newly activated VT. This identification enables other routers on the alternate path to use the same VT for routing the recovered traffic. If the failure is permanent, the VT construction algorithm should be run again using the new topology to calculate the new VTs. On the other hand, the non-affected traffic continues to be forwarded based on the SPF tree calculated by the underlying link state routing protocol.

The IPFRR functionality should be automated by implementing the IPFRR tasks as extensions to the underlying link state routing protocol running at each node since performing manual operations by a network administrator is either not tractable or prohibitively expensive. This intelligence can also be implemented using software agents which use well-defined interfaces to interact with the routing protocol. In a fully automated MT-IPFRR mechanism, the tasks in Table 1.1 are performed at each node without a human in the loop. Task 1 can be accomplished by using the link state database of the underlying routing protocol. Task 2 requires the development of heuristic methods which will allow all the nodes in a network to construct identical VTs, and hence to compute consistent alternate routes for any anticipated failure event. Such heuristic methods can be developed since all the nodes have the same view of the physical network topology thanks to the underlying link state routing protocol. Task 3 implements the constructed VTs within a router by configuring each network interface to belong to a subset of VTs. Tasks 4 and 5 allow the fast detection of failures and the activation of the VT excluding the failed component, respectively. The VT activation can be performed by marking the IP header of each rerouted packet with the identity of the newly activated VT using a packet marking technique such as the DSCP (Differentiated Services Code Point) field of IP header [8]. Task 6 enables the usage of re-converged shortest alternate routes computed by the underlying link state routing protocol. If an IPFRR mechanism provides protection against all network component failures, the automation of the steps listed in Table 1.1 leads to self-recovering networks against failures.

Table 1.1. Tasks to be performed by MT-IPFRR without a human in the loop

Index	*Task*
1	Network topology is extracted
2	Consistent VTs and hence alternate routes are calculated
3	MTR configuration is performed
4	A fast failure detection process monitors the status of links
5	When a failure event is detected, packets are marked to be forwarded over a VT which exludes the failed link
6	Packet marking process is stopped when the underlying routing protocol converges new routes after the failure event
7	The above steps are repeated for subsequent failures

The MRC constructs a configurable number of VTs which has to be set carefully to obtain full alternate path coverage [8]. The MRC randomly iterates through each node in the physical topology, and constructs VTs in a round-robin fashion to prevent each node from transiting any traffic in one of the VTs, which is called *node isolation*. Each VT constructed by the MRC contains all the nodes in the topology, and should provide a path among each pair of its nodes. The VT requirement of the MRC is defined to be the cardinality of the set of VTs that the MRC should construct to provide full failure coverage. Note that the cardinality of the set of VTs to be constructed is defined as the number of VTs configured as input to the MRC algorithm and should be sufficiently large to allow for a successful termination. The number of VTs which must be constructed by MRC to successfully isolate each node in one of the VTs should be kept small since a higher number of VTs increases the complexity in terms of processing time and state requirements of routers [21, 22]. If the operation of the MRC can be enhanced to take the topological characteristics into consideration, its VT requirement can be reduced since the VT requirement highly depends on the structure of network topologies [21].

Maximally Redundant Trees (MRT) relying on the redundant tree concept [23] is another technique that has been investigated by the IETF [9–11]. Each router running the MRT proactively calculates a special directed graph whose normally and reversely directed versions are used by each router to compute two maximally redundant trees rooted at each possible destination in the network. Maximal redundancy among these trees ensures that they share a minimum number of common links and nodes. The MRT computes only two VTs by default to provide protection against single failures. Although its operational complexity is relatively low, its alternate path lengths are not scalable to the network size and density.

Software Defined Network (SDN) is a new paradigm in the networking area to make an underlying communication network programmable by physically separating control and data forwarding planes. In traditional networks, control and data planes are closely tied together, and it is more challenging to dynamically adapt the operations of the network resources for the changing needs. The SDN architecture maintains a global view of the network by abstracting the underlying infrastructure for a centralized management. SDN provides a more efficient usage of limited network resources by its dynamic reconfiguration capability as the network conditions change.

The forwarding plane of SDN may experience link and node failures which should be fast recovered to seamlessly support the real-time services. IPFRR can be performed in SDN using two different approaches, namely, *restoration* and *protection* [24]. In the restoration approach, alternate routing tables are proactively computed by the controller in advance of the failure while the resources for the alternate paths are allocated upon the detection of the failure. This approach requires the switches to contact the controller to notify that it should update the flow tables of all the switches along the affected paths with the new flow entries. On the other hand, in the protection approach, both the computation of the alternate paths and the allocation of the resources are performed in advance of the failure. As opposed to the restoration case, the switches activate the alternate paths without the need to contact the controller in case of a failure. Therefore, the protection approach provides a faster failure recovery. However, it requires more resources to proactively install backup flow entries into the flow tables of switches.

1.1. Research Objectives

In this thesis, we aim at reducing the operational complexity of the MRC algorithm in terms of processing time and state requirements of routers by reducing its VT requirement. We focus on improving the MRC since its alternate path lengths are scalable with respect to the network size and density whereas it generally requires more than two VTs to provide full failure coverage. A reduction in the VT requirement of the MRC will reduce the overhead of computing SPF trees for multiple VTs, and maintaining multiple forwarding states in routers [19–22]. This will allow us to balance the trade-off between alternate path lengths and number of VTs in MT-IPFRR. We also aim at developing a new MT-IPFRR technique for SDN which is a new paradigm in the networking area.

1.2. Contributions

In this thesis, we describe the operational principles of the MRC and the MRT on example networks. We present the performance comparison of these mechanisms using synthetic network topologies with varying characteristics (i.e., number of nodes and average number of neighbors). Through extensive analysis, we show that the alternate path lengths achieved by the MRT are not scalable with respect to the network size and density whereas the alternate path lengths of the MRC only slightly change as the network size and density vary [25]. We believe that this is an important scalability result providing a guidance in the selection of MT-IPFRR mechanism. We also perform experiments using realistic ISP level topologies to investigate the impact of the network size and density on the performance of the MRC. The numerical results showed that the VT requirement of the MRC highly depends on the structural properties of network topologies [21, 26]. We discover that the impact of the topological properties other than network size and density should be investigated since a trend in the number of nodes in a topology as well as the density can not always account for the trend in the VT requirement of the MRC.

We developed an automated topological analysis tool to obtain the performance results for the MRC algorithm under varying topological conditions since the number of VTs that should be constructed by the MRC highly depends on the structure of network topologies [27]. Our tool integrates Boston University Representative Internet Topology Generator (BRITE) [28] and Cytoscape's Network Analyzer [29], and is used to generate diverse network topologies, extract topological characteristics, and correlate the topological characteristics with the performance of the MRC. We provide the formal definitions of both the topology generation models and the topological characteristics evaluated in our experiments. The topological dependency of the MRC algorithm is elaborately investigated to provide a reasoning why the MRC performs poorly in heterogeneous topologies which tend to have *hub* nodes with a much higher degree than the other nodes. Our topological analysis experiments were performed using $11,500$ topologies, which are generated by selectively varying the parameters of the existing topology generation models. Visual plots are provided to demonstrate the structural properties of the generated topologies in terms of heavily used network characterizing metrics including heterogeneity (H), centralization (C), and clustering coefficient (CC). H of a network reflects the inclination of the topology to have hub nodes, whose node degree is much higher than the rest of the network. H is defined as $H = \sigma_d/\mu_d$ in [30] where σ_d and μ_d are the standard deviation and the mean of the node degree distribution d, respectively, so that a network becomes more heterogeneous if it has more hub nodes whose node degree is much higher than the rest of the network. C of a network is an indicator of the compactness of the network. CC of a node is a measure of the inter-connectivity of the nodes's neighbours. The MRC is run in an automated fashion on the generated topologies to obtain the topological analysis results which show that the distribution of node degrees has a significant impact on the performance of the MRC. Inspired by the results of our topological analysis, we propose an extension to the MRC which takes the node degree information into account [27]. Numerical experiments using both synthetic and real networks show that our heuristic significantly reduces the complexity in terms of processing time and state requirements of routers by decreasing the number of VTs which should be constructed by the MRC. The analysis results confirm the effectiveness of our systematic approach in performing the topological analysis of networking algorithms. We also provide graph theoretical proofs to show why our extension improves the performance of the MRC especially in heterogeneous networks.

In this thesis, we also propose two variants of the MRC, namely mMRC-1 and mMRC-2, which take the structural properties of the VTs into consideration during their construction to reduce the VT requirement of the MRC, and hence, decrease the operational complexity in terms of processing time and state requirements of routers [31]. Extensive simulations show that our proposed heuristics retain the alternate path lengths within acceptable limits. Our heuristics do not attempt to find the set of VTs with the optimum cardinality, which is an NP (Nondeterministic Polynomial)-complete problem [19, 21]. Instead, we develop two variants of the MRC that more intelligently construct VTs to increase their robustness against network partitioning. During the VT construction process, mMRC-1 keeps the connectivity of the VTs as high as possible by using the node degree information, whereas mMRC-2 reduces the heterogeneity level of the VTs. Using our automated topology generation and analysis tool, our experiment results have been obtained on 3200 topologies with diverse structural properties. Numerical results show that our heuristics consistently require a smaller number of VTs than the MRC for all experiments using the synthetic topologies. mMRC-1 reduces the VT requirement of the MRC up to 31.84% and performs better than mMRC-2 as the heterogeneity level of the topologies in our experiments increases. On the other hand, mMRC-2 provides an improvement of up to 28.44% and achieves higher improvement percentages than mMRC-1 as the heterogeneity decreases. The alternate path lengths of mMRC-1 and mMRC-2 are only slightly higher than the alternate path lengths of the MRC.

We propose a new MT-IPFRR technique for SDN by defining all the tasks which should be performed by the controller during the failure recovery process [32, 33]. The central architecture of SDN makes it an ideal platform to deploy MT-IPFRR since the central computation of the VTs by the controller guarantees the consistency among the alternate routing tables. Our technique automates the tasks including the discovery of the underlying physical topology, keeping track of the failures in the data plane and the modification of the flow tables of the switches to activate a primary path in no-failure scenario or an alternate path upon detection of a failure. Our implementation utilizes the MRC algorithm to construct VTs which provide protection against single link/node failures in the data plane. Our experimental results show that our approach considerably reduces the recovery time from network failures compared to the reactive recovery in SDN.

The rest of this thesis is organized as follows. Chapter 2 provides our literature review. Chapter 3 presents our evaluation of MT-IPFRR mechanisms. Chapter 4 provides our automated topological analysis of the MRC. Chapter 5 introduces our extension to the MRC which takes the node degree information into account. Chapter 6 presents our topology-aware variants of the MRC. Chapter 7 describes our MT-IPFRR technique for failure recovery in SDN. Chapter 8, Chapter 9, and Chapter 10 provides our discussion, conclusions, and recommendations, respectively.

2. LITERATURE REVIEW

2.1. Topology-Aware Networking

The influence of the topological characteristics on the design of networking protocols is investigated in [34]. The authors in [35] examine the effect of network topologies on overall network performance for satellite applications. [36] proposes node degree distribution based metrics for analyzing the structure of network topologies. The structural properties of network topologies have an impact on the convergence time of inter-domain routing protocol BGP [37] and on the performance of overlay routing [38]. Simulation results demonstrating the impact of power-law topologies on the routing dynamics are presented in [39]. The work presented in [40] investigates how the topology of a network impacts the average path length which is considered to be the main measure of the network performance. It also proposes a topology design algorithm to reduce the average path length by introducing high-degree central nodes to the topology. The impact of the topological characteristics on the performance of wireless mesh networks is studied in [41] which considers the topological metrics including the number of nodes, the path length and the link density. It uses data mining to discover the relationships between the topological characteristics and the network performance. The influence of the network topology on the parallel streaming of high-quality video is examined in [42] using actual ISP networks. The work in [43] proposes the usage of topological awareness for geographical routing in wireless sensor networks which allows the nodes to obtain the overall topology information. The problem of finding the best possible topology which minimizes the overall power consumption and provides fault tolerance in wireless ad hoc networks is studied in [44–47] by adaptively adjusting the network topology to the current traffic conditions. The effect of the ring and star topologies on the performance of the in-vehicle network is analyzed in [48].

The evaluation of the IPFRR technologies requires experimenting on a large topology pool with diverse characteristics [21]. Our work in this thesis automates the topological evaluation of the IPFRR algorithms, which allows the experimentation on a large structurally diverse topology pool. Performance evaluation results obtained using a restricted topology pool may be misleading since the negative aspects of a networking algorithm which appear only under certain topological conditions may not be studied. Since the previous IPFRR evaluation studies do not benefit from such automation, they either use a small-size synthetic/real topology pool [21, 22] or a small set of topology generation models or only a small set of parameter values for the topology generation models [8, 9].

2.2. Network Resilience Through Multi Topology Routing

MTR has been standardized by IETF as extensions to OSPF and IS-IS [17, 18]. MTR is used to provide QoS support in a tactical mobile network in [49]. It uses VTs to improve the QoS and available capacity in different subsets of the physical topology by dynamically blocking certain traffic in the network. The work presented in [50] provides an automation tool to dynamically configure the link weights in VTs to balance the traffic load across the network. MTR is proposed to provide fault tolerance in inter-satellite networks in [51]. An energy efficient network topology design is performed in [52] which determines the optimal VTs in order to minimize energy consumption in the network by powering off as many unnecessary network elements as possible. It also proposes a fast re-route technique to re-route the traffic previously being transmitted using the most recently powered off network element. A novel routing protocol for wireless networks is presented in [53]. The proposed routing algorithm relies on VTs to route the traffic based on the security preferences of the nodes in the network. This provides each node with the freedom to decide the set of nodes that it considers secure to participate in the data transmission.

MT-IPFRR constructs VTs to re-route the recovered traffic in case of link/node failures on a detour path around the failed component. The work presented in [21] constructs VTs in such a way that certain links of a topology are not isolated in any VT to reduce the VT requirement. However, their VT construction algorithm increases the complexity of the forwarding process in routers. The algorithm proposed in [22] aims to maximize the number of isolated nodes per VT, which may reduce the number of VTs. However, the proposed approach requires a large number of iterations, and may not always succeed in constructing VTs. The techniques proposed in [54, 55] aim at reducing the alternate path lengths of the MRC by including the isolated links in a VT excluding the actually failed one in alternate paths as opposed to the case for the MRC which do not consider any isolated link for packet forwarding. However, they report that their approaches increase the size of alternate routing tables. The work in [56] proposes a VT construction algorithm to uniformly distribute the link loads in VTs. Their algorithm selects certain special nodes with a relatively higher node degree in a pre-processing step, and attempts to isolate the neighboring links of these special nodes in different VTs as much as possible to balance the path diversity across the VTs. The work presented in [57] relies on MT-IPFRR to reduce the power consumption for a green network while providing full protection against link failures. However, their evaluation considers only small-sized topologies. An MT-IPFRR algorithm is proposed in [58] to construct VTs which will provide alternate paths in case of large-scale geographical events to reduce the amount of disruptions in network traffic. For this purpose, the links, which are close to the geographical region affected most from the geographical event, are assigned a very high weight in a VT so that the alternate paths avoid the affected region. Similar to our heuristics in this thesis, [59] proposes an algorithm to enhance the MRC by prioritizing the isolation of the high-degree nodes. However, our heuristics take the topological characteristics of the VTs into account during their constructions. In [60], an upper bound determined by a special cycle existing in the network topology is proposed for the number of VTs required to provide full failure coverage, and the quality of this bound is evaluated. The implementation of the MRC using the MTR standard of IETF is discussed in [61] which favours the MRC as a promising candidate for performing IPFRR.

In this thesis, we compare the MRC and the low-point version of the MRT. The performance results show that the MRT’s alternate path lengths significantly increase with respect to network size and density while the MRC generates scalable alternate paths. The performance of both the low-point and the SPF versions of the MRT is compared with Not-via in [62], and it is concluded that both versions of the MRT generate significantly longer alternate paths compared to the optimum SPF routing. The work in [63] proposes an IPFRR scheme which combines the LFA and the MRT techniques by first attempting to find an LFA in case of a single failure, and using the MRT to determine the alternate path if the LFA does not exist. The problem of protecting the multicast communications against single link or node failures is considered in [64, 65]. It constructs redundant multicast trees in such a way that, when a single link or node fails, every destination node is still connected to the source node in at least one of the two trees.

2.3. Failure Recovery in Software Defined Networks

In [24], the restoration and protection approaches for SDN are studied using example networks, and an analytical model is introduced to calculate the failure recovery time in the restoration case. However, it does not provide a self-recovering framework to perform IPFRR in SDN to recover from any network failure. The work in [66] enhances the LFA with a loop detection mechanism, and discusses how this enhanced version of the LFA can be implemented in SDN without providing an actual implementation. The work in [67] proposes a protection scheme which attempts to reduce the recovery time by relying on a faster failure detection. End-to-end and segment protection mechanisms are analyzed in [68] and [69], respectively. The work in [70] proposes a monitoring method for SDN to reduce the controller load caused by the alternate path calculation. In [71], a runtime system that automates the failure recovery in SDN is proposed without providing any experimental results. When a failure is detected, a new controller is activated to compute the alternate paths. A congestion-aware failure recovery mechanism for hybrid SDN consisting of both traditional routers and SDN-enabled switches is proposed in [72] which activates multiple alternate paths in case of a single link failure to reduce the post-recovery congestion and balance the traffic load accross the network. The problem of protecting the controller traffic is addressed in [73–75] to ensure a reliable communication between the data and control planes. These approaches propose the placement of multiple controllers in control plane which allows the switches to have at least one operational connection to one of the controllers at all times. An algorithm to recover from the inter-domain link failures is proposed in [76]. They suggest the usage of an SDN architecture to provide a restoration path if one of the links among the BGP routers fails. In [77], multipath routing is used to provide fault tolerance against network failures in the SDN-based network of Google which connects its data center accross the planet.

3. MULTI TOPOLOGY ROUTING BASED IPFRR MECHANISMS

In this chapter, we introduce the MT-IPFRR technologies, namely the MRC and the MRT algorithms, describe their operational principles using example networks, and present a trade-off analysis of these two approaches. We implemented a comprehensive analysis tool to evaluate the performances of the MRC and the MRT mechanisms using various network topologies. Through extensive analysis, we show that the alternate path lengths achieved by the MRT are not scalable with respect to the network size and density while the alternate path lengths of the MRC only slightly change as the network size and density vary. We believe that this is an important scalability result providing a guidance in the selection of MT-IPFRR mechanism to recover from the failures in IP networks.

The rest of the chapter is organized as follows. Sections 3.1 and 3.2 explain the MRC and the MRT algorithms, respectively, and describe their operational principles on example networks. In Section 3.3, we introduce our in-depth analysis tool, and present the performance comparisons of the MRC and the MRT using network topologies with varying size and density.

3.1. Multiple Routing Configurations (MRC)

3.1.1. Overview

The MRC is a VT construction algorithm for MT-IPFRR which provides full protection against single link/node failures in 2-connected (2C) topologies [8]. Each VT constructed by the MRC has the same links and nodes as the physical topology. Although the links in each VT initially have the same weights as in the physical topology, the MRC modifies the weights of certain links in each VT during its execution by assigning them high values. A link in a VT can take one of the three possible weights: *(i)* the same weight as in the physical topology, *(ii)* a sufficiently large finite value called w_r, and *(iii)* infinity. A link with a weight of w_r is called *restricted* whereas a link with a weight of infinity is called *isolated*. w_r is intelligently selected so that a restricted link is used only to deliver traffic that is either sourced at or destined for the attached node [8]. On the other hand, an isolated link does not carry any traffic at all. To isolate a node, one of its attached links is assigned the weight of w_r, and all the remaining links take a weight larger than or equal to w_r. An isolated node can not be used to forward any transit traffic in the VT. The set of non-isolated nodes/links and non-restricted links in a VT is called the *backbone* of the VT, which is used to forward transit traffic. VTs constructed by the MRC satisfy the following properties [8]:

- *(P1)* Each link/node in the physical topology is isolated in exactly one of the VTs.
- *(P2)* The backbone of each VT is connected.
- *(P3)* Each isolated node in a VT is connected to the backbone via at least one restricted link.

The MRC maintains a Q_n queue, which contains all the nodes of the topology in an arbitrary order. The order defined by Q_n, in which the nodes are attempted to be isolated in VTs, is called the *node isolation order* in this thesis. The MRC attempts to isolate the topmost element of Q_n in a VT as long as its removal with its attached links does not partition the backbone of the VT. The topmost member is removed from Q_n once it becomes isolated in a VT. If a node can not be isolated in a VT, it is attempted to be isolated in a next VT in a round-robin fashion. If a node can not be isolated in any VT, the MRC fails to construct the set of VTs. To isolate a node x in a VT_x, all (x,y) links are traversed, where y represents a neighbour of x. The weight of a (x,y) link in VT_x is set to infinity if *(i)* there exists a VT_y where y is isolated, and *(ii)* the weight of the link (x,y) is w_r in VT_y, and *(iii)* the node x has a link with a finite weight other than (x,y) in VT_x. It is also set to an infinite weight if *(i)* there does not exist a VT_y, where y is isolated, and *(ii)* the node x has a link with a finite weight other than (x,y) in VT_x. Otherwise, the weight of the (x,y) link is set to w_r.

The number of VTs to be constructed by the MRC should be configured as an input to the algorithm. The MRC successfully terminates only if the number of VTs is configured to be sufficiently large. In the case that the MRC fails, it should be re-configured to construct a higher number of VTs to force its successful termination. The *minimum VT requirement (minVT)* of the MRC can be defined as follows:

Definition. $minVT \geq 2$ is an integer such that $\forall i : 2 \leq i < minVT$, the MRC fails to construct a set of VTs with the cardinality i, but the MRC succeeds to construct a set of VTs with the cardinality $minVT$.

$minVT$ is dependant on the structural properties of the physical topology, and should be kept as small as possible to reduce both the consumption of router resources and the computational overhead of MT-IPFRR to calculate the alternate forwarding tables [22]. Note that the set of VTs with the cardinality $minVT$ satisfies *P1*, *P2*, and *P3*.

3.1.2. Example Operation

Assuming that the number of VTs is configured to be 3, the VT construction procedure of the MRC is explained using the example topology with 9 nodes and 14 links shown in Fig. 3.1. Note that each VT initially contains all the links and nodes of the topology, and the initial node isolation order is randomly chosen to be $Q_n = \{R9, R8, R1, R2, R3, R4, R7, R5, R6\}$. The set of isolated nodes for each VT_i is denoted by I_i $(1 \leq i \leq 3)$ which are initially empty.

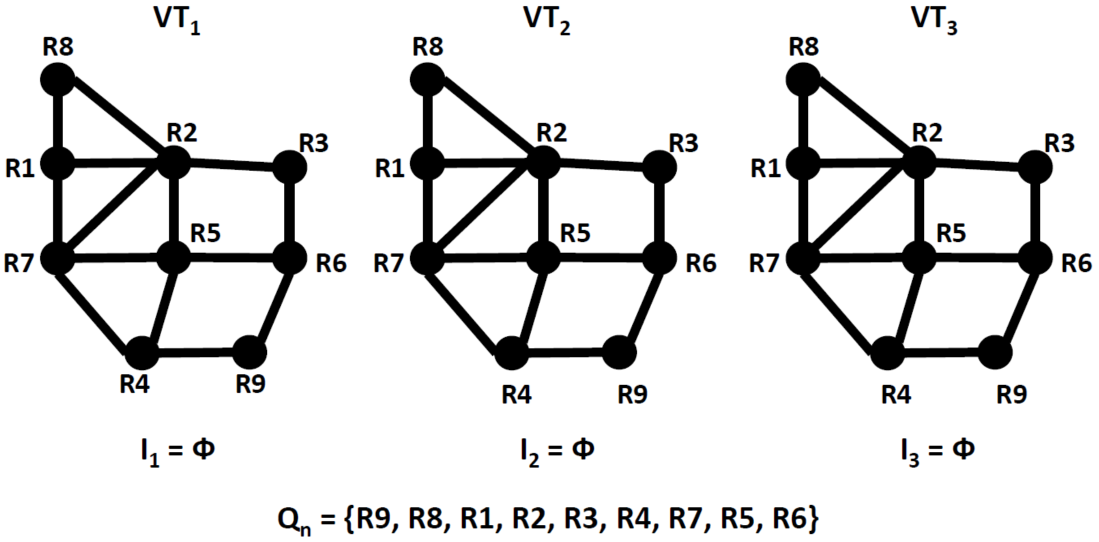


Figure 3.1. Initial structures of VTs

Fig. 3.2 shows how the link weights of the physical topology are modified by the MRC to isolate $R9$, $R6$ and $R3$. The non-isolated nodes and links are denoted by the black nodes and the bold lines while the isolated nodes, isolated and restricted links are indicated by the dotted circles, dotted and dashed lines, respectively. The topmost element, $R9$, of Q_n attempts to be isolated first in VT_1. In steps 1 and 2, the weights of the links $(R9, R4)$ and

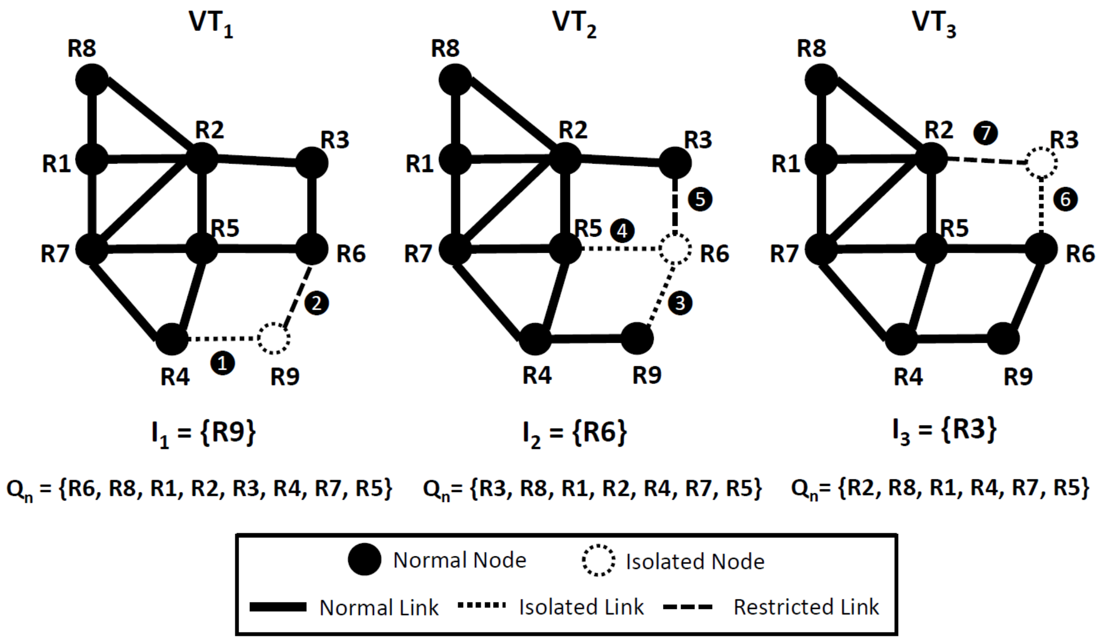


Figure 3.2. Structures of VTs after $R9$, $R6$ and $R3$ become isolated

$(R9,R6)$ are set to be infinity and w_r, respectively. The order in which the neighboring links of a node become isolated or restricted is also arbitrary in the MRC, which is indicated by the consecutive numbers within the black circles over the links. Note that $(R9,R6)$ is made restricted in step 2 since its isolation would cause $R9$ to become inaccessible to other nodes. The MRC attempts to isolate the next node in Q_n using a VT different from the one where the previous node was isolated. The aim of such an isolation is to balance the number of isolated nodes across VTs since such balancing may keep the alternate path lengths within acceptable limits. Following the isolation of $R9$, the MRC modifies Q_n since it must prioritize the isolation of $R6$ to guarantee the isolation of $(R9,R6)$, which has been made restricted in the previous step, even though $R6$ is located at the bottom of Q_n. Hence, $R6$ attempts to be isolated next in VT_2, and $(R6,R9)$ is traversed first. In step 3, the weight of $(R6,R9)$ is set to infinity since a link should be isolated in the same iteration as either of its end points [8]. $(R6,R9)$ can receive a weight of infinity since the other links of $R6$, namely $(R6,R5)$ and $(R6,R3)$, have the finite weights as in the physical topology at this point. The links $(R6,R5)$ and $(R6,R3)$ become isolated and restricted in steps 4 and 5, respectively. $R3$ is similarly isolated in VT_3 through the steps 6 and 7.

Fig. 3.3 illustrates how the nodes $R2$, $R8$, and $R1$ become isolated in VT_1, VT_2, and VT_3, respectively, whereas Fig. 3.4 shows the isolations of the nodes $R4$, $R7$, and $R5$. After

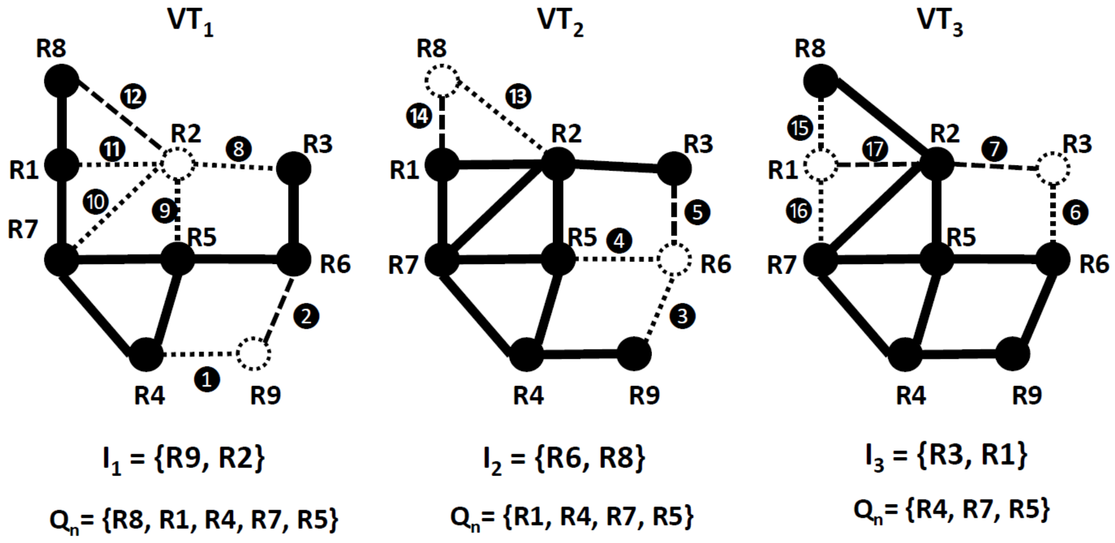


Figure 3.3. Structures of VTs after $R2$, $R8$ and $R1$ become isolated

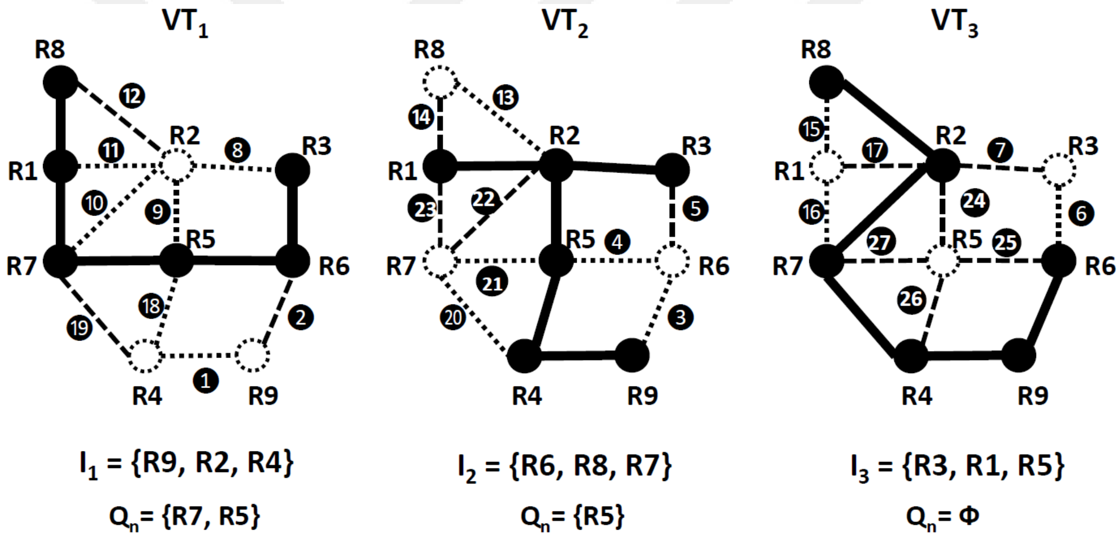


Figure 3.4. Structures of VTs after $R4$, $R7$ and $R5$ become isolated

27 steps, Q_n becomes empty, and, hence, the algorithm successfully terminates. Each VT ultimately has an equal number of isolated nodes, namely 3. The black nodes and bold lines in Fig. 3.4 constitute the backbone of each VT. Note that each node/link is isolated in exactly one VT. For example, the link $(R2, R7)$ is isolated in VT_1 while it is restricted and normal in VT_2 and VT_3, respectively. Each backbone in VTs forms a connected graph, and each isolated node in a VT is connected to backbone via a restricted link, which allows it to reach every other node in the same VT. For example, the isolated node $R2$ is connected to the backbone of VT_1 via the restricted link $(R2, R8)$.

As an example of the application of the MRC to the IPFRR, if the link $(R4, R5)$ fails, VT_1 is used to recover from this failure since the link $(R4, R5)$ is isolated in VT_1. In this case, the restricted link $(R4, R7)$ is the only link which can be used as the first or the last hop to deliver the traffic sourced at $R4$ and destined to any other node, or the traffic sourced at any other node and destined to $R4$, respectively, whereas the links on the backbone of VT_1 constitute the interior hops. If the failure of $(R4, R5)$ is detected to be permanent, the VTs are re-constructed using the updated topology.

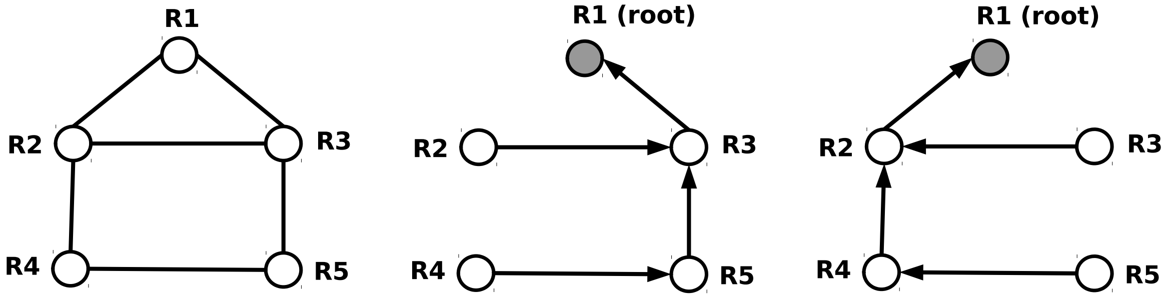
3.2. Maximally Redundant Trees (MRT)

3.2.1. Overview

The MRT provides full coverage against single network failures for all topologies including non-2-connected (N2C) ones as long as the failure does not partition the network [9]. Each router running MRT proactively calculates pairs of identically rooted maximally redundant trees, called *red* and *blue*, rooted in each possible destination in the network. Maximal redundancy among these two trees ensures that the two paths in red and blue trees from the same node to the identical root share a minimum number of common links and nodes. Upon the detection of a local failure, the affected traffic starts to be routed over whichever of the pre-determined red or blue trees excludes the failed link. On the other hand, non-affected traffic continues to be forwarded based on the shortest path tree calculated by the underlying link state routing protocol.

Fig. 3.5b and Fig. 3.5c demonstrate the red and blue trees which are rooted at $R1$ of an example 2C topology shown in Fig. 3.5a, respectively. It is possible to find two disjoint paths from any node to $R1$ since the topology is 2C. For example, $R5$ utilizes the paths $R5 - R3 - R1$ and $R5 - R4 - R2 - R1$ to reach $R1$ in red and blue trees, respectively. If $R5$ detects the failure of the link $(R5, R3)$, it starts to forward the affected traffic towards $R1$ using the blue tree which excludes the failed link. Note that the MRT calculates red and blue trees rooted at $R2$, $R3$, $R4$, and $R5$ as well even though they are not shown in Fig. 3.5.

The MRT relies on building a GADAG (Generalized Almost Directed Acyclic Graph), which is composed of either a single ADAG (Almost Directed Acyclic Graph) or multiple ADAGs in case of 2C and N2C topologies, respectively [9]. Each GADAG has a special



(a) 2-connected topology (b) Red tree (c) Blue tree

Figure 3.5. Red and blue trees calculated for an example topology

node called *global root,* which should be known to all nodes in the network prior to the execution of the MRT algorithm. Otherwise, GADAGs computed by each node would be different causing inconsistencies among the red and blue routing tables in the network. On the other hand, each ADAG has a special node called *local root,* which is the same as the global root in case of 2C topologies. The local root of an ADAG in a N2C topology is the articulation point which is closest to the GADAG's global root such that traffic sourced at the nodes within an ADAG should pass through the ADAG's local root to reach destinations in other ADAGs. The calculation of *low-point values,* which requires the execution of the DFS (Depth First Search) algorithm first, is a way of detecting articulation points in a network. An algorithm based on low-point value calculation is proposed in [9] for determining the local root of an ADAG.

While forming an ADAG, an initial loop which starts and ends at the local root is selected. Afterwards, this partial ADAG is incrementally extended by adding special directed paths of nodes called *ear*. The important point in deciding the direction of the links on an ear is to create no loops other than the ones containing the local root. These loops which start and end at the local root are needed by the MRT to construct two redundant alternate routes for a node *x* to reach the local-root by following the directed paths from *x* to the local-root and from the local-root to *x*.

3.2.2. Example Operation

We use the topology with 4 articulation points shown in Fig. 3.6 to demonstrate that the MRT is capable of operating even on N2C topologies. It shows the DFS and low-point numbers as well as the low-point inheritance relation among the nodes. Each node is

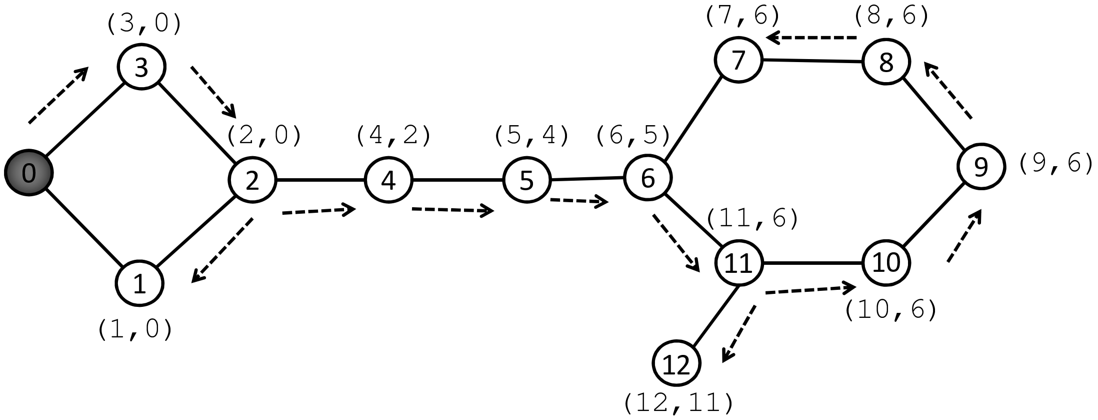


Figure 3.6. DFS and low-point numbers for an example N2C topology

labelled with a 2-tuple (a,b) where a and b stand for the DFS and low-point numbers, respectively. The global root (node 0) is shaded, and does not have any low-point number. For example, the fact that the nodes 8 and 9 have 6 as its low-point number means that the traffic sourced at these nodes and destined to the nodes 0, 1, 2, 3, 4 or 5 should pass through the node 6. A directed dashed line from a node *x* to *y* means that *y* inherits its low-point number from *x*. For example, the node 3 inherits its low-point number from the node 0. By exploiting the low-point inheritance relation [9] between the nodes in the example topology, the MRT algorithm determines the ears 0→1→2→3→0, 2→4→2, 4→5→4, 5→6→5, 6→7→8→9→10→11→6, and 11→12→11 in the shown order.

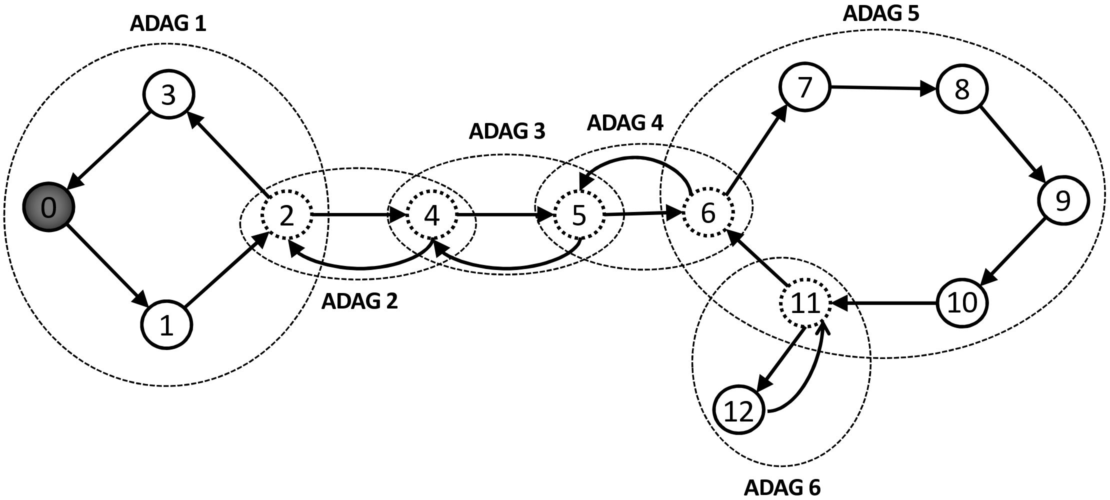
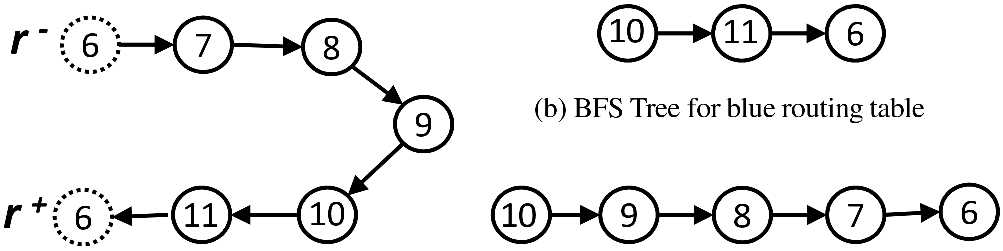


Figure 3.7. GADAG computed for the N2C topology in Fig. 3.6

Fig. 3.7 shows the GADAG calculated for the topology in Fig. 3.6, which is composed of 6 ADAGs. The local roots are symbolized with a small dashed circle while the large



(a) Modified ADAG

(b) BFS Tree for blue routing table

(c) BFS Tree for red routing table

Figure 3.8. Modified ADAG and BFS trees for red/blue routing tables

dashed ovals indicate the individual ADAGs. The articulation point in an ADAG, which is closest to the global root, is the local root of that ADAG. For example, ADAG 2 contains the nodes 2 and 4, and its local root is the node 2.

In the MRT, each node in the network firstly computes the next hops to the destinations in its own ADAG. For this purpose, each node splits its own local root r into two nodes, namely r^+ and r^-, such that r^+ and r^- have only incoming and outgoing links, respectively [9]. Afterwards, based on the normally and reversely directed versions of this modified ADAG, it computes two BFS (Breadth First Search) trees, namely red and blue trees, which are used to calculate the redundant paths to a certain destination. For example, the node 10 in Fig. 3.7 constructs the BFS trees in Fig. 3.8b and Fig. 3.8c based on the modified ADAG shown in Fig. 3.8a. The calculated BFS trees are utilized to create the blue and red routing tables [9]. As depicted in Fig. 3.8b, the node 10 uses the node 11 as the next-hop to the destinations 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, and 12 in the blue tree while it uses the node 9 as the next-hop in the red tree.

3.3. Experimental Results

In this section, we provide our comparison results for the MRC and the MRT using randomly generated topologies. We also present the performance results of the MRC using realistic ISP-level topologies.

3.3.1. Performance Comparison of the MRC and the MRT

To compare the performances of the MRC and the MRT, we developed a comprehensive analysis tool which has the following capabilities:

- Running both the MRC and the MRT consecutively on topologies with varying size and density which are randomly generated by the BRITE tool [28] using the Waxman model
- Comparing the performances of the MRC and the MRT against the re-converged OSPF in terms of alternate path lengths
- Computing the *success rate* (*SR*) of the MRC, namely, the percentage of topologies for which the MRC can successfully construct the configured number of VTs

The MRC software from Simula Research Laboratory (*www.simula.no*) was integrated into our analysis tool which processes the MRC output, namely the VTs, to compute the alternate paths in case of network failures. The MRT algorithm presented in [9] was implemented within our analysis tool. The correctness of our MRT implementation was verified such that the red and blue trees generated by our tool match the trees calculated for the sample topologies given in [9]. We also implemented a script which automates the task of evaluating the performances of the MRC and the MRT over various networks with 50, 100, 150, 200, 250, and 300 nodes, where initial link weights are set to unity. For each network size, we generated 50 random graphs. The alternate path lengths of the MRC with different number of VTs, namely, 3, 5, and 7, were compared against the alternate path lengths of the MRT. Note that there is no version of the MRT algorithm which utilizes more than 2 VTs. We report two separate comparison results corresponding to the sparse and dense versions of each network size, which were generated by setting the *m* parameter (i.e., the number of links per new node) in BRITE topology generator tool to 2 and 3 for the sparse and the dense topologies, respectively.

We provide three different sets of results for the MRC and the MRT as shown in Table 3.1a, 3.1b, and 3.1c where the number of nodes in a topology, the network density, and the success rate of the MRC are denoted by *n*, *m*, and *SR*, respectively. $RAPL_{MRC}$ and $RAPL_{MRT}$ represent the alternate path lengths of the MRC and the MRT relative to the re-converged OSPF, and can be formulated as $(TAPL/TAPL_{OSPF}) * 100$ where $TAPL$ and $TAPL_{OSPF}$ stand for the total lengths of all the alternate paths achieved by the MRC/MRT, and OSPF, respectively. Please note that the total length of all the alternate paths computed by OSPF correspond to 100%. The results show that the MRC may not always successfully terminate due to an insufficient number of VTs. In the cases that the MRC fails to compute the VTs (e.g., $n = 250$, $m = 2$ in Table 3.1a), the results for the MRT are deliberately not reported even though, by the nature of the algorithm, the MRT is always capable of determining alternate paths for any network topology as long as the failure does not partition the network. In other cases, each link in the network is failed sequentially, and the alternate paths to recover from the failure are computed for each of the MRC, the MRT, and OSPF.

Table 3.1a presents the comparison results for the MRC with 3 VTs and the MRT. No result is reported for the sparse topologies ($m = 2$) with 250 and 300 nodes since 3 VTs are not sufficient for the MRC to isolate all the nodes in these topologies. When the number of VTs is 3 and the number of nodes is 250 or 300, the number of isolated nodes per VT increases. In such a case, it is likely that the MRC's isolation of a node in a VT partitions its backbone. However, as the network density increases ($m = 3$), *SR* of the MRC improves because a higher path diversity of a dense network reduces the likelihood of partitioning the backbone. Fig. 3.9 shows that *SR* for the sparse networks significantly decrease as the network size grows. For example, *SR* for the sparse networks with 50 nodes is 44% while it becomes 0 for both 250 and 300 nodes. On the other hand, *SR* for the dense networks were higher than 90% for all cases. Table 3.1a also demonstrates that the alternate path lengths of the MRT significantly increase as the network size and density grow. For example, $RAPL_{MRT} = 525.39\%$ for the dense networks with 300 nodes is more than 5 times greater than the shortest one (i.e., 100%) utilized by the re-converged OSPF. We also observed that the alternate path lengths of the MRT for the dense networks are larger than in the case of the sparse networks while the contrary is true for the MRC.

Table 3.1b reports the comparison results for the MRC with 5 VTs and the MRT. The performance results of the MRT presented in Table 3.1b are similar to the ones reported in Table 3.1a. However, compared to the results in Table 3.1a, the alternate path lengths of the MRC are significantly reduced. For example, the MRC with 5 VTs has an *RAPL* of 116.24% in case of the sparse networks with 50 nodes while the MRC with 3 VTs adopts an *RAPL* of

Table 3.1. Comparison results for the MRC and the MRT

n	m	$RAPL_{MRC}$	SR	$RAPL_{MRT}$
50	2	129.05	44	227.50
	3	122.55	100	256.56
100	2	132.12	24	281.70
	3	123.94	100	324.32
150	2	129.98	8	329.65
	3	124.75	92	389.71
200	2	133.64	2	329.06
	3	124.32	94	427.83
250	2	-	0	-
	3	124.32	100	472.72
300	2	-	0	-
	3	124.79	90	525.39

(a) MRC with 3 VTs versus MRT

n	m	$RAPL_{MRC}$	SR	$RAPL_{MRT}$
50	2	116.24	100	230.23
	3	115.17	100	249.63
100	2	115.98	98	283.70
	3	114.99	100	325.80
150	2	116.43	100	317.94
	3	115.03	100	381.77
200	2	116.26	100	355.38
	3	115.03	100	422.17
250	2	116.23	98	384.58
	3	115.00	100	467.61
300	2	116.22	98	414.67
	3	114.90	100	512.40

(b) MRC with 5 VTs versus MRT

n	m	$RAPL_{MRC}$	SR	$RAPL_{MRT}$
50	2	111.80	100	224.35
	3	112.97	100	252.25
100	2	112.35	100	281.27
	3	112.24	100	327.92
150	2	112.45	100	330.51
	3	112.25	100	385.63
200	2	111.79	100	357.06
	3	111.88	100	434.36
250	2	111.91	100	394.67
	3	111.93	100	487.38
300	2	111.69	100	409.94
	3	111.76	100	515.39

(c) MRC with 7 VTs versus MRT

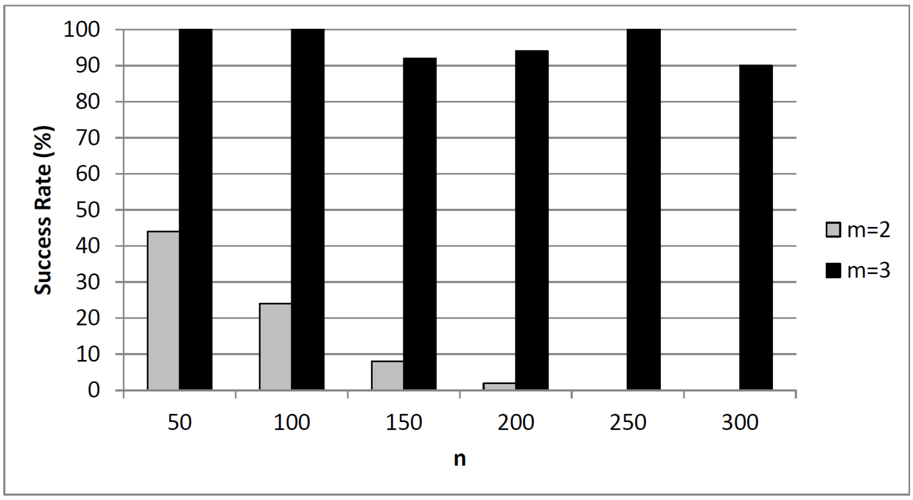


Figure 3.9. Success rates of the MRC for the construction of 3 VTs

129.05% for the same network size and density. Note that *SR* in Table 3.1b are higher than 98% for all network sizes and densities.

Table 3.1c reports the comparison results for the MRC with 7 VTs and the MRT. Compared to the results shown in Table 3.1b, the alternate path lengths of the MRC are slightly reduced. For example, the MRC with 7 VTs has an *RAPL* of 112.24% in case of the dense topologies with 100 nodes while the MRC with 5 VTs uses an *RAPL* of 114.99% for the same topologies. The MRC with 7 VTs yielded an *SR* of 100% for all network sizes and densities.

Fig. 3.10 shows the increase in the *RAPL* values for both the MRC with 7 VTs and the MRT with respect to the network size, namely n. As shown in the figure, there is a linear increase in the *RAPL* of the MRT. The difference between the *RAPL* values of the MRT for the dense ($m = 3$) and the sparse ($m = 2$) topologies gets higher when n increases. For example, for the topologies with 300 nodes, *RAPL* is 409.94% when $m = 2$ whereas *RAPL* is 515.39% when $m = 3$. Hence, the difference between them becomes $515.39\% - 409.94\% = 105.45\%$. On the other hand, the difference between the *RAPL* values of the MRT for the topologies with 50 nodes is only $252.25\% - 224.35\% = 27.9\%$.

3.3.2. Performance Evaluation of the MRC Using Realistic ISP Topologies

We extended our analysis tool explained in Section 3.3.1 to compute the minimum VT requirement of the MRC, namely $minVT$, by successively configuring an increasing number

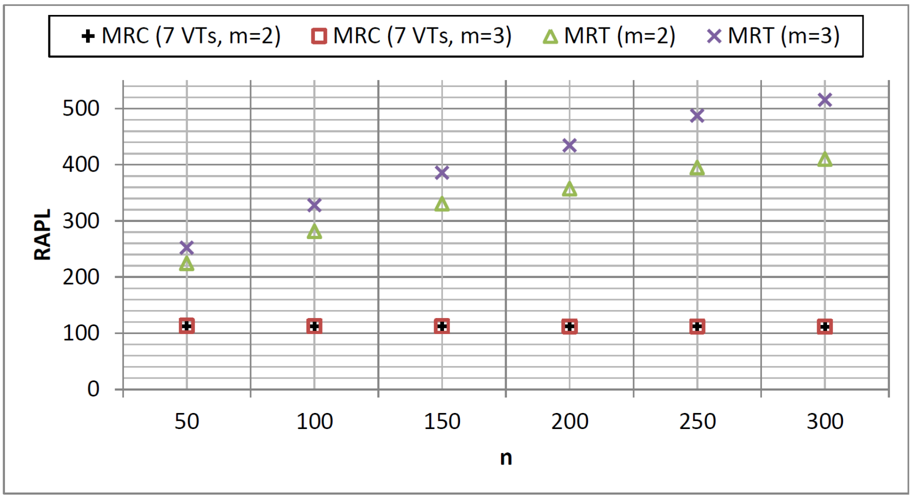


Figure 3.10. Comparison of the alternate path lengths of the MRC and the MRT

of VTs starting from 2 until the first successful termination of the MRC. In this section, we evaluate the alternate path lengths achieved by the MRC as well as its *minVT* using realistic ISP-level topologies. We also report *SR* of the MRC, namely the percentage of topologies for which the MRC can successfully construct the configured number of VTs, when the number of VTs is varied from 2 to 8. Table 3.2 lists the values for the BRITE parameters that we use in our experiments to generate realistic ISP-level topologies with 30 and 154 nodes [78]. These parameter values lead to the generation of the topologies which have a similar structure to *AT&T* (American Telephone and Telegraph Company) and *DFN* (Deutsches Forschungsnetz) ISP topologies. We randomly generated 70 topologies for each type of ISP network in Table 3.2 containing links with a unit weight. The name of each type of ISP-level topology has the format of *topology_name:X* where *X* represents the average number of links per node in the topology. As shown in Table 3.2, the topology with the highest density among all topologies with 154 nodes is *AT&T:3.33*.

Table 3.2. BRITE parameters used to generate realistic ISP-level topologies

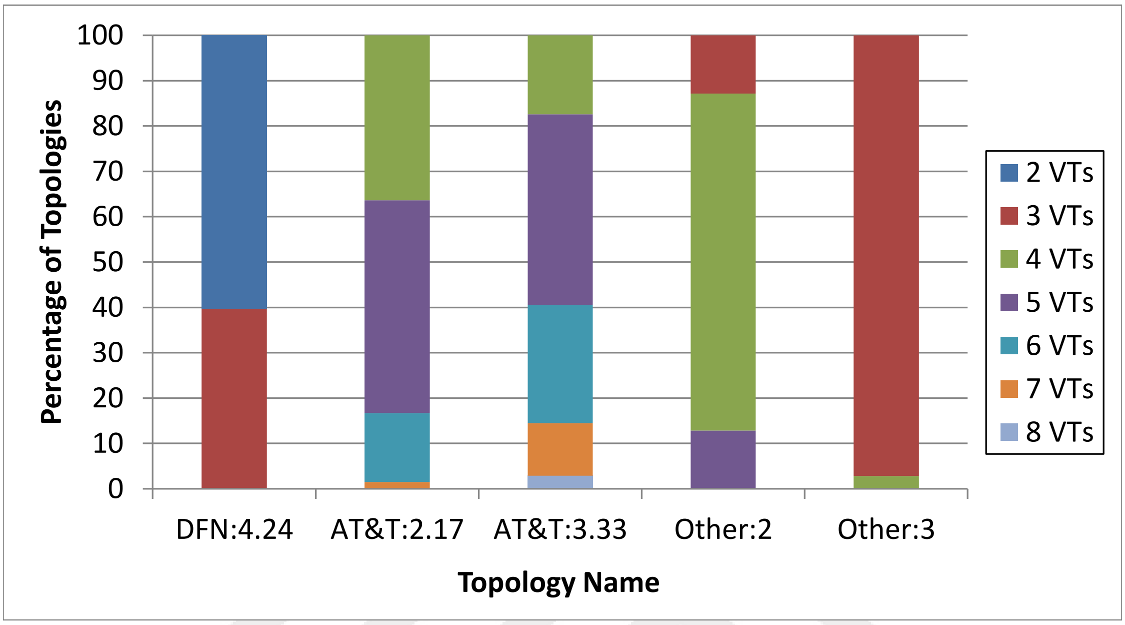
Topology Name	*Type*	*Method*	*AS#*	*Node#*	*Generation Model*	α	β	*m*
DFN:4.24	Bottom up	Random pick	17	30	GLP	0.42	0.62	3
AT&T:2.17	Bottom up	Random pick	31	154	GLP	0.15	0.15	2
AT&T:3.33	Bottom up	Random pick	31	154	GLP	0.42	0.62	2
Other:2	AS Only	–	–	154	Waxman	0.15	0.2	2
Other:3	AS Only	–	–	154	Waxman	0.15	0.2	3

Table 3.3 shows the *RAPL* values for the MRC when the number of VTs ranges from 2 to 8. As shown in the table, *RAPL* values for the MRC decrease as the number of VTs increases. This is due to the fact that an increase in the number of VTs improves the path diversity of each VT since each VT tends to have a smaller number of isolated nodes. We also observe that an increase in the density of *AT&T* topologies causes the *RAPL* values to decrease. For example, *RAPL* for 4 VTs is equal to 111.70% and 106.59% for the sparse and the dense versions of *AT&T* topologies, respectively. *RAPL* values for *Other:3* are 111.97% and 111.11% in case of 7 and 8 VTs, respectively, which are larger than the ones for *Other:2* topologies, namely 111.73% and 110.77%. These results contradict with the correlation between *RAPL* and network density observed for *AT&T* topologies. *RAPL* values for *DFN:4.24* are larger than the ones for *AT&T:3.33* even though *DFN:4.24* topologies have the highest density. These results indicate that it is not always possible to correlate the *RAPL* values of the MRC with the network density. As shown in Table 3.3, *RAPL* values for 2 and 3 VTs are not reported since the MRC fails to construct VTs in these cases.

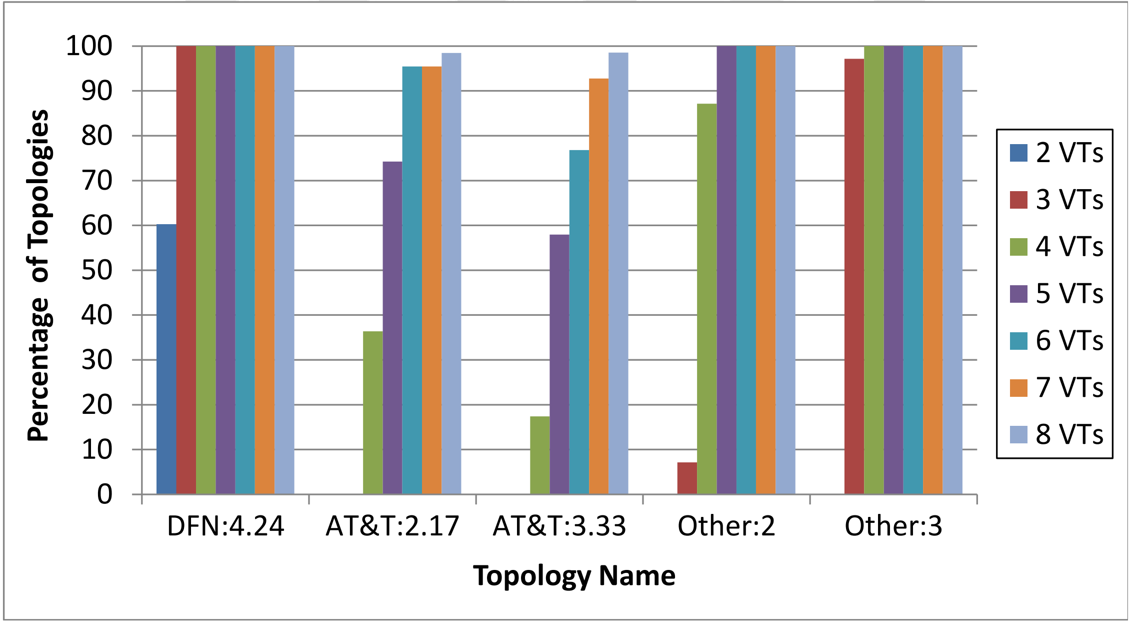
Table 3.3. RAPL values of the MRC for realistic ISP-level topologies

Topology Name	*RAPL* $(2VT)$	*RAPL* $(3VT)$	*RAPL* $(4VT)$	*RAPL* $(5VT)$	*RAPL* $(6VT)$	*RAPL* $(7VT)$	*RAPL* $(8VT)$
DFN:4.24	115.07	111.52	109.79	108.87	108.66	108.02	107.76
AT&T:2.17	–	–	111.70	109.67	108.23	107.15	106.75
AT&T:3.33	–	–	106.59	105.28	104.79	104.24	103.90
Other:2	–	132.01	120.72	116.33	113.57	111.73	110.77
Other:3	–	124.18	117.96	114.92	113.10	111.97	111.11

Fig. 3.11a demonstrates the performance results for the *minVT* of the MRC using the realistic ISP-level topologies. As shown in Fig. 3.11a, *minVT* is smaller for the most of the *DFN:4.24* topologies compared to other ISP-level networks such that *minVT* is equal to 2 for 60% of the topologies while it is equal to 3 for the remaining 40%. Note that the *DFN:4.24* topologies have the highest density in our experiments. However, the results depicted in Fig. 3.11a indicate that it is not always possible to correlate the *minVT* with the network density. For example, *minVT* is equal to 3, 4, or 5 in case of *Other:2* topologies while it may be larger, namely, 6, 7, or 8, in case of *AT&T:3.33* topologies which have a higher network density. Fig. 3.11b shows the *SR* when the number of VTs ranges from 2 to 8. For example, the *SR* values for the MRC vary as 36%, 74%, 95%, 95%, and 98% when the number of VTs ranges from 4 to 9, respectively, in case of *AT&T:2.17* whereas



(a) *minVT*



(b) *SR*

Figure 3.11. Minimum VT requirement of the MRC and and its success rates for realistic ISP-level topologies

the MRC fails when the number of VTs is configured to be 2 or 3. The results depicted in Fig. 3.11b expose an interesting behaviour of the MRC such that the MRC with 8 VTs does not successfully terminate for certain *AT&T:2.17* topologies even though the largest $minVT$ is 7 for *AT&T:2.17* as shown in Fig. 3.11a.

4. AUTOMATED TOPOLOGICAL ANALYSIS OF MULTIPLE ROUTING CONFIGURATIONS

Our work presented in Chapter 3 shows that the MRC provides scalable alternate paths compared to the MRT when the network size and density increase. Our experiments in Chapter 3 also reveal that the $minVT$ of the MRC highly depends on the structural properties of the network topologies. This implies that the performance of the MRC may differ even for topologies with the same number of nodes and densities.

In this chapter, we systematically analyze the impact of the topological characteristics on the performance of the MRC by developing an automated topological analysis tool to generate diverse network topologies, extract topological characteristics, and obtain the performance results for the MRC algorithm. Our tool integrates Boston University Representative Internet Topology Generator (BRITE) [28] and Cytoscape’s Network Analyzer [29], and automates both the generation and analysis of topologies with diverse structural properties. The impact of the topological characteristics on the performance of the MRC is elaborately investigated through extensive experiments to provide a reasoning why the MRC performs poorly in heterogeneous topologies which tend to have *hub* nodes with a much higher degree than the other nodes. We provide the formal definitions of both the topology generation models and the topological characteristics evaluated in our experiments.

Numerical experiments are performed using 6000 topologies, which are generated by selectively varying the parameters of the existing topology generation models. Visual plots are provided to demonstrate the structural properties of the generated topologies in terms of heavily used network characterizing metrics including heterogeneity (*H*), centralization (*C*), and clustering coefficient (*CC*). *H* of a network reflects the inclination of the topology to have hub nodes, whose node degree is much higher than the rest of the network. *C* of a network is an indicator of the compactness of the network. *CC* of a node is a measure of the inter-connectivity of the nodes’s neighbours. The MRC is run in an automated fashion on the generated topologies to obtain the topological analysis results which show that the node degree distribution has a significant impact on the performance of the MRC.

The rest of the chapter is organized as follows. Section 4.1 explains our systematic approach for performing the topological dependency analysis of the MRC algorithm. Section 4.2 provides our topological analysis results.

4.1. Topological Analysis of Multiple Routing Configurations

4.1.1. Our Workflow

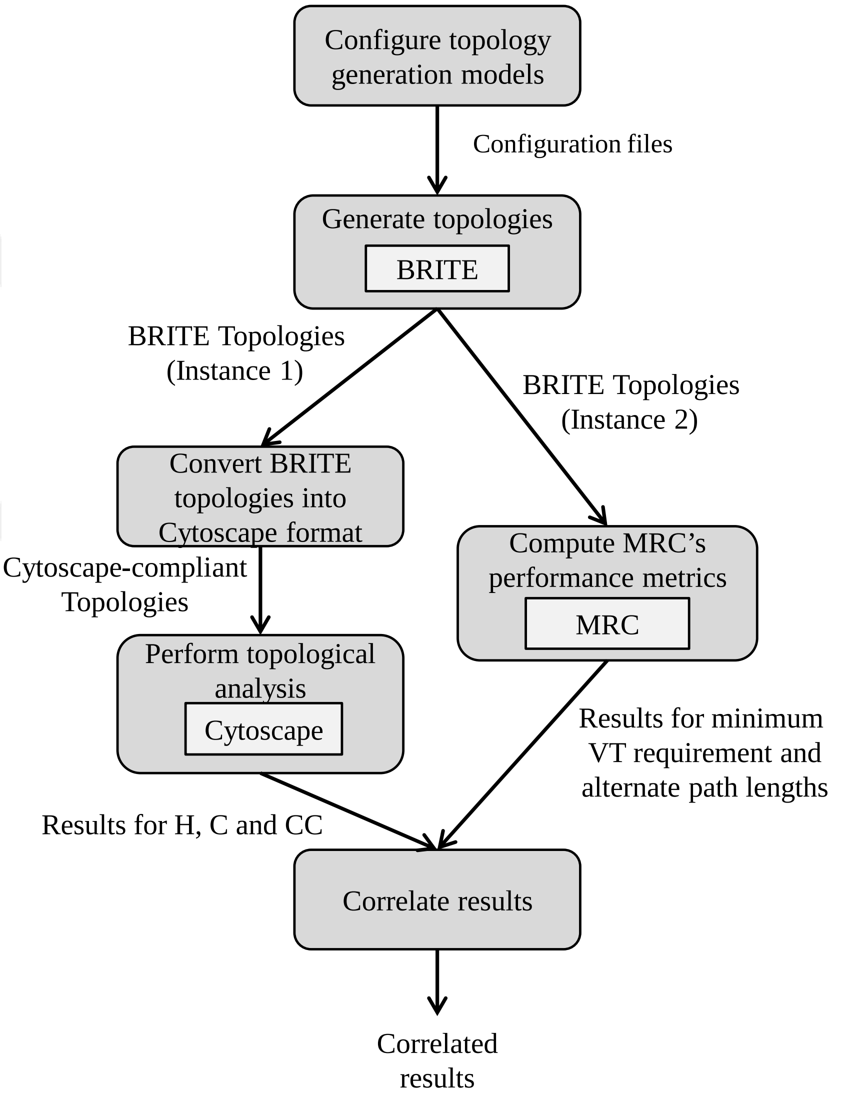

Configure topology generation models
Configuration files
Generate topologies
BRITE
BRITE Topologies (Instance 1)
BRITE Topologies (Instance 2)
Convert BRITE topologies into Cytoscape format
Cytoscape-compliant Topologies
Compute MRC's performance metrics
MRC
Perform topological analysis
Cytoscape
Results for minimum VT requirement and alternate path lengths
Results for H, C and CC
Correlate results
Correlated results

Figure 4.1. Workflow of our topological analysis

We develop our automation tool to perform the topological analysis of the MRC by following the workflow shown in Fig. 4.1, which uses the Boston University Representative Internet Topology Generator (BRITE) tool [28] to generate the topologies in our experiments and the Cytoscape's Network Analyzer [29] to evaluate the topological characteristics. The configuration files of BRITE are created through an interface which collects the values of the configuration settings for the topology generation models. Each configuration file contains *key-value* pairs which are used to specify the name of the configuration setting along with its value to be used in the generation of the topologies [79]. The configuration settings include *n* (i.e., number of nodes), *m* (i.e., number of links per node), the topology generation model to be used along with its specific parameters, and the link attributes such as bandwidth and delay.

BRITE is executed to randomly generate a user-defined number of topologies in a special format based on the configuration files. Each BRITE-formatted topology contains three sections listing information for the topology generation model used, the nodes and the links [79]. The topologies in BRITE format are converted into Cytoscape-compliant topologies [29] in parallel with the computation of the MRC's performance metrics including minimum VT requirement and alternate path lengths. Cytoscape's Network Analyzer is executed on each topology to compute the network characterizing metrics including *heterogeneity (H), centralization (C),* and *clustering coefficient (CC)*. Finally, we correlate the results for the minimum VT requirement and alternate path lengths with the results for *H*, *C*, and *CC*.

4.1.2. Topological Characteristics In Our Workflow

We compute the metrics of *H, C,* and *CC* for the topologies in our experiments which provide insight into how network density is distributed over the nodes in a topology. These metrics can be efficiently computed by Cytoscape's Network Analyzer, and this efficient computation is important for us since our topological analysis results are obtained using an extensive number of topologies.

H of a network reflects the inclination of the topology to have *hub* nodes, whose degree is much higher than the rest of the network, so that a network becomes more heterogeneous if it has more hub nodes. *H* is defined in [30] as:

$$H = \sigma_d / \mu_d \tag{4.1}$$

where σ_d and μ_d are the standard deviation and the mean of the node degree distribution d, respectively. For the topology in Fig. 4.2, the degree of the nodes $R1$ through $R7$ is 3 whereas the degree of $R8$ is 7. Hence, μ_d is equal to $(3+3+3+3+3+3+3+7)/8 = 3.5$. The variance of the degrees is computed to be $(7 \times (3-3.5)^2 + (7-3.5)^2)/8 = 1.750$. Then, σ_d becomes $\sqrt{1.750} = 1.322$. Finally, the H value for the topology is the ratio $1.322/3.5 = 0.377$.

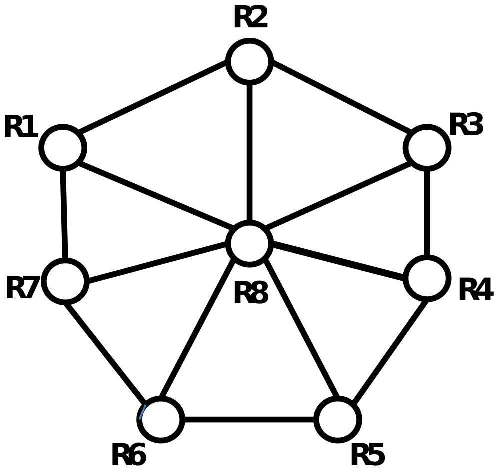


Figure 4.2. An example topology to analyze the topological characteristics

C of a network is an indicator of the compactness of the network, where a node located at a minimum distance from all other nodes is considered to be the most central [30, 80]. C is defined in [30] as:

$$C = \frac{n}{n-2}\left(\frac{d_{max}}{n-1} - \delta\right) \tag{4.2}$$

where n, d_{max} and δ represent the number of nodes in the topology, the largest node degree and the network density, respectively. δ in Eq. **4.2** is defined in [30] as:

$$\delta = \frac{\mu_d}{n-1} \tag{4.3}$$

where μ_d is the mean of the node degree distribution d. For example, the δ of the topology in Fig. 4.2 is $3.5/7 = 0.5$, where $\mu_d = 3.5$. Since d_{max} is equal to the degree of $R8$, namely 7, the C value is computed as $(8/6) \times (7/7 - 0.5) = 0.666$ using Eq. **4.2**.

CC of a node u is a measure of the inter-connectivity of u's neighbours, and is computed as the ratio ϕ_u/ω_u where ϕ_u is the number of actual links interconnecting u's neighbors while ω_u is the number of all links interconnecting u's neighbors if the topology were fully connected. CC of a network topology is the average of the CC values of all nodes, and is defined in [81] as:

$$CC = \frac{\sum_{u\in V}(\phi_u/\omega_u)}{n} \tag{4.4}$$

where V is the set of all nodes in the topology. For example, the number of the links interconnecting the neighbors of $R8$ in Fig. 4.2, which corresponds to the the length of the outer cycle, is 7. On the other hand, if the topology was fully connected, the number of the links interconnecting $R8$'s neighbors would be $6\times 7/2 = 21$. Therefore, CC of $R8$ is equal to $7/21 = 0.333$. The CC values for the remaining nodes are equivalent, namely, $2/3 = 0.666$. Since the CC value for the entire topology is the average of the CC values for all nodes, the CC value for the network becomes $(7\times 0.666+0.333)/8 = 0.624$.

4.1.3. Topology Generation Models In Our Workflow

Our workflow adopts all the existing router-level topology generation models provided by BRITE, and selectively configures them to create networks with different topological characteristics. These models include *Waxman* [28], *Barabasi-Albert (BA)* [82], *Barabasi-Albert2 (BA2)* [83], and *Generalized Linear Preference (GLP)* [84]. BA, BA2, and GLP models rely on the concept of *preferential connectivity* which is widely used to generate power-law topologies. Preferential connectivity reflects the inclination of a new node to interconnect with an existing node with the highest degree during the topology construction process. It is underlined in [85, 86] that the degree distributions of the routers on Internet exhibit power-law characteristics.

The models used in our workflow define the interconnections among nodes according to certain probability functions which are defined as follows:

- *Waxman:* This model uses a probability function, which is mainly based on the randomly determined physical distances among the nodes to be created, to decide if there exists a link between each node pair. Its probability function is defined in [28] as:

$$P(x,y) = \alpha e^{-D/(\beta L)}, 0 < \alpha, \beta \leq 1 \qquad \textbf{(4.5)}$$

where x and y are the nodes to be interconnected, α and β are the model-specific parameters, D is the physical distance between x and y, and L is the maximum distance between any two nodes.

- *BA*: This model interconnects a new node x to an existing node y using the following probability function which is defined in [28] as:

$$P(x,y) = \frac{d_y}{\sum_{u \in V} d_u} \qquad \textbf{(4.6)}$$

where d_y and V denote the degree of the node y and the set of the nodes already added to the topology, respectively. The denominator in Eq. **4.6** is the sum of the degrees of all the nodes in V.

- *BA2:* This model expands an initially created core topology by either adding new links with the probability of p ($0 \leq p < 1$), or rewiring the existing nodes with the probability of q ($0 \leq q < 1 - p$), or adding new nodes with the probability of $1 - p - q$ in each iteration. It decides the end-point y of a newly added link originating from an existing node x using the following probability function which gives smaller preferance to the high degree nodes compared to Eq. **4.6** [83]:

$$P(x,y) = \frac{d_y + 1}{\sum_{u \in V} (d_u + 1)} \qquad \textbf{(4.7)}$$

- *GLP:* Similar to BA and BA2 models, this model expands a core topology by adding new links with the probability of p ($0 \leq p < 1$) or by adding new nodes with the probability of $1 - p$ in each iteration. However, GLP utilizes a generalized probability function originally defined in [84]:

$$P(x,y) = \frac{d_y - \beta}{\sum_{u \in V} (d_u - \beta)} \qquad \textbf{(4.8)}$$

where $\beta < 1$ is a model specific parameter. As β grows, the probability of a new link to connect to higher degree nodes also increases. Eq. **4.8** is reduced to Eq. **4.6** and Eq. **4.7** for $\beta = 0$ and $\beta = -1$, respectively.

4.2. Experimental Results

Waxman, BA, BA2, and GLP models are utilized to generate topologies with varying *H*, *C*, and *CC* values to evaluate the topological dependency of the MRC. We observe the following trends in structural properties of the topologies in our experiments: *(i)* When the number of nodes (n) varies from 100 to 300, the generated topologies exhibit increasing trend in *H* and decreasing trends in *C* and *CC*, respectively, *(ii)* When the number of links per node (m) varies from 2 to 4, the generated topologies exhibit decreasing trend in *H* and increasing trends in *C* and *CC*, respectively, *(iii)* When the parameter β in the GLP model varies from 0.1 to 0.6, the generated topologies exhibit increasing trends in all metrics.

Table 4.1. Parameters used to generate our topology pool

Model	***Fixed Model Parameters***	***Varying Model Parameters***
Waxman	$\alpha = 0.15$, $\beta = 0.2$	$n \rightarrow 100, 200, 300$ $m \rightarrow 2, 3, 4$
BA	–	$n \rightarrow 100, 200, 300$ $m \rightarrow 2, 3, 4$
BA2	$p = 0.001$, $q = 0$	$n \rightarrow 100, 200, 300$ $m \rightarrow 2, 3, 4$
	$p = 0.1$, $m = 2$	$n \rightarrow 100, 200, 300$ $q \rightarrow 0, 0.05, 0.1, 0.15$
GLP	$p = 0.2$, $\beta = 0.6$	$n \rightarrow 100, 200, 300$ $m \rightarrow 2, 3, 4$
	$p = 0.2$, $m = 2$	$n \rightarrow 100, 200, 300$ $\beta \rightarrow 0.1, 0.2, 0.4, 0.6$

Table 4.1 shows the different parameter settings used by the topology generation models in our experiments. In each model, certain parameters are systematically varied to observe how the structures of the various topologies affect the performance of the MRC. The constant values used in Table 4.1 lead to the generation of topologies which better expose the trends in the MRC 's performance. For example, in Waxman model, 9 different subsets of topologies are created using the constant values 0.15 and 0.2 for α and β, respectively,

since 9 different pairs of values are used for n and m. Since we generate 100 topologies using each pair of values, the total number of generated topologies becomes $9 \times 100 = 900$ for Waxman. The pairs of values for varying parameters are similarly determined for other topology generation models. The total number of topologies used in all of our experiments is 6000, which is the sum of 900 for Waxman, 900 for BA, 2100 for BA2 and 2100 for GLP. We report the average $minVT$ of the MRC (ave_{minVT}), the average values of H, C and CC (ave_H, ave_C and ave_{CC}, respectively), and the number of the MRC's partitioning based failures (F_{par}).

ave_{minVT} is defined as:

$$ave_{minVT} = \left(\sum_{l\,=\,2}^{9} f(l) \times l \right) / (100 \; - \; N_{N2C}) \tag{4.9}$$

where l varies from 2 to 9 representing the range for the $minVT$s observed in our experiments, the function $f(l)$ returns the number of topologies for which the minimum VT requirement of the MRC is equal to l, and N_{N2C} denotes the number of non-2-connected (N2C) topologies. The denominator is chosen to be $100 - N_{N2C}$ since each experimented topology pool contains 100 topologies, and N2C topologies are excluded from the computation since the MRC can not be executed on such topologies.

ave_H, ave_C and ave_{CC} are reported for each topology pool with 100 topologies, which are generated as specified in Table 4.1, and are computed as:

$$ave_X = \left(\sum_{T_r \notin N2C} X_{T_r} \right) / (100 - N2C) \tag{4.10}$$

where X represents H, C, or CC while T_r and X_{T_r} are the rth topology ($1 \leq r \leq 100$) and the heterogeneity, centralization or clustering levels of T_r, respectively. N2C topologies are excluded from the computation.

$F_{par}^{T_r \notin N2C}$ is defined to be the number of the cases where the MRC fails to construct a certain number of VTs for the topology $T_r \notin N2C$ since there exists a node in T_r whose isolation disconnects the backbone of each VT. Note that $0 \leq F_{par}^{T_r \notin N2C} < (minVT - 1)$ for T_r. For example, if $minVT = 5$, the MRC fails to construct 2, 3 and 4 VTs, and $F_{par}^{T_r \notin N2C}$ gets

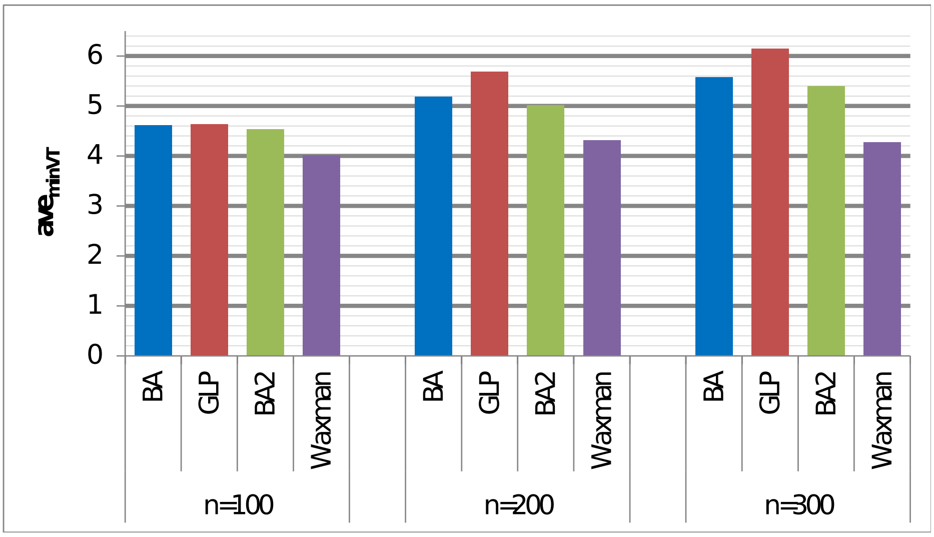


Figure 4.3. Analysis of ave_{minVT} of the MRC with respect to n (**BA:** m=2 **GLP:** m=2, p=0.2, β=0.6 **BA2:** m=2, p=0.1, q=0 **Waxman:** m=2, α=0.15, β=0.2)

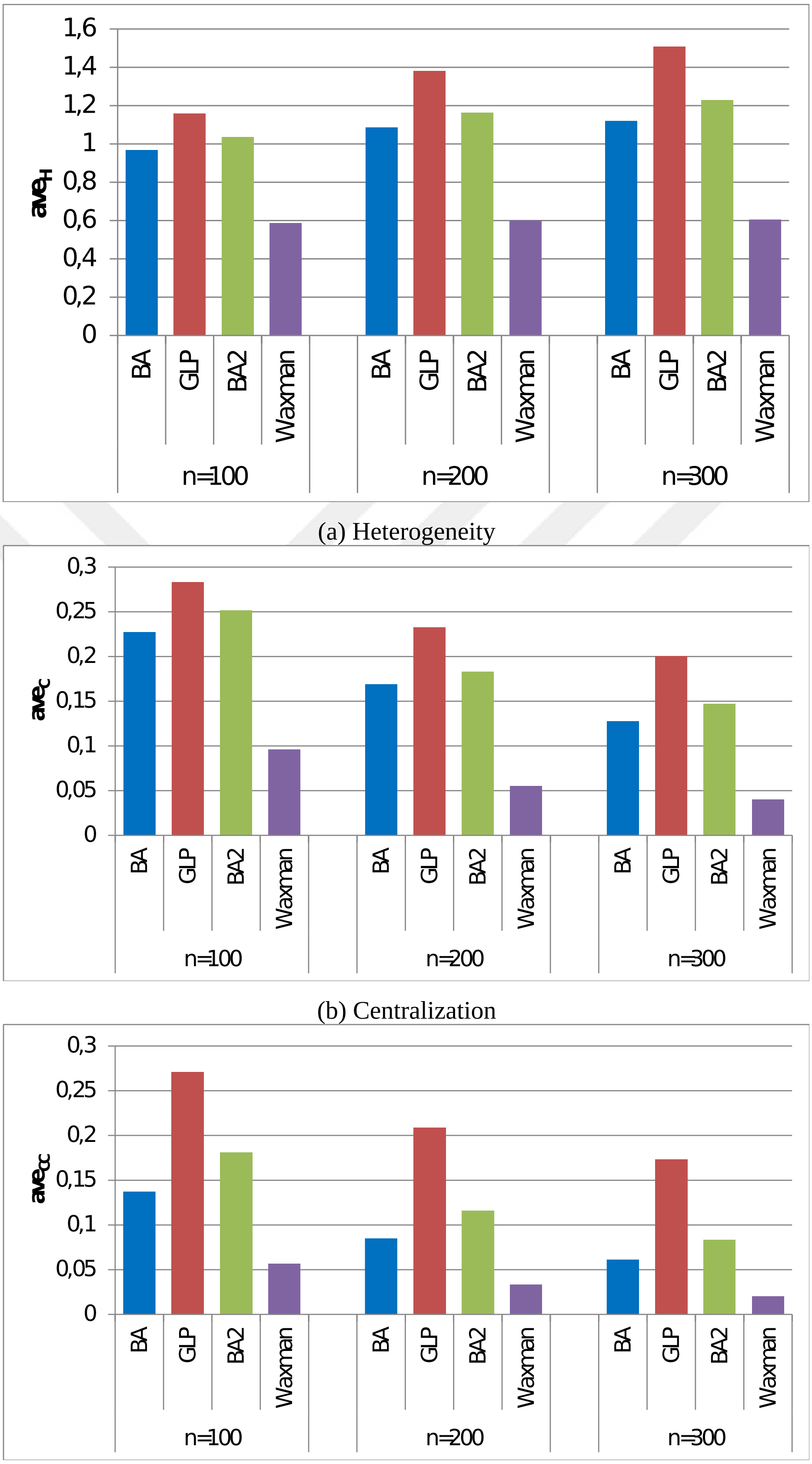
equal to 3 if all these failures are due to partitioning. We report F_{par} for each topology pool with 100 topologies, which is defined as:

$$F_{par} = \sum_{r=1}^{100} F_{par}^{T_r \notin N2C} \qquad \textbf{(4.11)}$$

where $F_{par}^{T_r \notin N2C}$ denotes the number of partitioning based failures for $T_r \notin N2C$.

Fig. 4.3 shows the results for ave_{minVT} when the number of nodes in a network topology (n) varies from 100 to 300. The model-specific constant parameters of each model are listed within the parentheses in the caption of the figure. As shown in Fig. 4.3, apart from Waxman topologies, ave_{minVT} increases as n gets higher. For example, ave_{minVT} becomes 4.63, 5.68 and 6.14 for GLP topologies for $n = 100$, $n = 200$, and $n = 300$, respectively. For Waxman topologies, ave_{minVT} increases when n changes from 100 to 200, but decreases when n changes from 200 to 300.

Fig. 4.4a, Fig. 4.4b, and Fig. 4.4c report the results for ave_H, ave_C, and ave_{CC}, respectively, which are calculated for the same topologies used in Fig. 4.3. Apart from the Waxman topologies, when n becomes higher, ave_C, and ave_{CC} decrease while ave_H increases. For example, for the GLP model, ave_H, ave_C, and ave_{CC} are 1.15, 0.28, and 0.27 for n=100 while they are 1.50, 0.20, and 0.17 for n=300, respectively. Note that the



(a) Heterogeneity

(b) Centralization

(c) Clustering Coefficient

Figure 4.4. Analysis of the topological characteristics with respect to n (**BA:** m=2 **GLP:** m=2, p=0.2, β=0.6 **BA2:** m=2, p=0.1, q=0 **Waxman:** m=2, α=0.15, β=0.2)

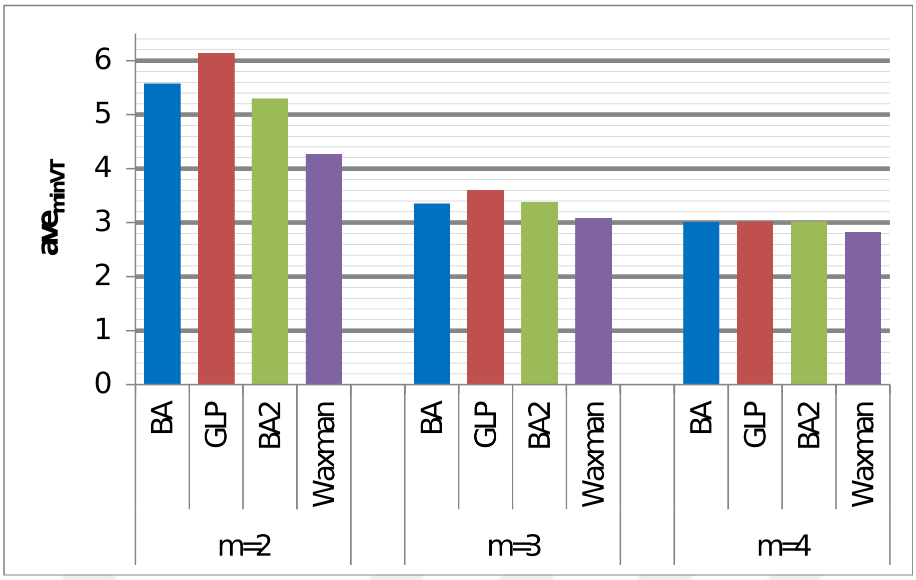
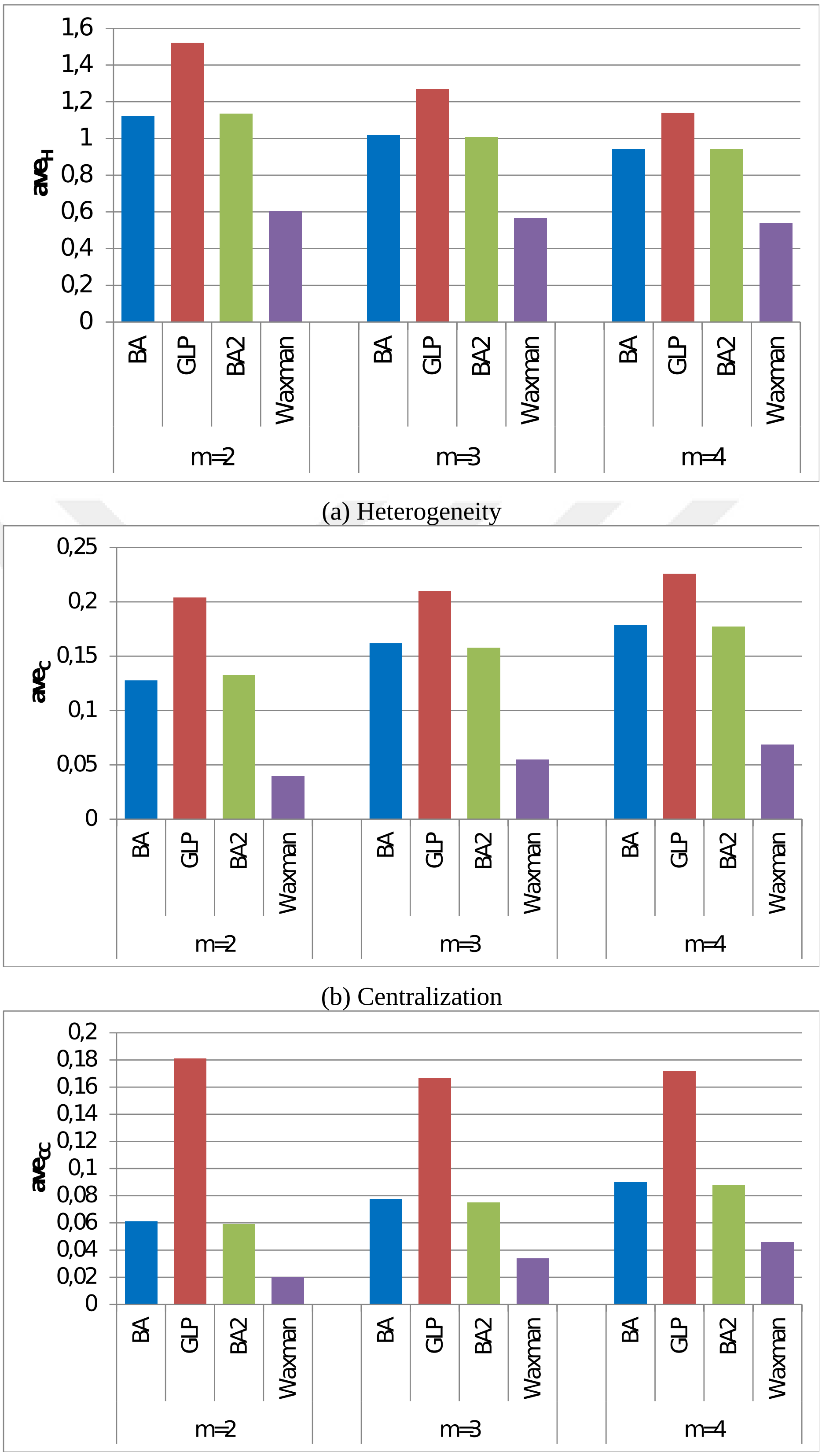


Figure 4.5. Analysis of ave_{minVT} of the MRC with respect to m (**BA:** n=300 **GLP:** n=300, p=0.2, β=0.6 **BA2:** n=300, p=0.001, q=0 **Waxman:** n=300, α=0.15, β=0.2)

order among the magnitudes of ave_H, ave_C, and ave_{CC} reported for different models is the same regardless of n such that $ave_{X(GLP)} > ave_{X(BA2)} > ave_{X(BA)} > ave_{X(Waxman)}$ where $X \in \{H,C,CC\}$. For instance, as shown in Fig. 4.4a, the order among ave_H values when n=300 is 1.50 $(GLP) >$ 1.22 $(BA2) >$ 1.12 $(BA) >$ 0.60 $(Waxman)$. Note that, for n=300, the ave_H value for the Waxman topologies is 40% of the ave_H value for the GLP topologies, which indicates that the structures of any two topologies with the same number of nodes and link densities may be significantly different.

Fig. 4.5 shows the results for ave_{minVT} when the number of links per node (m) varies from 2 to 4. As shown in Fig. 4.5, for all topology generation models, ave_{minVT} decreases as m increases. For example, ave_{minVT} becomes 6.14, 3.60, and 3.02 for the GLP topologies for $m = 2$, $m = 3$, and $m = 4$, respectively. Fig. 4.6a, Fig. 4.6b, and Fig. 4.6c report the results for ave_H, ave_C, and ave_{CC}, respectively, which are calculated for the same topologies used in Fig. 4.5. For each model, excluding the ave_{CC} value for the GLP model, when m becomes higher, ave_C and ave_{CC} increase while ave_H decreases. For example, for the GLP model, ave_H becomes 1.52, 1.27, and 1.13 for the GLP topologies for $m = 2$, $m = 3$, and $m = 4$, respectively.

Table 4.2 shows the results for ave_{minVT}, ave_H, ave_C and ave_{CC} with respect to β in the GLP model for n=300. As shown in the table, ave_{minVT} increases as β increases. It



(a) Heterogeneity

(b) Centralization

(c) Clustering Coefficient

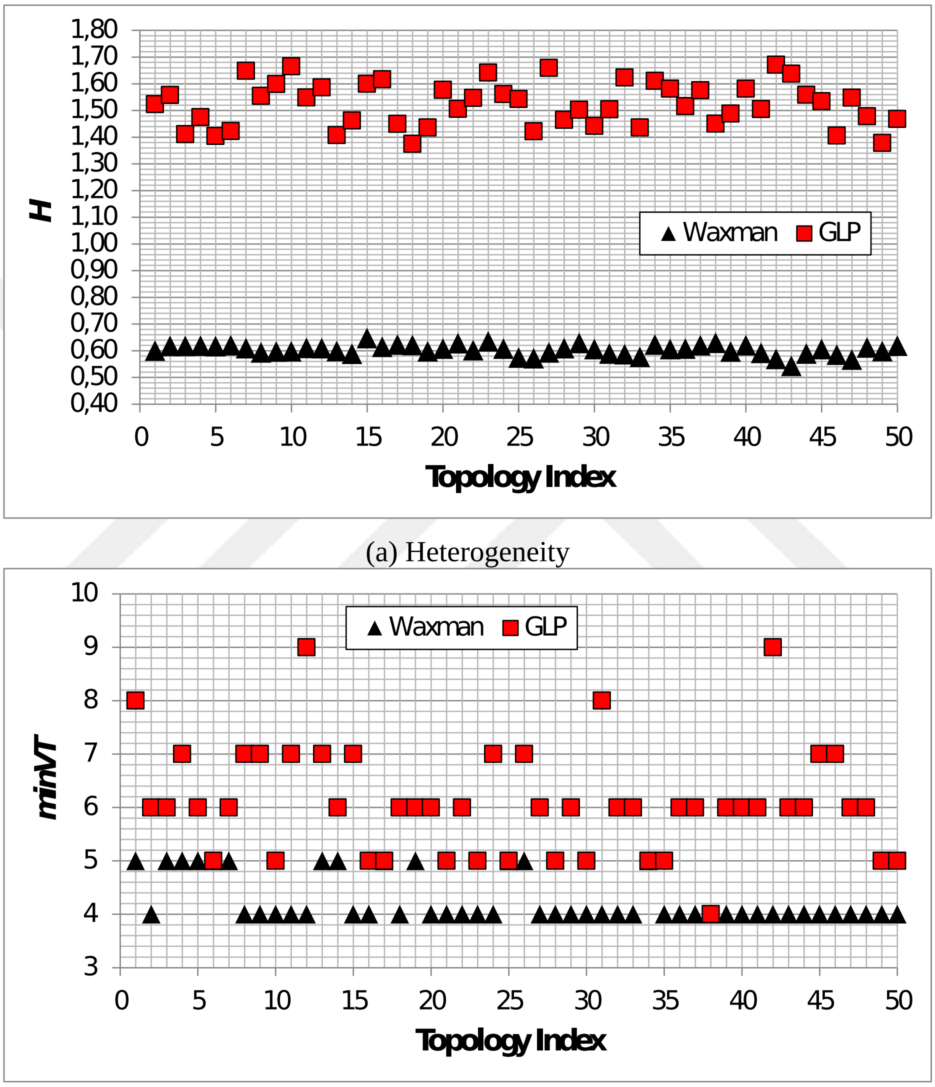
Figure 4.6. Analysis of the topological characteristics with respect to *m* (**BA:** *n*=300 **GLP:** *n*=300, *p*=0.2, β=0.6 **BA2:** *n*=300, *p*=0.001, *q*=0 **Waxman:** *n*=300, α=0.15, β=0.2)

becomes 5.54, 5.58, 5.62, and 6.06 for GLP topologies when $\beta = 0.1$, $\beta = 0.2$, $\beta = 0.4$, and $\beta = 0.6$, respectively. As β becomes higher, ave_H, ave_C and ave_{CC} simultaneously increase. For example, ave_H becomes 1.34, 1.37, 1.44, and 1.51 for GLP topologies when $\beta = 0.1$, $\beta = 0.2$, $\beta = 0.4$, and $\beta = 0.6$, respectively. We also performed other experiments using BA2 and Waxman models. When the parameter q (i.e. the probability of rewiring) of the BA2 model is varied, the results for ave_{minVT} do not exhibit any apparent trend. In Waxman model, we observed that both the heavy-tailed and random node placement schemes yield similar results. The results in Fig. 4.4, Fig. 4.6, and Table 4.2 show that there is an important correlation between ave_{minVT} and ave_H. The MRC requires a higher number of VTs to successfully isolate each node in one of the VTs when a network topology becomes more heterogeneous (i.e., ave_H increases).

Table 4.2. Topological analysis of ave_{minVT} of the MRC with respect to β parameter of GLP for $n = 300$

	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.4$	$\beta = 0.6$
ave_{minVT}	5.54	5.58	5.62	6.06
ave_H	1.34	1.37	1.44	1.51
ave_C	0.17	0.17	0.19	0.20
ave_{CC}	0.12	0.13	0.15	0.17

Fig. 4.7 demonstrates the example distributions for the heterogeneity and the minimum VT requirement over 50 topologies per each of Waxman and GLP models for $n = 300$ and $m = 2$. Note that the Waxman topologies are the least heterogeneous while the GLP topologies are the most heterogeneous in our experiments. Fig. 4.7a shows that GLP topologies have significantly higher H values compared to Waxman topologies such that H ranges from 1.375 to 1.671 for the GLP while it ranges from 0.543 to 0.647 for the Waxman. Fig. 4.7b depicts that the $minVT$ values for the GLP topologies are mostly higher than the $minVT$ for the Waxman such that $minVT$ is equal to 4 or 5 for only 14 topologies out of 50 in case of GLP, and for all the topologies in case of Waxman. $minVT$ values for the remaining GLP topologies range from 6 to 9. These results indicate that the heterogeneity level of a topology is an important factor which influence $minVT$. However, there might be other structural properties as well which affect $minVT$ as shown in Fig. 4.7b such that



(a) Heterogeneity

(b) Minimum VT Requirement

Figure 4.7. Example distributions for *H* and *minVT*

minVT is the same, that is, 4, for the 38th Waxman and GLP topologies even though their corresponding H values are significantly different, namely 0.629 and 1.450, respectively.

Table 4.3 shows that partitioning based failures constitute more than 99% of all failures of the MRC for all topology generation models as indicated within the parentheses. Note that F_{par} is minimum (230) for Waxman topologies, which are the least heterogeneous ones in our experiments ($ave_H = 0.60$), while it is maximum (331) for GLP topologies, which are the most heterogeneous ones ($ave_H = 1.38$). These results indicate that the MRC suffers mostly from the partitioning based failures as the heterogeneity of topologies gets higher. An increase in F_{par} increases the minimum VT requirement. Inspired by the evaluation in this section, we propose a new heuristic in the following section enhancing the MRC.

Table 4.3. Comparison of partitioning based failures of the MRC ($n = 200$, $m = 2$)

Model	#All Failures	F_{par}	ave_H
Waxman	231	230	0.60
BA	318	317	1.08
*BA*2	301	299	1.16
GLP	332	331	1.38

5. MULTIPLE ROUTING CONFIGURATIONS WITH NON-RANDOM NODE ISOLATION

In Chapter 4, we discovered a significant correlation between the performance of the MRC and the topological characteristics through an extensive analysis using our automated topological analysis tool. The MRC needs to construct a higher number of VTs to provide full alternate path coverage if a network topology tends to become more heterogeneous, namely, having more hub nodes whose node degree is much higher than the rest of the network. When the number of VTs gets higher, the time required to compute alternate routing tables, the look-up time for alternate next-hops, and the amount of required Forwarding Information Base (FIB) storage increase [19–22]. We show in Chapter 4 that the major cause for the correlation between the $minVT$ and the heterogeneity level of a network topology is the failures of the MRC due to the parititioning of the VT backbones. Therefore, the $minVT$ of the MRC can be reduced if VT backbones which are more robust against partitioning can be constructed.

In this chapter, inspired by the results of the topological analysis experiments in Chapter 4, we propose an extension to the MRC which takes the node degree information into account. We provide proofs to show why our extension improves the performance of the MRC especially in heterogeneous networks. We show through extensive experiments using both synthetic and real networks that our extension significantly reduces the $minVT$ of the MRC, and, hence, decreases the complexity in terms of processing time and the state requirements of routers. The analysis results confirm the effectiveness of our systematic approach in performing the topological analysis of networking algorithms.

The rest of this chapter is organized as follows. Section 5.1 presents our analysis of the impact of the node isolation order, namely Q_n, on the performance of the MRC. Section 5.2 explains our extension to the MRC. Section 5.3 provides our experimental results.

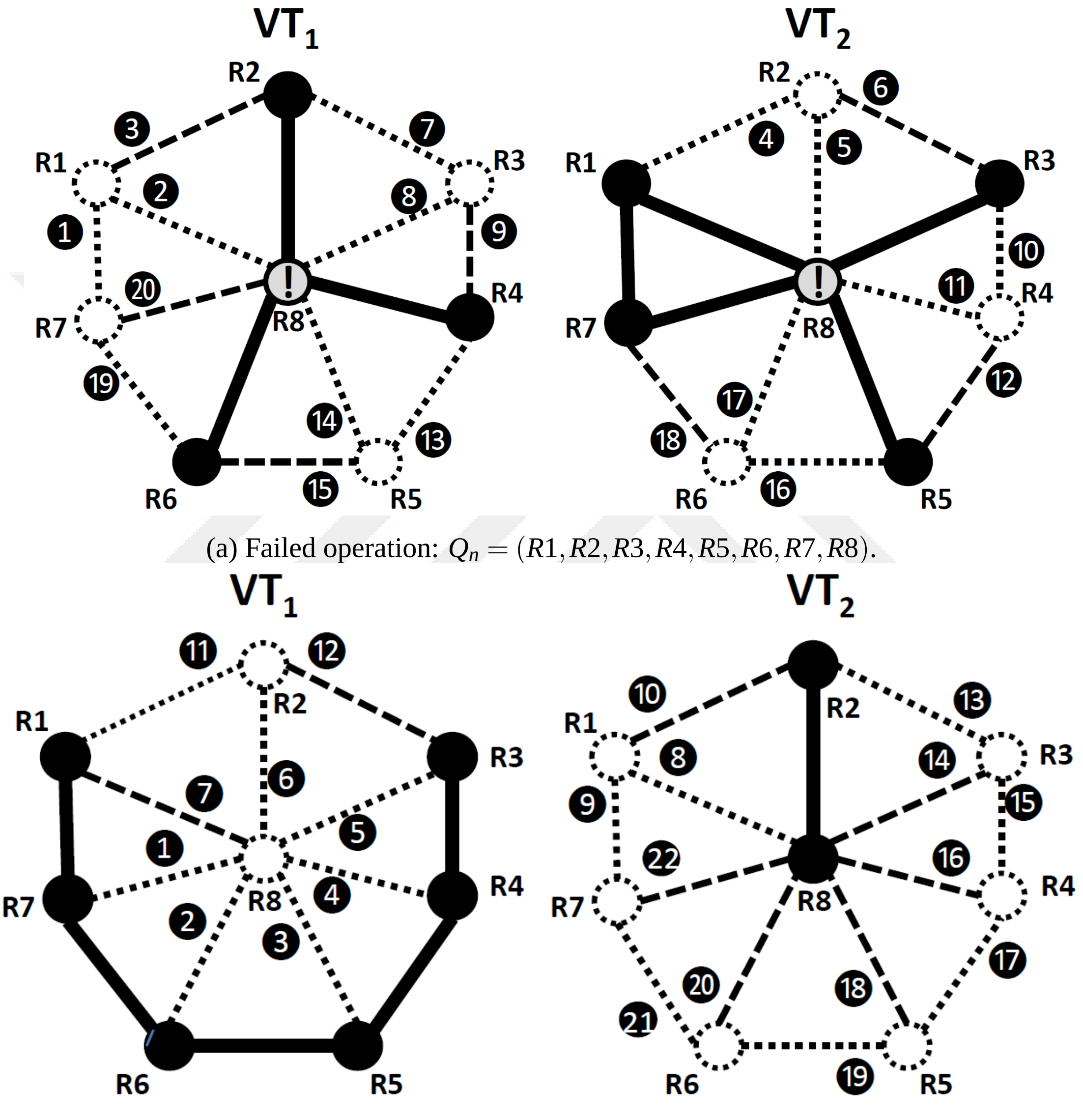
5.1. Analysis of the Impact of Node Isolation Order

The MRC fails to construct a set of VTs if a node cannot be isolated in any VT. In this case, it should be re-configured to use a higher number of VTs, which increases its

$minVT$. The MRC cannot isolate a node u in a VT_i for two reasons [8]: *(i)* The isolation of a neighboring link (u,v) in VT_i causes all the neighboring links of u to become isolated in VT_i, and *(ii)* the removal of u along with its neighboring links disconnects the backbone of VT_i. The former case occurs if all the neighboring links of u except (u,v) have infinite weights in VT_i, and v is already isolated in a VT_j, where (u,v) has the weight of w_r. We show in Chapter 4 that the latter case is the main reason for the majority of the failures of the MRC especially in highly heterogenous topologies. Therefore, the emergence of this type of failure may be prevented if VTs with more robust backbones against partitioning can be constructed, which will in turn reduce the $minVT$. This task can be accomplished if the nodes are intelligently arranged within the Q_n as opposed to the arbitrary arrangement of the Q_n by the MRC. The case *(ii)* may appear more frequently in heterogenous topologies if the hub nodes are located towards the bottom of the Q_n. The isolation of these hub nodes in later iterations may more likely disconnects the VT backbones since this may lead to topologies that are no longer 2-connected.

5.1.1. Example Operations Showing the Impact of Q_n

For the sake of clarification, the failed operation of the MRC to construct 2 VTs is explained using an example topology with 8 nodes and 14 links shown in Fig. 5.1a. The Q_n is assumed to be $(R1,R2,R3,R4,R5,R6,R7,R8)$ where $R8$ is located at the bottom. This example topology is selected since the degree of the nodes $R1$ through $R7$ is 3 while the degree of $R8$ is significantly higher than 3, namely 7, representing an example for a heterogeneous topology. The order, in which the neighboring links of a node are made isolated or restricted, is also arbitrary in the MRC, which is indicated by the consecutive numbers within the black circles over the links. In steps 1 through 20, $R1$, $R2$, $R3$, $R4$, $R5$, $R6$, and $R7$ are successfully isolated in VT_1, VT_2, VT_1, VT_2, VT_1, VT_2, and VT_1, respectively. Note that, as nodes get isolated in the VTs, the connectivity of each VT backbone decreases, and $R8$ tends to play a more important role to keep each VT backbone connected. The attempt to isolate $R8$ fails in both VT_1 and VT_2 since its isolation partitions each VT backbone as indicated by the exclamation marks. The disconnection of all VT backbones causes the MRC to fail to construct 2 VTs. In this case, the MRC should be re-configured for 3 VTs to increase the likelihood of a successful termination. Fig. 5.1b shows the successful operation of the MRC to construct 2 VTs for the same physical topology where the MRC



(a) Failed operation: $Q_n = (R1, R2, R3, R4, R5, R6, R7, R8)$.

(b) Successful operation: $Q_n = (R8, R1, R2, R3, R4, R5, R6, R7)$

Figure 5.1. Impact of the Q_n on the successful termination of the MRC

uses $Q_n = (R8, R1, R2, R3, R4, R5, R6, R7)$. Note that, as opposed to the Q_n in Fig. 5.1a, this Q_n places the hub node $R8$ at the top of the queue, and hence, prioritizes its isolation. In steps 1 through 22, $R8$, $R1$, $R2$, $R3$, $R4$, $R5$, $R6$, and $R7$ are successfully isolated in VT_1, VT_2, VT_1, VT_2, VT_2, VT_2, VT_2, and VT_2, respectively. $R8$ is successfully isolated in VT_1 in this case since no node has been isolated in VT_1 yet, and hence, the connectivity of its backbone is maximum. Note that $R4$, $R5$, $R6$, and $R7$ are isolated in VT_2 since the isolation of any of these nodes in VT_1 partitions its backbone. These example operations of the MRC show that the minimum VT requirement of the MRC significantly depends on the placement of the nodes in the Q_n. Our experiments indicate that a Q_n, which prioritizes the isolation of the high-degree nodes, more likely allows the MRC to successfully construct a fewer number of VTs.

5.1.2. Graph Theoretical Analysis

We analyze the impact of the arrangement of the nodes in the Q_n on VT partitioning in heterogeneous networks using the topology T shown in Fig. 5.2 assuming that the number of nodes on the outer cycle (λ) is sufficiently large. The hub node u_0 of T is connected to every other node while the non-hub nodes ($\{u_i : 1 \leq i \leq \lambda\}$) form a cycle. T possesses the property that each node on the outer cycle has a degree of 3 regardless of λ whereas u_0 has a degree of λ. Note that T is homogeneous for $\lambda = 3$. With an increase in λ, the degree of u_0 increases without changing the degree of u_i ($1 \leq i \leq \lambda$), which results in a more heterogeneous topology.

Proposition 1: For $\lambda > 3$ and the number of VTs configured to be 2, the MRC successfully constructs VTs if and only if u_0 is isolated in a VT within the first four iterations.

Proving the *Proposition 1* requires proving both the *Proposition 1.1* and the *Proposition 1.2.*

Proposition 1.1: For $\lambda > 3$ and the number of VTs configured to be 2, the MRC fails to construct VTs if u_0 is not isolated in a VT within the first four iterations.

Proof 1.1: Let us assume that u_i, u_j, u_k, and u_l ($i \neq j \neq k \neq l$ and $1 \leq i, j, k, l \leq \lambda$) be four successive nodes on the outer cycle, and u_i is randomly selected to be isolated in VT_1 in the first iteration. u_i is successfully isolated in VT_1 since its removal does not partition

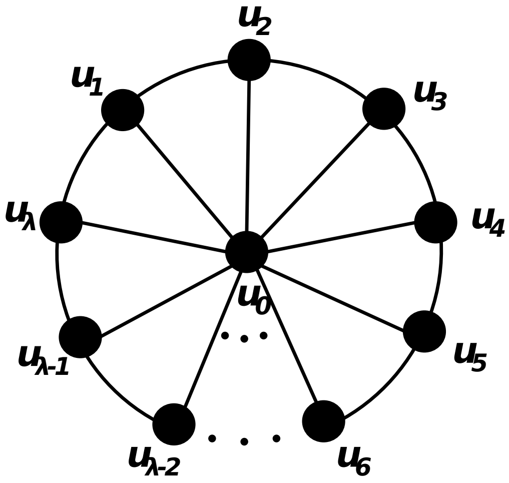
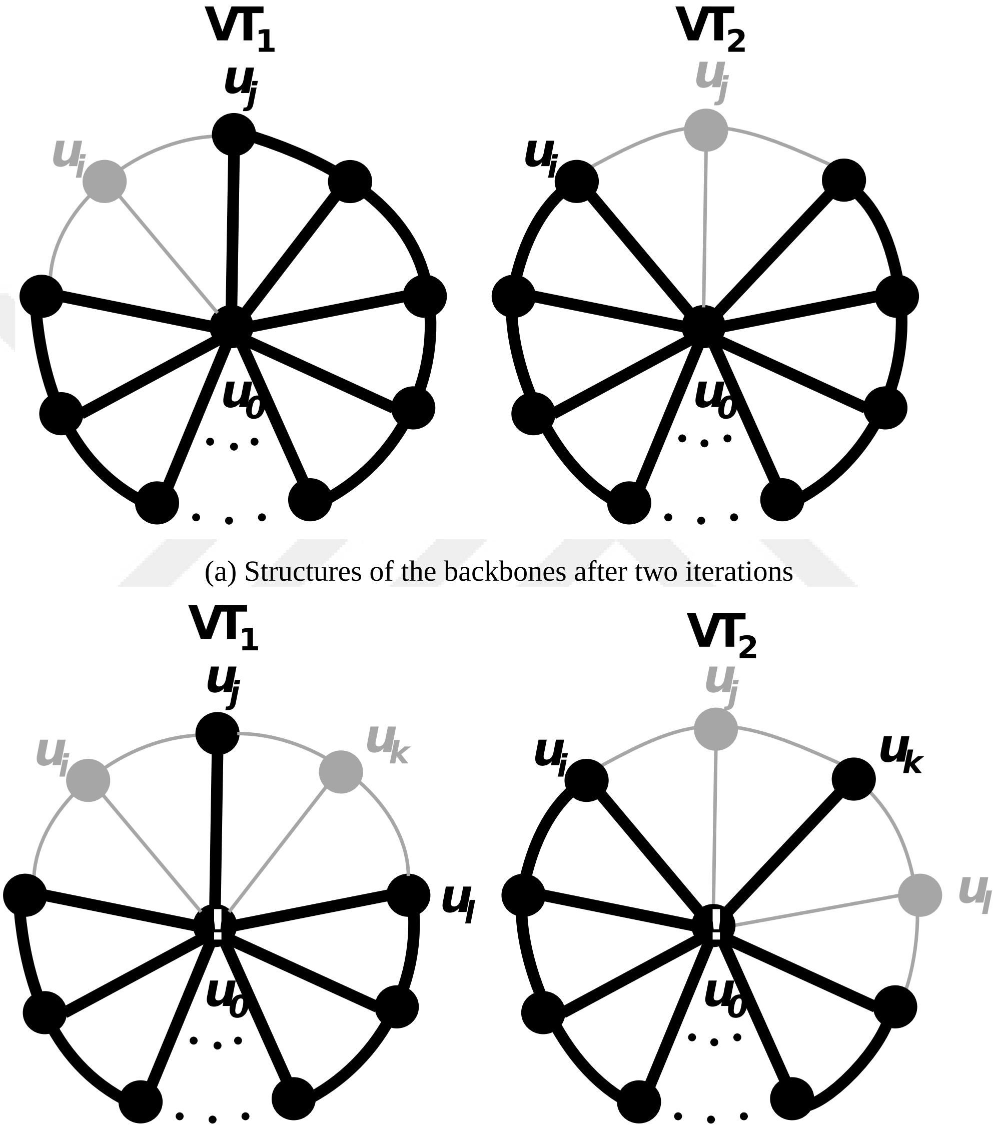


Figure 5.2. An example highly heterogeneous topology

the backbone of VT_1, which is initially 2-connected. Assuming the link (u_i,u_j) becomes restricted in the first iteration, (u_i,u_j) must be isolated along with u_j in the following iteration as explained in Section 3.1. u_j is successfully isolated in VT_2 for the same reason specified for u_i. The structures of VT_1 and VT_2 after the first two iterations are depicted in Fig. 5.3a where the nodes and links which are not part of the backbones are identified by the gray color. Note that the backbones of both VT_1 and VT_2 are still 2-connected at the end of the first two iterations so that they do not contain any articulation point.

Assuming the link (u_j,u_k) becomes restricted in the second iteration, (u_j,u_k) should be isolated along with u_k in VT_1 in the third iteration. Note that the isolation of u_k causes u_0 to become an articulation point in the backbone of VT_1. The link (u_k,u_l) supposedly becomes restricted in the third iteration so that u_l is isolated in VT_2 in the fourth iteration, which causes u_0 to become an articulation point in the backbone of VT_2 as well. The structures of VT_1 and VT_2 after the first four iterations are shown in Fig. 5.3b. u_0 is an articulation point in both VTs at this stage as indicated by the exclamation marks, and, hence, the MRC is guaranteed to fail at a later iteration while attempting to isolate u_0 since its isolation will disconnect u_j and u_k from the rest of the backbones of both VT_1 and VT_2, respectively. ■

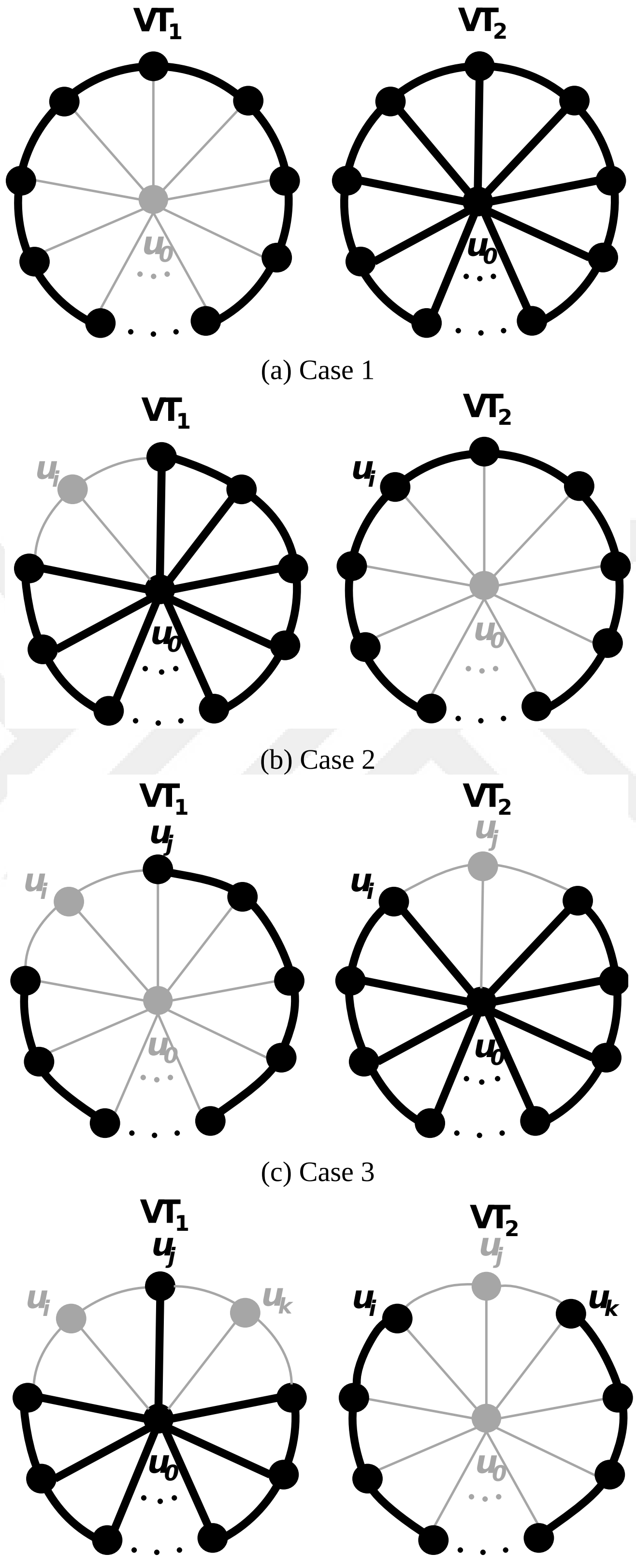
Proposition 1.2: For $\lambda > 3$ and the number of VTs configured to be 2, the MRC successfully constructs VTs if u_0 is isolated in a VT within the first four iterations.



(a) Structures of the backbones after two iterations

(b) Structures of the backbones after four iterations

Figure 5.3. The structures of the backbones of VT_1 and VT_2 after two and four iterations



(a) Case 1

(b) Case 2

(c) Case 3

(d) Case 4

Figure 5.4. Four possible cases for the isolation of u_0 within the first four iterations

Proof 1.2: Let us assume that u_i, u_j, and u_k ($i \neq j \neq k$ and $1 \leq i,j,k \leq \lambda$) be three successive nodes on the outer cycle. Fig. 5.4 demonstrates the four possible cases for the structures of VT backbones following the isolation of u_0 within the first four iterations. Fig. 5.4a, Fig. 5.4b, Fig. 5.4c, and Fig. 5.4d show the respective cases *(1)* u_0 is located at the top of the Q_n so that it is isolated in VT_1 in the first iteration, *(2)* u_i and u_0 are isolated in VT_1 and VT_2 in the first and second iterations, respectively, *(3)* u_i, u_j, and u_0 are isolated in VT_1, VT_2, and VT_1 in the first, second and third iterations, respectively, *(4)* u_i, u_j, u_k, and u_0 are isolated in VT_1, VT_2, VT_1, and VT_2 in the first, second, third and fourth iterations, respectively.

Note that, in cases *(2)*, *(3)* and *(4)*, u_i is isolated in the first iteration since it is located at the top of the Q_n, and the node to isolate in the next iteration is determined as the node which is the end-point of the link which is made restricted in the current iteration. For example, u_0 is isolated in the second iteration in case *(2)* since the link (u_i,u_0) is made restricted in the first iteration. In all cases shown in Fig. 5.4, there exists a VT whose backbone contains the hub node u_0 so that each node of the backbone other than u_0 can reach the other nodes by using u_0 as the next hop. Therefore, the isolation of any node u other than u_0 does not affect the existing paths among the pairs of nodes, and, hence, is guaranteed not to partition the backbone since the backbone remains connected after the isolation of u. For example, the backbone of VT_2 shown in Fig. 5.4a contains u_0, and, hence, all the nodes on the outer cycle can reach the other nodes via u_0. This ensures that any node on the outer cycle can be isolated in VT_2 without partitioning the backbone of VT_2. ■

The neighborhood of each hub node in a heterogeneous network has a structure which resembles the topology T and contains one or more hub nodes such as u_0. A Q_n prioritizing the isolation of high-degree nodes allows the MRC to successfully construct a fewer number of VTs compared to a Q_n which lacks such a prioritization. Note that isolating the hub nodes in a heterogeneous network in earlier iterations allows for constructing VTs which are more robust against network partitioning similar to the cases shown in Fig. 5.4. If a Q_n which does not locate the high-degree nodes towards the top of the queue is employed by the MRC, the earlier node isolations may result in VT backbones where most of the paths passes through the high-degree nodes, which are more probable to get partitioned by the subsequent isolations of these high-degree nodes. This is especially true if the pre-configured number of VTs is not sufficiently large. In such a case, the MRC should be re-configured to use a higher number of VTs which will reduce the likelihood of a VT backbone to get partitioned since each VT backbone will be more connected due to a reduced number of isolated nodes per VT.

5.2. Enhanced MRC Algorithm

5.2.1. Problem Statement

The problem of constructing a set of VTs with the optimum cardinality (k_{opt}), which satisfies the properties *P1*, *P2*, and *P3* described in Section 3.1.1 can be formalized as follows:

Definition. Given a graph $G = (V,E)$ where V and E are the sets of nodes and links, respectively, compute sets of nodes I_i where $2 \leq i \leq k_{opt}$ that satisfy the constraints *(i)* G is still connected when all the nodes in any I_i along with their neighboring links are removed from G, *(ii)* $\bigcup_{i=1}^{k_{opt}} I_i = V$, and *(iii)* k_{opt} is optimum.

In an MT-IPFRR technique, there is generally a trade-off between the number of VTs and the alternate path lengths. Therefore, the construction of a set of VTs with the cardinality k_{opt} minimizes the processing time and state requirements of the routers while it significantly increases the alternate path lengths.

Our heuristics in this study improve the applicability of the MRC to the MT-IPFRR by significantly reducing the $minVT$ while keeping its advantage in terms of alternate path lengths. Similar to the MRC, our heuristics do not attempt to construct a set of VTs with the cardinality k_{opt}, which is NP-complete, and has similarities with the Minimum Set Cover problem [87].

5.2.2. Algorithm

Algorithm 1 describes the operation of our enhanced MRC algorithm (MRC$_{\text{Des+}}$) where $G = (V,E)$ (V: set of vertices, E: set of edges) and k represent the network topology graph and the number of VTs to be constructed, respectively. All VTs are initially the same as the network topology (line 1). $sortNodes()$ function arranges the nodes in V in descending order according to their degrees, and the ordered nodes are assigned to the Q_n such that the topmost element of the Q_n contains the highest-degree node (line 2). The loop between the lines 6-16 iterates through the nodes of the Q_n to isolate them in VTs in a round-robin fashion. $top()$ function returns the topmost node, and then removes it from the Q_n (line 7). $sortLinks()$ function arranges the neighboring links of u, namely $E(u)$, in ascending order according to the degree of the remote end-points, and the ordered links are assigned to $A_{E(u)}$ such that $A_{E(u)}[1]$ contains the link to the lowest-degree neighbor (line 8). Function 1 shows the operation of $sortLinks()$ where $N(l)$ and $D(v)$ denote the remote end-point of the link l and the degree of the node v, respectively. $isolateNode()$ isolates nodes in VTs using the same rules as the MRC, and continues to make each neighboring link isolated starting from $A_{E(u)}[1]$ as long as u has at least one non-isolated neighboring link (line 10). Therefore, since $isolateNode()$ avoids making u inaccessible in a VT by isolating all of its neighboring links, the last member of $A_{E(u)}$ (the link whose remote end-point has the largest degree) is more likely made restricted. Since the isolation of the remote end-point of a restricted link is prioritized in the next iteration if it is not isolated in any VT yet [8], (MRC$_{\text{Des+}}$) more likely continues its execution with the isolation of the highest-degree neighbor. This way, the nodes are isolated according to their locations in the Q_n as much as possible. Otherwise, the actual node isolation order would more frequently deviate from the order specified by the Q_n. If a node can not be isolated in any VT, the MRC fails (line 14).

Algorithm 1 MRC$_{\text{Des+}}$

```
 1: $VT_1 \leftarrow VT_2 \leftarrow \dots VT_k \leftarrow G = (V,E)$          #
    Initialize all VTs
 2: $Q_n \leftarrow$ sortNodes($V$)
 3:
 4: $index \leftarrow 1$
 5: $success \leftarrow$ FALSE
 6: while $Q_n \neq \emptyset$ do
 7:     $u \leftarrow$top($Q_n$)
 8:     $A_{E(u)} \leftarrow$sortLinks($E(u)$)
 9:     while $success$ == FALSE $\wedge$ $index < k$ do
10:         $success \leftarrow$isolateNode($A_{E(u)}$, $VT_{index}$)
11:         $index$++
12:     end while
13:     if $success$ == FALSE then
14:         return FAILURE
15:     end if
16: end while
```

Function 1 `sortLinks(`$E(u)$`)`

```
$A_{E(u)} \leftarrow E(u)$
for $a \leftarrow 1,\ |A_{E(u)}|$ do
    for $b \leftarrow 1,\ |A_{E(u)}| - a$ do
        $v_b \leftarrow N(A_{E(u)}[b])$
        $v_{b+1} \leftarrow N(A_{E(u)}[b+1])$
        if $D(v_b) > D(v_{b+1})$ then
            swap($A_{E(u)}[b]$, $A_{E(u)}[b+1]$)
        end if
    end for
end for
return $A_{E(u)}$                    # $D(N(A_E[1])) \leq D(N(A_E[2])) \leq ... \leq D(N(A_E[|E(u)|]))$
```

5.2.3. Complexity and Convergency Analysis

The worst case running time of the MRC algorithm is bound by $\mathcal{O}(k\Delta|V||E|)$ where Δ represents the maximum node degree [8]. The MRC$_{\text{Des+}}$ does not increase this upper bound since its operations with the asymptotically dominant overhead are inherited from the MRC. The additional overhead of the MRC$_{\text{Des+}}$ results from *sortNodes*() and *sortLinks*() functions, which are bound by $\mathcal{O}(|V|^2)$ and $\mathcal{O}(\Delta^2)$, respectively. Since *sortLinks*() is called for each node, the total additional complexity of the MRC$_{\text{Des+}}$ is bound by $\mathcal{O}(|V|\Delta^2)$. The upper bound $\mathcal{O}(k\Delta|V||E|)$ of the MRC algorithm asymptotically dominates $\mathcal{O}(|V|\Delta^2)$ since both Δ and k is bound by $|V|$.

Upon detection of a failure in the MT-IPFRR, a router activates the VT that does not use the failed component to forward any traffic, and re-computes the VTs using the updated topology. Therefore, the frequency of the re-computation of the VTs depends on the frequency of network failures. In a network where the time interval between successive failures is short, it is critical to complete the construction of the new VTs within this time interval. Since our complexity analysis shows that our heuristic does not increase the upper bound of the MRC, the overhead of our heuristic will be similar to the MRC while it reduces the number of VTs needed for full coverage of network failures.

It is shown in [8] that the MRC terminates successfully for any 2-connected topology as long as k is sufficiently large ($2 \leq k \leq |V|$). Supposed that k is set to $|V|$, the MRC creates $|V|$ virtual topologies by isolating each node in a different virtual topology. Since the VTs

Table 5.1. Different topology pools used in our experiments

Topology Pool	***Subsets***	***Fixed Model Parameters***
P_1	$G^{BA}_{n=100}, G^{BA}_{n=200}, G^{BA}_{n=300}$	$m = 2$
P_2	$G^{BA}_{m=2}, G^{BA}_{m=3}, G^{BA}_{m=4}$	$n = 300$
P_3	$G^{GLP}_{n=100}, G^{GLP}_{n=200}, G^{GLP}_{n=300}$	$p = 0.2$, $\beta = 0.6$, $m = 2$
P_4	$G^{GLP}_{m=2}, G^{GLP}_{m=3}, G^{GLP}_{m=4}$	$p = 0.2$, $\beta = 0.6$, $n = 300$
P_5	$G^{BA2}_{n=100}, G^{BA2}_{n=200}, G^{BA2}_{n=300}$	$p = 0.01$, $q = 0$, $m = 2$
P_6	$G^{BA2}_{m=2}, G^{BA2}_{m=3}, G^{BA2}_{m=4}$	$p = 0.001$, $q = 0$, $n = 300$
P_7	$G^{Waxman}_{n=100}, G^{Waxman}_{n=200}, G^{Waxman}_{n=300}$	$\alpha = 0.15$, $\beta = 0.2$, $m = 2$
P_8	$G^{Waxman}_{m=2}, G^{Waxman}_{m=3}, G^{Waxman}_{m=4}$	$\alpha = 0.15$, $\beta = 0.2$, $n = 300$

initially have the same 2-connected structure as the physical topology, each VT backbone is guaranteed to remain connected by the isolation of a single node. Since the MRC$_{\text{Des+}}$ differs from the MRC only in that it iterates through the nodes of the topology as well as the neighboring links of the currently processed node in a pre-determined order rather than arbitrarily, the MRC's convergence analysis also applies to the MRC$_{\text{Des+}}$.

5.3. Experimental Results

5.3.1. Experimental Setup

We used our automated topology generation and analysis tool introduced in Chapter 4 that integrates BRITE [28] and Cytoscape's Network Analyzer [29] to perform experiments on eight different topology pools that were generated by using four different topology generation models as shown in Table 5.1. In Chapter 4, we performed the topological analysis of the MRC for 6000 topologies that were generated using a diverse range of parameter values for topology generation models. However, we observed that only the fixed parameter values listed in Table 5.1 led the generation of topology pools (P_1 through P_8) with a wide range of heterogeneity levels that clearly exposed a trend in the VT requirement of the MRC with respect to network size (n) and network density (m).

Each topology pool P_i $(1 \leq i \leq 8)$ is the union of subsets $G^M_{n=k}$ or $G^M_{m=j}$. Each subset $G^M_{n=k}$ or $G^M_{m=j}$ forming a P_i contains 100 topologies, where $k \in \{100, 200, 300\}$, $j \in \{2, 3, 4\}$, and $M \in \{$BA, GLP, BA2, Waxman$\}$ represent the value for n (the number of nodes in a topology), *m* (the approximate number of links per node), and the topology generation model, respectively. All the topologies in each subset $G^M_{n=k}$ or $G^M_{m=j}$ were randomly generated by using the same topology generation model *M* and the same *k* or *j* value for *n* or *m*, respectively. The reason for a P_i to contain multiple $G^M_{n=k}$ or $G^M_{m=j}$ subsets where *n* or *m* takes different values was to observe the effects of network size and density on the performance of our algorithms. The parameters specific to topology generation models including *p*, *q*, α and β [28] were set to the constant values given in Table 5.1. For example, P_4 is formed by three subsets such that $P_4 = G^{GLP}_{m=2} \bigcup G^{GLP}_{m=3} \bigcup G^{GLP}_{m=4}$, and, *p*, β and *n* parameters of the GLP model are assigned 0.2, 0.6 and 300, respectively, to generate each of these subsets. Since the cardinality of each subset $|G^M_{n=k}|$ or $|G^M_{m=j}|$ is equal to 100, the number of topologies in a P_i is equal to the number of subsets contained in that P_i multiplied by 100. Therefore, the total number of topologies used in our experiments was 2400. The experiments were also performed using eight different real network topologies that were heavily used in the literature. We believe that experimenting on such a large number of topologies ensures that the performance improvement of our heuristics is validated for a wide range of network deployment scenarios, and hence increases their applicability for real deployments.

To compute the $minVT$ for a topology, the number of VTs, which is an input to our algorithm, was varied starting from 2 until the algorithm successfully terminated for the first time. For each subset with 100 topologies, ave_{minVT} was calculated using Eq. **4.9** in Section 4.2 where *l* changed from 2 to 9 reflecting the range for the $minVT$ observed in our experiments, and the function $f(l)$ returned the number of topologies in $G^M_{n=k}$ or $G^M_{m=j}$ subset that required *l* VTs.

We also report *RAPL* for our experiments defined in Section 3.3.1. Alternate path lengths for a topology were computed by failing each neighboring link of each node, and calculating the resulting alternate paths for the affected destinations on the VT excluding the failed link. For each subset, *RAPL* for the optimum SPF routing was equal to 100%. We compute the average heterogeneity level (ave_H) of each $G^M_{m=j}$ or $G^M_{n=k}$ subset constituting a P_i using Eq. **4.10** in Section 4.2.

In addition to ave_{minVT}, *RAPL*, and ave_H, *improvement percentage (IP)* is reported in our experiments. This metric represents the performance improvement achieved by our heuristics compared to the MRC in terms of the $minVT$. *IP* is calculated as $100 - 100 *$

$ave_{minVT}/ave^{MRC}_{minVT}$ where ave_{minVT} and ave^{MRC}_{minVT} represent the ave_{minVT} values for for our heuristics and the MRC, respectively.

5.3.2. Evaluation of Minimum VT Requirement

Table 5.2a presents the ave_{minVT} values for the MRC and the MRC$_{\text{Des+}}$ along with the corresponding improvement percentages (IP) using the topologies where n is varied from 100 to 300. IP values specify the amount of the reductions in the ave_{minVT} of the MRC. The ave_{minVT} values of the MRC$_{\text{Des+}}$ are also compared to the other variations of the MRC, namely the MRC$_{\text{Des}}$ and the MRC$_{\text{AbsDes}}$, which we developed for our experiments. The MRC$_{\text{Des}}$ arranges the nodes in a topology in descending order according to their degrees like the MRC$_{\text{Des+}}$. However, unlike the MRC$_{\text{Des+}}$, it does not arrange the neighbouring links of the currently processed node in ascending order according to the degrees of the remote end-points. The MRC$_{\text{AbsDes}}$ differs from the MRC$_{\text{Des}}$ and the MRC$_{\text{Des+}}$ in that it never deviates from the initially determined node isolation order while selecting the next node to isolate. We implemented the MRC$_{\text{AbsDes}}$ by significantly modifying the MRC's node isolation logic.

As shown in Table 5.2a, IP values monotonically increase up to 24.95% as n gets higher excluding the Waxman topologies, namely P_7, which are the least heterogeneous in our experiments. Note that the reduction in the ave_{minVT} at these rates means a significant reduction both in the size of the forwarding tables of routers and in the computation time of alternate routing tables. Since a network generally becomes more heterogeneous when n increases, these results indicate that the MRC$_{\text{Des+}}$ performs better when the heterogeneity increases. For example, the ave_{minVT} values of the MRC for the GLP topologies vary as 4.63, 5.68, and 6.14 for $G^{GLP}_{n=100}$, $G^{GLP}_{n=200}$, and $G^{GLP}_{n=300}$, respectively, while the ave_{minVT} values of the MRC$_{\text{Des+}}$ for the same topologies change as 3.81, 4.31, and 4.62, which result in the corresponding IP values of 17.75%, 24.21%, and 24.68%. The comparison results for the MRC$_{\text{Des+}}$, the MRC$_{\text{Des}}$, and the MRC$_{\text{AbsDes}}$ show that the MRC$_{\text{Des+}}$ yields slightly larger ave_{minVT} values than the other two variations only for $G^{BA}_{n=200}$, $G^{BA2}_{n=100}$, and $G^{GLP}_{n=300}$. For example, the MRC$_{\text{AbsDes}}$ achieves an IP value of 25.40 for $G^{GLP}_{n=300}$ which is slightly larger than that of the MRC$_{\text{Des+}}$, namely 24.68. These results indicate that the structure of topologies determines the amount of reductions in the $minVT$ so that the other variations

Table 5.2. ave_{minVT} and its confidence intervals with 95% confidence with respect to n ($m = 2$)

Pool	*Algorithm*	*Subsets In P_i*					
		$G^M_{n=100}$		$G^M_{n=200}$		$G^M_{n=300}$	
		ave_{minVT}	*IP*	ave_{minVT}	*IP*	ave_{minVT}	*IP*
P_1	MRC	4.61	–	5.18	–	5.57	–
	$\text{MRC}_{\text{Des+}}$	3.77	18.22	4.09	21.04	4.18	24.95
	MRC_{Des}	3.82	17.13	4.05	21.81	4.20	24.59
	$\text{MRC}_{\text{AbsDes}}$	3.90	15.40	4.12	20.46	4.27	23.33
P_3	MRC	4.63	–	5.68	–	6.14	–
	$\text{MRC}_{\text{Des+}}$	3.81	17.75	4.31	24.21	4.62	24.68
	MRC_{Des}	3.94	14.75	4.31	24.21	4.68	23.79
	$\text{MRC}_{\text{AbsDes}}$	3.87	16.39	4.35	23.43	4.58	25.40
P_5	MRC	4.53	–	5.01	–	5.39	–
	$\text{MRC}_{\text{Des+}}$	3.71	18.10	3.96	20.95	4.17	22.63
	MRC_{Des}	3.71	18.10	4.01	19.96	4.22	21.70
	$\text{MRC}_{\text{AbsDes}}$	3.58	20.97	4.06	18.96	4.19	22.26
P_7	MRC	4.01	–	4.31	–	4.27	–
	$\text{MRC}_{\text{Des+}}$	3.35	16.45	3.55	17.63	3.73	12.64
	MRC_{Des}	3.54	11.72	3.79	12.06	3.90	8.66
	$\text{MRC}_{\text{AbsDes}}$	3.65	8.97	3.86	10.44	4.01	6.08

(a) Average Minimum VT Requirement

Pool	*Algorithm*	*Subsets In P_i*								
		$G^M_{n=100}$			$G^M_{n=200}$			$G^M_{n=300}$		
		σ^2	*LL*	*UL*	σ^2	*LL*	*UL*	σ^2	*LL*	*UL*
P_1	MRC	0.42	4.48	4.73	0.57	5.02	5.33	0.53	5.42	5.71
	$\text{MRC}_{\text{Des+}}$	0.36	3.65	3.88	0.26	3.98	4.19	0.18	4.09	4.26
	MRC_{Des}	0.31	3.70	3.93	0.20	3.95	4.14	0.22	4.10	4.29
	$\text{MRC}_{\text{AbsDes}}$	0.35	3.78	4.01	0.16	4.03	4.20	0.19	4.18	4.35
P_3	MRC	0.59	4.46	4.80	0.80	5.50	5.87	0.92	5.94	6.34
	$\text{MRC}_{\text{Des+}}$	0.28	3.69	3.92	0.32	4.19	4.43	0.41	4.49	4.76
	MRC_{Des}	0.17	3.85	4.04	0.41	4.17	4.44	0.35	4.55	4.80
	$\text{MRC}_{\text{AbsDes}}$	0.26	3.75	3.98	0.38	4.22	4.48	0.40	4.45	4.71
P_5	MRC	0.43	4.39	4.66	0.41	4.88	5.13	0.54	5.24	5.53
	$\text{MRC}_{\text{Des+}}$	0.20	3.61	3.80	0.22	3.86	4.05	0.18	4.08	4.25
	MRC_{Des}	0.30	3.59	3.82	0.21	3.91	4.10	0.23	4.12	4.31
	$\text{MRC}_{\text{AbsDes}}$	0.26	3.47	3.68	0.19	3.97	4.14	0.25	4.08	4.29
P_7	MRC	0.21	3.91	4.10	0.23	4.21	4.40	0.19	4.18	4.35
	$\text{MRC}_{\text{Des+}}$	0.25	3.25	3.44	0.25	3.45	3.64	0.21	3.63	3.82
	MRC_{Des}	0.27	3.43	3.64	0.18	3.70	3.87	0.09	3.84	3.95
	$\text{MRC}_{\text{AbsDes}}$	0.25	3.55	3.74	0.16	3.78	3.93	0.09	3.95	4.06

(b) Confidence Intervals

of the MRC may also achieve high IP values for certain topologies depending on their structures.

Table 5.2b presents the variance (σ^2) of each $minVT$ distribution used to compute ave_{minVT} and the confidence intervals for ave_{minVT} with 95% confidence using the same topologies in Table 5.2a. LL and UL correspond to the lower and upper limits of a confidence interval, respectively. LL and UL of a confidence interval (LL, UL) for a certain ave_{minVT} are calculated as $ave_{minVT} - U_r$ and $ave_{minVT} + U_r$, where $U_r = t\sigma / \sqrt{100 - N_{N2C}}$, t is the appropriate percentage point for Student's t-distribution with $100 - N_{N2C} - 1$ degrees of freedom [88], σ is the standard deviation of the $minVT$ distribution and $100 - N_{N2C}$ represents the number of the experimented topologies. Our results show that the upper limit of each confidence interval reported for the MRC$_{\text{Des+}}$ is always smaller than the lower limit of the corresponding confidence interval for the MRC, which indicates that the MRC$_{\text{Des+}}$ requires a smaller $minVT$ than the MRC with 95% confidence with respect to n. For example, for $G^{GLP}_{n=300}$, the confidence intervals for the MRC and the MRC$_{\text{Des+}}$ are $(5.94, 6.34)$ and $(4.49, 4.76)$, respectively. The upper limit of each confidence interval for both the MRC$_{\text{Des}}$ and the MRC$_{\text{AbsDes}}$ is also always smaller than the lower limit of the corresponding confidence interval for the MRC. However, the upper limits for these variations are closer to the lower limits for the MRC compared to the MRC$_{\text{Des+}}$ excluding $G^{BA}_{n=200}$, $G^{BA2}_{n=100}$, and $G^{GLP}_{n=300}$.

Table 5.3a lists ave_{minVT} as well as the corresponding IP values with respect to m for the MRC, the MRC$_{\text{Des+}}$, the MRC$_{\text{Des}}$, and the MRC$_{\text{AbsDes}}$. The results show that the ave_{minVT} values for the MRC$_{\text{Des+}}$ are smaller than the ave_{minVT} of the MRC in all cases, and IP values for the MRC$_{\text{Des+}}$ monotonically decrease as m increases, excluding Waxman topologies, namely P_8. Note that the heterogeneity level of a topology decreases with respect to m. For example, ave_{minVT} values of the MRC for the GLP topologies vary as 6.14, 3.60, and 3.02 for $G^{GLP}_{m=2}$, $G^{GLP}_{m=3}$, and $G^{GLP}_{m=4}$, respectively, while ave_{minVT} values of the MRC$_{\text{Des+}}$ for the same topologies change as 4.62, 3.03, and 2.73, which result in the corresponding IP values of 24.68%, 15.80%, and 9.50%. The comparison results for the MRC$_{\text{Des+}}$, the MRC$_{\text{Des}}$, and the MRC$_{\text{AbsDes}}$ show that the MRC$_{\text{AbsDes}}$ yields higher IP values than both the MRC$_{\text{Des+}}$ and the MRC$_{\text{Des}}$ for both $G^{GLP}_{m=2}$ and $G^{GLP}_{m=3}$ while it performs worse for both $G^{Waxman}_{m=3}$ and $G^{Waxman}_{m=4}$ by increasing the ave_{minVT} of the MRC. For example, the MRC$_{\text{AbsDes}}$

Table 5.3. ave_{minVT} and its confidence intervals with 95% confidence with respect to m ($n = 300$)

Pool	*Algorithm*	*Subsets In* P_i					
		$G^M_{m=2}$		$G^M_{m=3}$		$G^M_{m=4}$	
		ave_{minVT}	IP	ave_{minVT}	IP	ave_{minVT}	IP
P_2	MRC	5.57	–	3.35	–	3.01	–
	$\text{MRC}_{\text{Des+}}$	4.18	24.95	3.04	9.25	2.77	7.97
	MRC_{Des}	4.20	24.59	3.01	10.14	2.86	4.98
	$\text{MRC}_{\text{AbsDes}}$	4.27	23.33	3.09	7.76	2.94	2.32
P_4	MRC	6.14	–	3.60	–	3.02	–
	$\text{MRC}_{\text{Des+}}$	4.62	24.68	3.03	15.80	2.73	9.50
	MRC_{Des}	4.68	23.79	3.03	15.80	2.79	7.39
	$\text{MRC}_{\text{AbsDes}}$	4.58	25.40	3.01	16.45	2.73	9.50
P_6	MRC	5.30	–	3.38	–	3.02	–
	$\text{MRC}_{\text{Des+}}$	4.12	22.26	3.04	10.05	2.85	5.62
	MRC_{Des}	4.20	20.75	3.03	10.35	2.80	7.28
	$\text{MRC}_{\text{AbsDes}}$	4.17	21.32	3.05	9.76	2.92	3.31
P_8	MRC	4.27	–	3.08	–	2.83	–
	$\text{MRC}_{\text{Des+}}$	3.73	12.64	3.01	2.27	2.63	7.06
	MRC_{Des}	3.90	8.66	3.02	1.94	2.64	6.71
	$\text{MRC}_{\text{AbsDes}}$	4.01	6.08	3.14	−1.94	2.94	−3.88

(a) Average Minimum VT Requirement

Pool	*Algorithm*	*Subsets In* P_i								
		$G^M_{m=2}$			$G^M_{m=3}$			$G^M_{m=4}$		
		σ^2	LL	UL	σ^2	LL	UL	σ^2	LL	UL
P_2	MRC	0.53	5.42	5.71	0.22	3.25	3.44	0.05	2.96	3.05
	$\text{MRC}_{\text{Des+}}$	0.18	4.09	4.26	0.03	3.00	3.07	0.17	2.68	2.85
	MRC_{Des}	0.22	4.10	4.29	0.01	2.99	3.02	0.12	2.79	2.92
	$\text{MRC}_{\text{AbsDes}}$	0.19	4.18	4.35	0.08	3.03	3.14	0.05	2.89	2.98
P_4	MRC	0.92	5.94	6.34	0.35	3.47	3.73	0.12	2.94	3.09
	$\text{MRC}_{\text{Des+}}$	0.41	4.49	4.76	0.03	2.99	3.07	0.19	2.64	2.82
	MRC_{Des}	0.35	4.55	4.80	0.03	2.99	3.07	0.16	2.71	2.88
	$\text{MRC}_{\text{AbsDes}}$	0.40	4.45	4.71	0.01	2.98	3.03	0.19	2.64	2.82
P_6	MRC	0.51	5.15	5.44	0.31	3.26	3.49	0.06	2.97	3.06
	$\text{MRC}_{\text{Des+}}$	0.20	4.02	4.21	0.03	3.00	3.07	0.12	2.77	2.92
	MRC_{Des}	0.24	4.10	4.29	0.02	2.99	3.06	0.18	2.71	2.88
	$\text{MRC}_{\text{AbsDes}}$	0.18	4.08	4.25	0.04	3.00	3.09	0.09	2.85	2.98
P_8	MRC	0.19	4.18	4.35	0.07	3.02	3.13	0.14	2.75	2.90
	$\text{MRC}_{\text{Des+}}$	0.21	3.63	3.82	0.01	2.99	3.02	0.23	2.53	2.72
	MRC_{Des}	0.09	3.84	3.95	0.01	2.99	3.04	0.23	2.54	2.73
	$\text{MRC}_{\text{AbsDes}}$	0.09	3.95	4.06	0.12	3.07	3.20	0.09	2.87	3.00

(b) Confidence Intervals

increases the ave_{minVT} by 3.88% for $G^{Waxman}_{m=4}$ as indicated by the negative IP value. On the other hand, the MRC$_{\text{Des}}$ yields higher IP values for $G^{BA}_{m=3}$, $G^{BA2}_{m=3}$, and $G^{BA2}_{m=4}$. These results show that the structural properties of topologies may have varying impacts on the performance of different variations of the MRC.

Table 5.3b presents the variance of the $minVT$ values and the corresponding confidence intervals with 95% confidence, which are computed using the same topologies in Table 5.3a. Table 5.3b shows that the upper limit of each confidence interval reported for the MRC$_{\text{Des+}}$ is mostly smaller than the lower limit of the corresponding confidence interval for the MRC. This result indicates that the MRC$_{\text{Des+}}$ mostly requires a smaller $minVT$ with 95% confidence in our experiments with respect to m. For example, for $G^{BA}_{m=2}$, the confidence intervals for the MRC and the MRC$_{\text{Des+}}$ are $(5.42, 5.71)$ and $(4.09, 4.26)$, respectively. Similarly, the upper limit of each confidence interval for both the MRC$_{\text{Des}}$ and the MRC$_{\text{AbsDes}}$ is mostly smaller than the lower limit of the corresponding confidence interval of the MRC.

We also performed numerical experiments to evaluate the minimum VT requirement for the MRC, the MRC$_{\text{Des+}}$, the MRC$_{\text{Des}}$, and the MRC$_{\text{AbsDes}}$ using real network topologies. Table 5.4 shows the $minVT$ values for the real network topologies where Geant and COST239 are used from [21] while the other topologies are used from topology-zoo.org. A single node is removed from Xspedius, Bandcon and Ans to make them 2-connected since single failures can be fully protected only in 2-connected topologies. As shown in Table 5.4, in 7 out of 8 topologies, namely Geant, ATT North America, BT North America, Xspedius, Bandcon, Abilene and Ans, at least one of our MRC variations achieved a smaller $minVT$ compared to the MRC. For example, the $minVT$ values for the MRC$_{\text{Des+}}$, the MRC$_{\text{Des}}$, and the MRC$_{\text{AbsDes}}$ vary as 2, 3, and 3, respectively, for ATT North America while the $minVT$ for the MRC is 4. The performance improvement achieved by the MRC$_{\text{Des+}}$ is significant particularly in ATT North America and BT North America since it reduced the $minVT$ from 4 to 2 and from 6 to 4, respectively. Note that ATT North America and BT North America have the highest heterogeneity levels in our experiments, namely 0.502 and 0.567. These results indicate that isolating the nodes of a topology beginning from the highest-degree node mostly provides a reduction in the $minVT$ for real networks even though the amount of reduction may vary depending on the topological structure.

Table 5.4. Evaluation of minimum VT requirement using real networks

Network	|*V*|	|*E*|	*H*	*C*	*CC*	MRC	$\text{MRC}_{\text{Des+}}$	MRC_{Des}	$\text{MRC}_{\text{AbsDes}}$
Geant	19	30	0.474	0.301	0.063	5	4	4	4
ATT North America	25	56	0.502	0.250	0.556	4	2	3	3
BT North America	36	76	0.567	0.175	0.323	6	4	4	4
Xspedius	33	48	0.400	0.136	0.022	5	5	4	5
Bandcon	20	27	0.372	0.135	0.093	6	6	4	4
Abilene	11	14	0.196	0.056	0.152	5	5	5	4
Ans	17	24	0.278	0.083	0.255	5	5	4	4
COST239	11	26	0.130	0.156	0.439	2	2	2	2

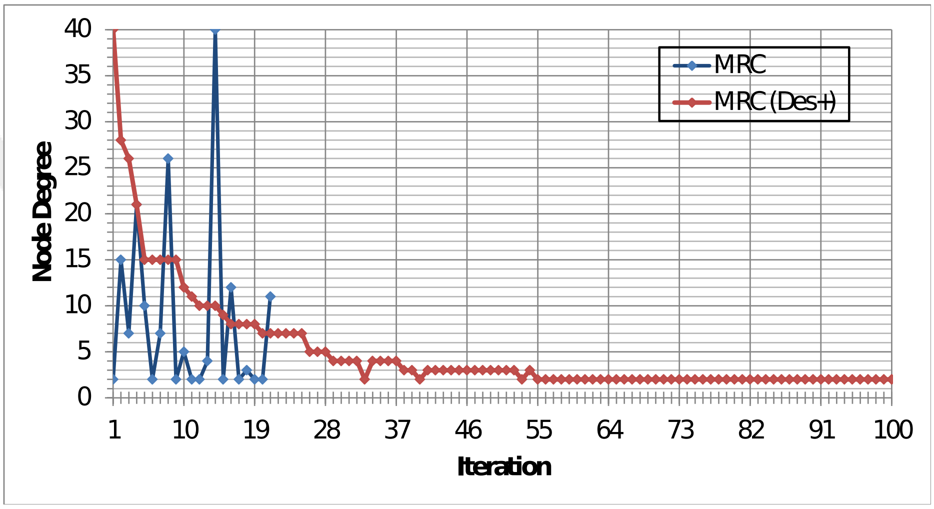


Figure 5.5. Iteration distribution for the node degrees

We analyze the operations of the MRC and the $\text{MRC}_{\text{Des+}}$ using an example 100-node GLP topology to demonstrate the impact of node isolation order on a successful termination. Fig. 5.5 shows the degree distributions of the isolated nodes for the MRC and the $\text{MRC}_{\text{Des+}}$ when they are both configured to construct 3 VTs. Since the isolation of 21st node by the MRC partitions all VTs, and, hence, the MRC fails, the degree distribution of the MRC for the remaining isolations can not be provided. On the other hand, the $\text{MRC}_{\text{Des+}}$ successfully constructs the VTs. As shown in Fig. 5.5, the degree distribution for the $\text{MRC}_{\text{Des+}}$ exhibits a decreasing trend while the distribution for the MRC fluctuates since the $\text{MRC}_{\text{Des+}}$ pre-processes the nodes and links to isolate the nodes in descending order according to their degrees, but the MRC's selection of nodes is arbitrary. Note that the $\text{MRC}_{\text{Des+}}$ deviates from the descending order on 33rd, 40th, and 53rd iterations by selecting a node to isolate whose degree does not conform to the descending order.

5.3.3. Evaluation of Partitioning Based Failures

Table 5.5 presents the F_{par} values for different variations of the MRC using the GLP topologies with $n = 200$ and $m = 2$. As shown in the table, the partitioning based failures of the MRC are reduced from 331 to 208, 203, and 211 by the MRC$_{\text{Des+}}$, the MRC$_{\text{Des}}$ and the MRC$_{\text{AbsDes}}$, respectively. Note that the number of all failures for the MRC$_{\text{Des+}}$ and the MRC$_{\text{Des}}$ are equal even though the MRC$_{\text{Des}}$ provides the largest amount of reduction in F_{par}. The MRC$_{\text{AbsDes}}$ performs worse than both the MRC$_{\text{Des+}}$ and the MRC$_{\text{Des}}$ by providing a larger F_{par} of 211. These results indicate that ordering the nodes in a topology according to their degrees prior to the construction of VTs significantly reduces the partitioning based failures of the MRC.

Table 5.5. Comparison of the number of partitioning based failures for the GLP topologies ($n = 200$, $m = 2$)

Algorithm	#All Failures	F_{par}
MRC	332	331
MRC$_{\text{Des+}}$	208	208
MRC$_{\text{Des}}$	208	203
MRC$_{\text{AbsDes}}$	212	211

5.3.4. Evaluation of Alternate Path Lengths

Table 5.6 shows the relative alternate path lengths (*RAPL*) of the MRC or the MRC$_{\text{Des+}}$ with respect to n and m, which are computed using the GLP topologies. The alternate path lengths are reported relative to the re-converged optimum SPF routing. As shown in the table, both the MRC and the MRC$_{\text{Des+}}$ achieves alternate path lengths, which are close to the optimum SPF routing, namely 100%. For example, the MRC$_{\text{Des+}}$ achieves the *RAPL* values of 108.36%, 107.18%, and 106.71% for $G^{GLP}_{n=100}$, $G^{GLP}_{n=200}$, and $G^{GLP}_{n=300}$, respectively, while the MRC achieves the *RAPL* values of 109.02%, 107.05%, and 106.40% for the same topologies. These results indicate that the MRC$_{\text{Des+}}$ significantly reduces the minimum VT requirement of the MRC at no cost of increasing its alternate path lengths.

Table 5.6. Alternate path lengths for the GLP topologies

Algorithm	$G^{GLP}_{n=100}$	$G^{GLP}_{n=200}$	$G^{GLP}_{n=300}$
MRC	109.02	107.05	106.40
MRC$_{\text{Des+}}$	108.36	107.18	106.71

(a) With respect to *n*

Algorithm	$G^{GLP}_{m=2}$	$G^{GLP}_{m=3}$	$G^{GLP}_{m=4}$
MRC	106.40	111.73	114.46
MRC$_{\text{Des+}}$	106.71	109.92	110.11

(b) With respect to *m*

6. TOPOLOGY-AWARE MULTIPLE ROUTING CONFIGURATIONS

In Chapter 5, we proposed a new heuristic algorithm enhancing the MRC which constructs VTs whose backbones are more robust against partitioning. This task was accomplished by arranging the nodes in Q_n according to the node degree information. VTs with robust backbones can also be constructed by intelligently selecting the VT to isolate a node as opposed to the case as in the MRC where the nodes are isolated in VTs in a round-robin fashion.

In this chapter, we propose two variants of the MRC, namely mMRC-1 and mMRC-2, which take the structural properties of the VTs into consideration during their construction to reduce the $minVT$. Extensive simulations show that our proposed heuristics retain the alternate path lengths within acceptable limits. Similar to our heuristic proposed in Chapter 5, our heuristics do not attempt to find the set of VTs with the optimum cardinality, which is an NP-complete problem [19,21]. Instead, we develop two variants of the MRC that more intelligently construct VTs to increase their robustness against network partitioning. During the VT construction process, mMRC-1 keeps the connectivity of the VTs as high as possible by using the node degree information, whereas mMRC-2 reduces the heterogeneity level of the VTs. Using our automated topology generation and analysis tool, our experiment results have been obtained on 3200 topologies with diverse structural properties. Numerical results show that our heuristics consistently require a smaller number of VTs than the MRC for all experiments using the synthetic topologies. mMRC-1 reduces the VT requirement of the MRC up to 31.84% and performs better than mMRC-2 as the heterogeneity level of the topologies in our experiments increases. On the other hand, mMRC-2 provides an improvement of up to 28.44% and achieves higher improvement percentages than mMRC-1 as the heterogeneity decreases. The alternate path lengths of mMRC-1 and mMRC-2 are only slightly higher than the alternate path lengths of the MRC.

The rest of this chapter is organized as follows. Section 6.1 describes our proposed variants of the MRC. Section 6.2 reports the experimental results.

6.1. Topology-Aware MRC Algorithms

Table 6.1. Notation

$G = (V, E)$	Graph of the physical topology (V: set of nodes, E: set of links)
k	Pre-configured number of VTs to be constructed
A_V	Array of ordered nodes
$E(u)$	Set of neighboring links of node u
$A_{E(u)}$	Array of ordered neighboring links of node u
A_{VT}	Array of ordered VTs
$D(u)$	Degree of node u
$\mathbb{N}(u)$	Set of neighboring nodes of node u
$D_{max}(\mathbb{N}(u))$	Maximum neighbor degree within the neighborhood of node u
$B(VT_i)$	Backbone of VT_i
$D_{B(VT_i)}(u)$	Degree of node u (excluding restricted and isolated links) in $B(VT_i)$
$\Delta H(B(VT_i) \backslash u)$	Change in the heterogeneity level of $B(VT_i)$ if node u is isolated in VT_i

In this section, we propose our variants of the MRC, namely mMRC-1 and mMRC-2. Our variants employ a Q_n queue whose members are arranged in descending order according to their degrees. The notation shown in Table 6.1 is used to describe the operations of mMRC-1 and mMRC-2.

6.1.1. mMRC-1

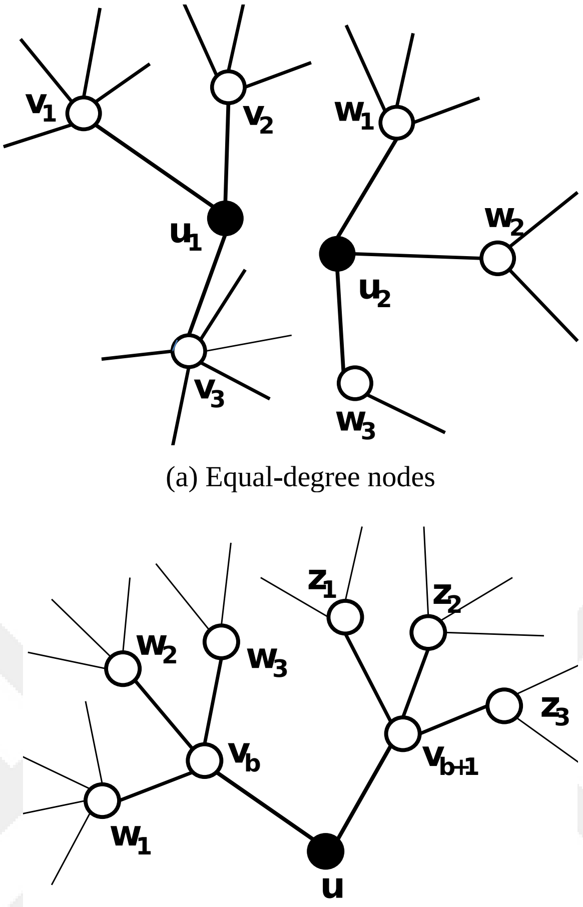
6.1.1.1. Description

Algorithm 1 describes the operation of mMRC-1 where $G = (V, E)$ and k represent the network topology and the number of VTs, respectively. All VTs are initially the same as the network topology (line 1), and these VTs are assigned to the array A_{VT} (line 2). $sortNodes()$ function arranges the nodes in V in descending order according to their degrees, and the ordered nodes are assigned to the Q_n so that the top element of the Q_n contains the highest-degree node (line 3). If there are two equal-degree nodes, the node whose neighbor with the maximum degree among all the neighbors has a higher

Algorithm 1 mMRC-1's VT construction

```
 1: $VT_1 \leftarrow VT_2 \leftarrow \ldots VT_k \leftarrow G = (V,E)$          #
    Initialize all VTs
 2: $A_{VT}[1] \leftarrow VT_1 \ldots A_{VT}[k] \leftarrow VT_k$
 3: $Q_n \leftarrow$ sortNodes($V$)
 4:
 5: for each $u \in Q_n$ do
 6:     $index \leftarrow 1$
 7:     $success \leftarrow$ FALSE
 8:     $A_{E(u)} \leftarrow$sortLinks($E(u)$)
 9:     $A_{VT} \leftarrow$sortVTsV1($A_{VT}, u$)
10:     while $success$ == FALSE $\wedge$ $index <= k$ do
11:         $success \leftarrow$isolateNode($u$, $A_{E(u)}$, $A_{VT}[index]$)
12:         $index$++
13:     end while
14:     if $success$ == FALSE then
15:         return FAILURE
16:     end if
17: end for
```

number of links than the neighbors of the other node, precedes the other node in the ordering. Fig. 6.1a shows an example degree assignment for two equal-degree nodes u_1 and u_2 where $D(u_1) = D(u_2) = 3$. The neighborhoods of u_1 and u_2 are defined by $\mathbb{N}(u_1) = \{v_1, v_2, v_3\}$ and $\mathbb{N}(u_2) = \{w_1, w_2, w_3\}$, respectively. v_3 has the maximum degree in $\mathbb{N}(u_1)$, namely, $D_{max}(\mathbb{N}(u_1)) = D(v_3) = 6$, while w_1 has the maximum degree in $\mathbb{N}(u_2)$, namely, $D_{max}(\mathbb{N}(u_2)) = D(w_1) = 4$. Since $D(v_3) > D(w_1)$, u_1 is placed at a position closer to the top of the Q_n with respect to w_1. Therefore, the isolation of u_1 is prioritized since the loop between lines 5-17 iterates through each node u in the Q_n beginning from the topmost node. *sortLinks*() function arranges the neighboring links of each node u, namely $E(u)$, in ascending order according to the degree of the remote end-points. The ordered links are assigned to $A_{E(u)}$ so that $A_{E(u)}[1]$ contains the link to the lowest degree neighbor (line 8). When there are two links (u, v_b) and (u, v_{b+1}) whose v_b and v_{b+1} end-points have the same degree, (u, v_{b+1}) precedes (u, v_b) in $A_{E(u)}$ if v_{b+1}'s maximum neighbor degree is smaller than the maximum neighbor degree of v_b. Fig. 6.1b demonstrates an example scenario where the end-points v_b and v_{b+1} of two neighboring links of the currently isolated node u have equal degrees, that is, $D(v_b) = D(v_{b+1}) = 4$. The neighborhoods of v_b and v_{b+1} excluding u are defined by $\mathbb{N}(v_b) = \{w_1, w_2, w_3\}$ and $\mathbb{N}(v_{b+1}) = \{z_1, z_2, z_3\}$, respectively. w_1 has the maximum degree in $\mathbb{N}(v_b)$, namely, $D_{max}(\mathbb{N}(v_b)) = D(w_1) = 5$. On the other hand, z_2 has the



(a) Equal-degree nodes

(b) Equal-degree end-points of two neighboring links

Figure 6.1. Tie breaking mechanisms of mMRC-1 and mMRC-2

maximum degree in $\mathbb{N}(v_{b+1})$, namely, $D_{max}(\mathbb{N}(v_{b+1})) = D(z_2) = 4$. Since $D(w_1) > D(z_2)$, the link (u, v_b) is placed at the position with the higher index of $b+1$, namely, $A_{E(u)}[b+1]$, whereas the link (u, v_{b+1}) is located in $A_{E(u)}[b]$.

Function 1 defines $sortVTsV1()$ that arranges A_{VT} in ascending order according to the number of links of the currently isolated node u on the backbone of each VT, namely, $D_{B(VT_i)}(u)$ (line 9). mMRC-1 successively attempts to isolate u in VTs starting from $A_{VT}[1]$ (line 11) until it successfully isolates u. $isolateNode()$ function uses the same node isolation rules as the MRC described in Section 3.1 except that it iterates through the neighboring links of u starting from $A_{E(u)}[1]$ to assign the weight of infinity to as many links as possible while this iteration is random in the MRC. Therefore, for the example scenario in Fig. 6.1b, $isolateNode()$ sets the weight of the link (u, v_{b+1}) to infinity with a higher probability. It is more likely that $isolateNode()$ assigns w_r to the link (u, v_b), and hence, mMRC-1 continues its execution with the isolation of v_b in the next step, which has a relatively higher-degree

Function 1 `sortVTsV1(`A_{VT}, u`)`

```
for $a \leftarrow 1,\ |A_{VT}|$ do
    for $b \leftarrow 1,\ |A_{VT}| - a$ do
        if $D_{B(A_{VT}[b])}(u) > D_{B(A_{VT}[b+1])}(u)$ then
            swap($A_{VT}[b]$, $A_{VT}[b+1]$)
        end if
    end for
end for
return $A_{VT}$                    # $D_{B(A_{VT}[1])}(u) \leq D_{B(A_{VT}[2])}(u) \leq \ldots \leq D_{B(A_{VT}[k])}(u)$
```

neighbor. Note that mMRC-1 with these features will less frequently diverge from the initial node isolation order specified by the Q_n. $isolateNode()$ returns *FALSE* if one of the following three conditions hold: *(i)* u is a neighbor of the previously isolated node v, and $A_{VT}[index]$ is the same as the VT where v is isolated, *(ii)* the link (v,u) has the weight w_r in $A_{VT}[index]$ [8], and *(iii)* the isolation of u disconnects $B(A_{VT}[index])$.

The structures of 3 VTs constructed by mMRC-1 are given in Fig. 6.2 for an example topology with 6 nodes and 9 links. The steps of mMRC-1 to generate these VTs are shown in Table 6.2, where the currently isolated node u, the degree of u in the backbone of each VT ($D_{B(VT_i)}(u)$), the selected VT to isolate u, and the success status of the isolation of u in the selected VT (i.e., T for True, F for False) are listed for each iteration. Initially, the highest-degree node $R4$ has the same degree in each backbone ($D_{B(VT_1)}(R4) = D_{B(VT_2)}(R4) = D_{B(VT_3)}(R4) = 4$) so that VT_1 is randomly selected to isolate $R4$ as indicated by the bold font in the first row of Table 6.2. Note that the link $(R4,R3)$ is assigned the weight w_r in VT_1 as shown by the dashed lines in Fig. 6.2 since this link has the lowest precedence in $R4$'s link isolation order. The next node $R3$ in the node isolation order is also attempted to be isolated in VT_1 since $D_{B(VT_1)}(R3) = 3$ is minimum in step 2. The isolation of $R3$ in VT_1 fails because the link $(R4,R3)$ is already restricted in VT_1, and therefore, $R3$ and $R4$ cannot be isolated in the same VT. In this case, VT_2 is randomly selected to isolate $R3$ since $D_{B(VT_2)}(R3) = D_{B(VT_3)}(R3) = 4$. After 9 iterations, all VTs are successfully generated. As shown in the table, the node isolation order utilized by mMRC-1 is $Q_n = (R4,R3,R2,R5,R6,R1)$, and the nodes in this order are successfully isolated in VT_1, VT_2, VT_3, VT_1, VT_1 and VT_1, respectively. The resulting VTs computed by mMRC-1 are shown in Fig. 6.2 where the sets of isolated nodes are $I_1 = \{R1,R4,R5,R6\}$, $I_2 = \{R3\}$, and $I_3 = \{R2\}$ for VT_1, VT_2 and VT_3, respectively. Note that each isolated node has at least one

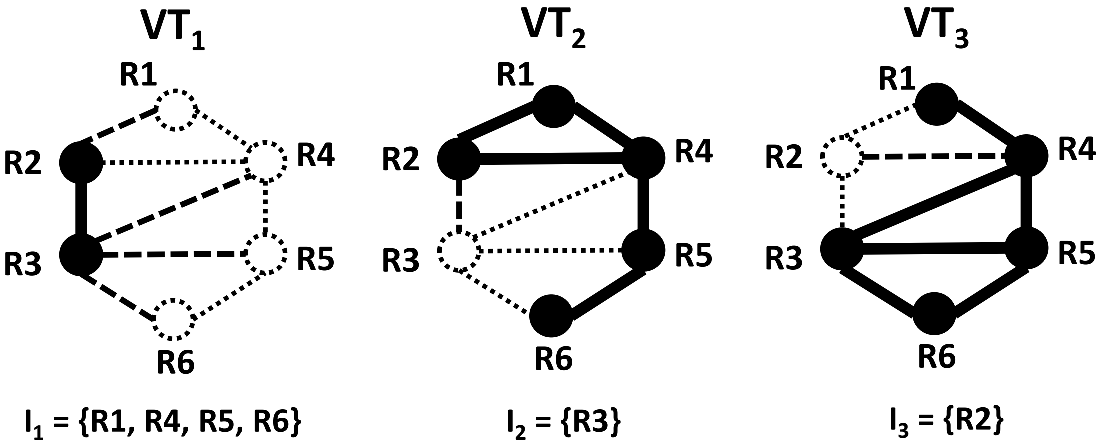


Figure 6.2. Example VTs constructed by mMRC-1

Table 6.2. Operation of mMRC-1 on an example network

Step	*Node u*	$D_{B(VT_1)}(u)$	$D_{B(VT_2)}(u)$	$D_{B(VT_3)}(u)$	*VT*	*Success*
1	*R4*	**4**	4	4	VT_1	*T*
2	*R3*	**3**	4	4	VT_1	*F*
3	*R3*	3	**4**	4	VT_2	*T*
4	*R2*	**2**	2	3	VT_1	*F*
5	*R2*	2	**2**	3	VT_2	*F*
6	*R2*	2	2	**3**	VT_3	*T*
7	*R5*	**2**	2	3	VT_1	*T*
8	*R6*	**1**	1	2	VT_1	*T*
9	*R1*	**1**	2	1	VT_1	*T*

restricted link (dashed line), which connects the isolated node to the VT backbone, and each VT satisfies the properties *P1*, *P2*, and *P3* in Section 3.1.

6.1.1.2. Convergence

It is shown in [8] that the MRC terminates successfully for any 2-connected topologies as long as k is sufficiently large ($2 \leq k \leq |V|$). Supposed that k is set to $|V|$, the MRC creates $|V|$ virtual topologies by isolating each node in a different virtual topology. Since the VTs initially have the same 2-connected structure as the physical topology, each VT backbone is guaranteed to remain connected by the isolation of a single node.

mMRC-1 sorts VTs prior to the isolation of each node. Depending on the VT ordering, two different cases may occur assuming $k = |V|$:

- *Case 1:* If each VT appears as the topmost element $A_{VT}[1]$ in Algorithm 1 for only once during the iterations of the loop between lines 7-17, mMRC-1 operates the same as the MRC by successfully isolating each node in a different VT. In such a case, on any iteration of the loop, the number of VTs, which contain only non-isolated nodes, and hence, have the same 2-connected structure as the physical topology, is equal to the number of the remaining nodes in the Q_n, which are not isolated in any VT yet.
- *Case 2:* If a VT, which already contains an isolated node is re-located at $A_{VT}[1]$, and is re-used for the isolation of another node v, a fewer number of VTs is sufficient for mMRC-1 to successfully isolate all nodes. During the iterations of the loop in Algorithm 1 following the isolation of v, the number of VTs that contain only non-isolated nodes, and have the same 2-connected structure as the physical topology, is higher than the number of the remaining non-isolated nodes in the Q_n since there exists at least one VT that is used to isolate multiple nodes.

Proposition: mMRC-1 always successfully terminates for any 2-connected topology when k is selected to be sufficiently large.

Proof: mMRC-1 extends the MRC by intelligently selecting *(i)* the initial node isolation order indicated by the Q_n in Algorithm 1; *(ii)* the order in which neighboring links of the currently isolated node are iterated through; and *(iii)* the order in which VTs are chosen to isolate a certain node. *(i)* and *(ii)* together determine the actual order in which the nodes are isolated in the VTs. Assuming $k = |V|$, mMRC-1 always succeeds no matter which node isolation order is used since the number of VTs with a 2-connected backbone is always at least equal to the number of the non-isolated nodes in the Q_n as indicated by *Case 1* and *Case 2*. Similarly, for $k = |V|$, regardless of the VT order chosen in *(iii)*, the *index* in Algorithm 1 will be incremented until the current node is successfully isolated in a VT. *Case 1* and *Case 2* guarantee that such a VT exists. ■

6.1.1.3. Complexity

The worst case running time of the MRC algorithm is bound by $\mathscr{O}(k\Delta|V||E|)$ where Δ represents the maximum node degree [8]. mMRC-1 does not increase this upper bound since its operations with the asymptotically dominant overhead are inherited from the MRC. The additional overhead of mMRC-1 results from $sortNodes()$, $sortLinks()$ and $sortVTsV1()$

functions, which are bound by $\mathscr{O}(|V|^2)$, $\mathscr{O}(\Delta^2)$ and $\mathscr{O}(k^2)$, respectively. Since *sortLinks*() and *sortVTsV*1() are called for each node, the total additional complexity of mMRC-1 is bound by $\mathscr{O}(|V|(\Delta^2+k^2))$. The upper bound $\mathscr{O}(k\Delta|V||E|)$ of the MRC algorithm asymptotically dominates $\mathscr{O}(|V|(\Delta^2+k^2))$ since both Δ and k are bound by $|V|$.

6.1.2. mMRC-2

6.1.2.1. Description

Function 2 `sortVTsV2(`A_{VT}, u`)`

```
for $a \leftarrow 1,\ |A_{VT}|$ do
    for $b \leftarrow 1,\ |A_{VT}| - a$ do
        if $\Delta H(B(A_{VT}[b])\backslash u) < \Delta H(B(A_{VT}[b+1])\backslash u)$ then
            swap($A_{VT}[b]$, $A_{VT}[b+1]$)
        end if
    end for
end for
return $A_{VT}$                    # $\Delta H(B(A_{VT}[1])\backslash u) \geq ... \geq \Delta H(B(A_{VT}[k])\backslash u)$
```

mMRC-2 has the same main loop as mMRC-1 described in Algorithm 1. The only difference is that mMRC-2 uses the heterogeneity metric to sort the VTs. Function 2 defines *sortVTsV*2() that arranges A_{VT} in descending order according to the amount of anticipated reductions in their heterogeneity levels. ΔH for a backbone B_i is computed by subtracting the heterogeneity level of B_i after node u is isolated from the current heterogeneity level of B_i. Note that ΔH can take positive or negative values so that a positive ΔH value means a reduction in the heterogeneity level while $-\Delta H$ corresponds to an increase. If u cannot be isolated in a selected VT, mMRC-2 moves to the VT with the next largest ΔH value.

The steps of mMRC-2 to construct the VTs for the same topology in Fig. 6.2 are demonstrated in Table 6.3, where the ΔH value computed for each backbone B_i in the case that u is isolated in B_i is given for each iteration. Since all the VTs initially have identical structures, the removal of the highest-degree node $R4$ from each backbone results in the same ΔH value such that $\Delta H(B(VT_1)\backslash R4) = \Delta H(B(VT_2)\backslash R4) = \Delta H(B(VT_3)\backslash R4) = -0.044$. Note that the ΔH value is negative for each backbone meaning that the removal of $R4$ causes the heterogeneity level of each backbone to increase. Because all ΔH values are equal, VT_1 is randomly selected to isolate $R4$. The next node in node isolation order, namely

Table 6.3. Operation of mMRC-2 on an example network

Step	*Node u*	$\Delta H(B(VT_1)\backslash u)$	$\Delta H(B(VT_2)\backslash u)$	$\Delta H(B(VT_3)\backslash u)$	*VT*	*Success*
1	$R4$	$-\mathbf{0.044}$	-0.044	-0.044	VT_1	T
2	$R3$	$\mathbf{0.316}$	-0.044	-0.044	VT_1	F
3	$R3$	0.316	$-\mathbf{0.044}$	-0.044	VT_2	T
4	$R2$	-0.261	$-\mathbf{0.017}$	-0.061	VT_2	F
5	$R2$	-0.261	-0.017	$-\mathbf{0.061}$	VT_3	T
6	$R5$	-0.017	-0.261	$\mathbf{0.000}$	VT_3	T
7	$R6$	$-\mathbf{0.017}$	-0.037	-0.020	VT_1	T
8	$R1$	-0.020	$-\mathbf{0.017}$	-0.020	VT_2	T

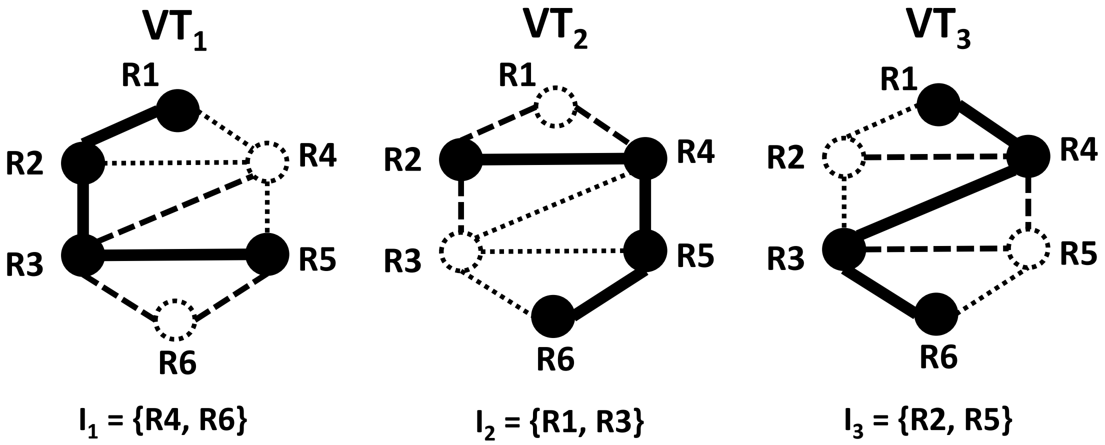


Figure 6.3. Example VTs constructed by mMRC-2

$R3$, is also attempted to be isolated in VT_1 since $\Delta H(B(VT_1)\backslash R3) = 0.316$ is maximum among all VTs in step 2. The isolation of $R3$ in VT_1 fails for the same reasons explained for mMRC-1. In this case, VT_2 is randomly selected to isolate $R3$ since $\Delta H(B(VT_2)\backslash R3) = \Delta H(B(VT_3)\backslash R3) = -0.044$. After 8 iterations, the resulting VTs are shown in Fig. 6.3 where the sets of isolated nodes are $I_1 = \{R4, R6\}$, $I_2 = \{R1, R3\}$, and $I_3 = \{R2, R5\}$ for VT_1, VT_2, and VT_3, respectively. As shown in the table, the node isolation order utilized by mMRC-2 is the same as mMRC-1, but the order of VTs used to isolate the nodes is different from mMRC-1, that is, VT_1, VT_2, VT_3, VT_3, VT_1, and VT_2. Note that each VT in Fig. 6.3 satisfies the properties *P1*, *P2*, and *P3* in Section 3.1.

6.1.2.2. Convergence and Complexity

Since mMRC-2 differs from mMRC-1 only in its methodology to sort VTs which is implemented by $sortVTsV2()$ function, the same convergency analysis performed for mMRC-1 in Section 6.1.1.2 applies to mMRC-2.

The algorithm for mMRC-2 has the same structure as mMRC-1 except the implementation of $sortVTsV2()$, which arranges the VTs in descending order according to the heterogeneity metric, namely, the ratio of the standard deviation (σ_d) of the node degree distribution to its mean node degree (μ_d). The computation of both σ_d and μ_d for a VT requires iterating through each node in the VT along with its neighboring links. Therefore, the computation of the heterogeneity for a VT is bound by $\mathcal{O}(\Delta|V|)$. Since this computation is repeated for each VT, the total overhead resulting from the computation of the heterogeneity for all VTs is bound by $\mathcal{O}(k\Delta|V|)$. mMRC-2 does not increase the upper bound of the MRC since the MRC's upper bound $\mathcal{O}(k\Delta|V||E|)$ asymptotically dominates $\mathcal{O}(k\Delta|V|)$.

6.2. Experimental Results

We evaluate the performance of our heuristics in terms of their minimum VT requirement ($minVT$) and alternate path lengths by using a large number of network topologies with diverse properties. We report the average improvement percentages for $minVT$, the heterogeneity levels of topologies, the correlation between heterogeneity and

$minVT$, and alternate path lengths. Extensive simulation results show that our heuristics significantly reduce $minVT$ while the alternate path lengths are only slightly increased.

6.2.1. Evaluation of Minimum VT Requirement

Table 6.4a lists ave_{minVT} for mMRC-1, mMRC-2, and the MRC along with the corresponding improvement percentages (IP) with respect to the parameter m. The results show that our algorithms require significantly smaller ave_{minVT} values compared to the MRC for all cases. For example, when executed on P_4, ave_{minVT} values of the MRC for $G_{m=2}^{GLP}$, $G_{m=3}^{GLP}$ and $G_{m=4}^{GLP}$ vary as 6.14, 3.60 and 3.02, respectively. For the same subsets, mMRC-1 and mMRC-2 provide smaller ave_{minVT} values such as 4.18, 3.00, 2.59 and 4.39, 3.03, 2.62, respectively, which result in 31.84%, 16.77%, 14.08% and 28.44%, 15.80%, 13.02% as corresponding IP values. We believe that these improvements significantly reduce the operational cost of the MRC, and hence, make it more suitable for its application to the IPFRR. Note that, for the GLP model, mMRC-1 provides the highest IP while mMRC-2 performs notably better in the Waxman model that shows that the performance of our heuristics closely depends on the structures of network topologies.

Table 6.4b presents the variance (σ^2) of the $minVT$ values used to calculate each ave_{minVT} and the corresponding confidence intervals with 95% confidence that are computed using the same topology pools in Table 6.4a. LL and UL correspond to the lower and upper limits of a confidence interval, respectively. LL and UL of a confidence interval (LL, UL) for a certain ave_{minVT} are calculated as $ave_{minVT} - U_r$ and $ave_{minVT} + U_r$, where $U_r = t\sigma/\sqrt{100 - N_{N2C}}$, t is the appropriate percentage point for Student's t-distribution with $100 - N_{N2C} - 1$ degrees of freedom [88], σ is the standard deviation of the $minVT$ distribution and $100 - N_{N2C}$ represents the number of the experimented topologies. Table 6.4b shows that the upper limit of each confidence interval reported for mMRC-2 is always smaller than the lower limit of the corresponding confidence interval for the MRC. This result indicates that mMRC-2 requires a smaller $minVT$ than the MRC with 95% confidence with respect to m for all of the randomly generated topologies in our experiments. For example, for $G_{m=2}^{GLP}$ subset of P_4, the confidence intervals for mMRC-2 and the MRC are $(4.29, 4.49)$ and $(5.94, 6.34)$, respectively. Similarly, mMRC-1 yields confidence intervals whose upper limits are smaller than the corresponding lower limit of the MRC for the topology pools other than $G_{m=4}^{Waxman}$.

Table 6.4. Average minimum VT requirement and its confidence intervals with 95% confidence for topology pools with varying m

Pool	*Algorithm*	*Subsets In P_i*					
		$G^M_{m=2}$		$G^M_{m=3}$		$G^M_{m=4}$	
		ave_{minVT}	IP	ave_{minVT}	IP	ave_{minVT}	IP
P_2	MRC	5.57	–	3.35	–	3.01	–
	mMRC-1	4.06	27.10	3.00	10.44	2.71	9.96
	mMRC-2	4.08	26.75	3.01	10.14	2.69	10.63
P_4	MRC	6.14	–	3.60	–	3.02	–
	mMRC-1	4.18	31.84	3.00	16.77	2.59	14.08
	mMRC-2	4.39	28.44	3.03	15.80	2.62	13.02
P_6	MRC	5.30	–	3.38	–	3.02	–
	mMRC-1	4.00	24.52	3.00	11.24	2.77	8.27
	mMRC-2	4.01	24.33	3.00	11.24	2.68	11.25
P_8	MRC	4.27	–	3.08	–	2.83	–
	mMRC-1	3.91	8.43	3.00	2.59	2.68	5.30
	mMRC-2	3.52	17.56	2.98	3.24	2.21	21.90

(a) ave_{minVT} and IP values

Pool	*Algorithm*	*Subsets In P_i*								
		$G^M_{m=2}$			$G^M_{m=3}$			$G^M_{m=4}$		
		σ^2	LL	UL	σ^2	LL	UL	σ^2	LL	UL
P_2	MRC	0.53	5.42	5.71	0.22	3.25	3.44	0.05	2.96	3.05
	mMRC-1	0.07	4.00	4.11	0.00	3.00	3.00	0.20	2.61	2.80
	mMRC-2	0.17	3.99	4.16	0.01	2.99	3.02	0.21	2.59	2.78
P_4	MRC	0.92	5.94	6.34	0.35	3.47	3.73	0.12	2.94	3.09
	mMRC-1	0.17	4.09	4.27	0.00	3.00	3.00	0.24	2.49	2.69
	mMRC-2	0.24	4.29	4.49	0.03	2.99	3.07	0.23	2.52	2.72
P_6	MRC	0.51	5.15	5.44	0.31	3.26	3.49	0.06	2.97	3.06
	mMRC-1	0.06	3.95	4.04	0.00	3.00	3.00	0.17	2.68	2.85
	mMRC-2	0.17	3.92	4.09	0.00	3.00	3.00	0.21	2.58	2.77
P_8	MRC	0.19	4.18	4.35	0.07	3.02	3.13	0.14	2.75	2.90
	mMRC-1	0.10	3.84	3.97	0.00	3.00	3.00	0.21	2.58	2.77
	mMRC-2	0.25	3.42	3.61	0.01	2.95	3.00	0.16	2.12	2.29

(b) σ^2 and confidence intervals

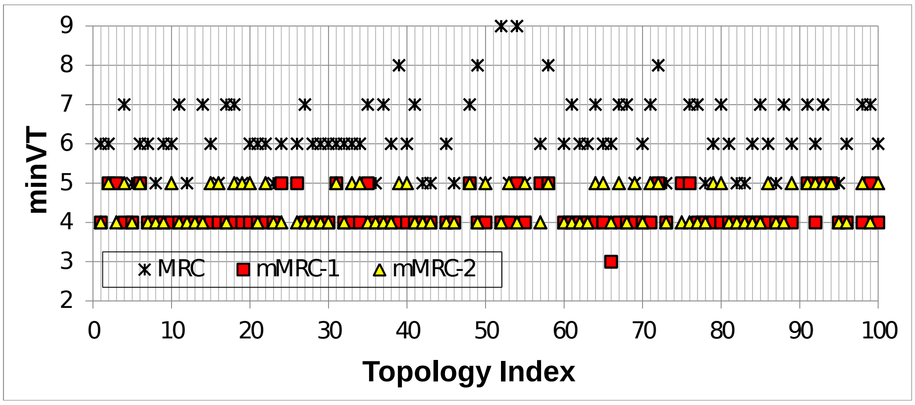
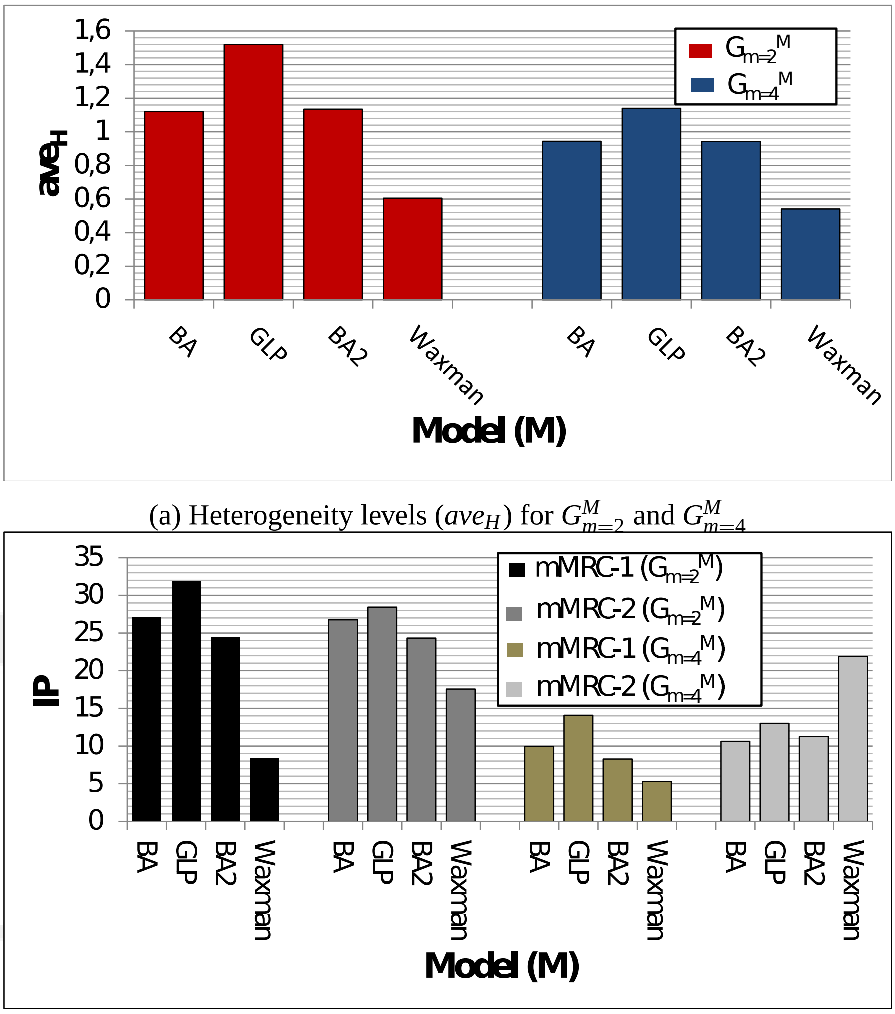


Figure 6.4. *minVT* distribution for $G^{GLP}_{m=2}$ subset of P_4

Fig. 6.4 shows the *minVT* distribution for the MRC and our heuristics using $G^{GLP}_{m=2}$ subset of P_4. Note that no *minVT* value is reported for 9 non-2-connected (N2C) topologies in $G^{GLP}_{m=2}$ indexed by 25, 44, 47, 51, 56, 59, 74, 90, and 97 since neither the MRC nor our heuristics can create VTs for N2C topologies. Out of 91 2-connected topologies, mMRC-1 requires 3, 4, and 5 VTs for 1, 72, and 18 topologies, respectively. On the other hand, mMRC-2 requires 4 and 5 VTs for 55 and 36 topologies, respectively. Fig. 6.4 demonstrates that our heuristics never require a *minVT* larger than 5 while the MRC results in *minVT* values up to 9. For 81 out of 91 2-connected topologies, both mMRC-1 and mMRC-2 require a smaller *minVT* than the MRC. For example, both of our heuristics require a significantly smaller *minVT* of 4 while the MRC requires a *minVT* of 9 for the same topology indexed by 52. For the remaining 10 topologies, either mMRC-1 or mMRC-2 requires a smaller *minVT* than the MRC, or our heuristics require the same *minVT* as the MRC.

The correlation between the heterogeneity levels of network topologies and improvement percentages is shown in Fig. 6.5. The values of ave_H for $G^{M}_{m=2}$ and $G^{M}_{m=4}$ subsets ($M \in V_M = \{$BA, GLP, BA2, Waxman$\}$) are presented in Fig. 6.5a while the values of *IP* for the same subsets are shown in Fig. 6.5b. $G^{GLP}_{m=2}$ and $G^{GLP}_{m=4}$ subsets of P_4 have the highest heterogeneity among the subsets with $m = 2$ and $m = 4$, respectively. On the other hand, $G^{Waxman}_{m=2}$ and $G^{Waxman}_{m=4}$ of P_8 have the lowest heterogeneity levels. Fig. 6.5b demonstrates that, among all $G^{M}_{m=2}$ and $G^{M}_{m=4}$ subsets, mMRC-1 provides its maximum and minimum improvements on the subsets of GLP and Waxman models, respectively (i.e., the most and the least heterogeneous subsets). These results indicate that mMRC-1 achieves higher improvement percentages as the heterogeneity level of a network increases. On the



(a) Heterogeneity levels (ave_H) for $G^M_{m=2}$ and $G^M_{m=4}$

(b) Improvement percentages (IP) for P_2, P_4, P_6 and P_8

Figure 6.5. The correlation between ave_H and IP values

other hand, similar to mMRC-1, among all $G^M_{m=2}$ subsets, mMRC-2 has its highest and lowest IP values on the subsets generated by the GLP and Waxman models, respectively. In the case of $G^M_{m=4}$ subsets, mMRC-2 provides its maximum improvement on $P8$ (i.e., the least heterogeneous subset) while the IP for mMRC-1 is the lowest on $P8$. Note that this topology subset is the least heterogeneous one in average. A possible explanation for this behaviour is that, for less heterogeneous network topologies, the likelihood of network partitioning of VTs by the successive node isolations will be smaller if one prefers to firstly isolate a node that keeps the heterogeneity levels of VTs as low as possible.

Table 6.5a presents the ave_{minVT} for mMRC-1, mMRC-2, and the MRC along with the corresponding IP values using the topology pools where n is varied from 100 to 300. In these experiments, GLP and Waxman models (i.e., P_3 and P_7) yield the highest and

Table 6.5. Average minimum VT requirement and its confidence intervals with 95% confidence for topology pools with varying n

Pool	*Algorithm*	*Subsets In P_i*					
		$G^M_{n=100}$		$G^M_{n=200}$		$G^M_{n=300}$	
		ave_{minVT}	IP	ave_{minVT}	IP	ave_{minVT}	IP
P_1	MRC	4.61	–	5.18	–	5.57	–
	mMRC-1	3.59	22.12	3.92	24.32	4.06	27.10
	mMRC-2	3.59	22.12	3.97	23.35	4.08	26.75
P_3	MRC	4.63	–	5.68	–	6.14	–
	mMRC-1	3.65	21.03	3.98	29.88	4.18	31.84
	mMRC-2	3.74	19.12	4.08	28.12	4.39	28.44
P_5	MRC	4.53	–	5.01	–	5.39	–
	mMRC-1	3.58	20.97	3.82	23.75	3.98	26.15
	mMRC-2	3.54	21.85	3.87	22.75	3.99	25.97
P_7	MRC	4.01	–	4.31	–	4.27	–
	mMRC-1	3.51	12.46	3.63	15.77	3.91	8.43
	mMRC-2	3.19	20.44	3.39	21.34	3.52	17.56

(a) ave_{minVT} and IP values

Pool	*Algorithm*	*Subsets In P_i*								
		$G^M_{n=100}$			$G^M_{n=200}$			$G^M_{n=300}$		
		σ^2	LL	UL	σ^2	LL	UL	σ^2	LL	UL
P_1	MRC	0.42	4.48	4.73	0.57	5.02	5.33	0.53	5.42	5.71
	mMRC-1	0.24	3.49	3.68	0.19	3.83	4.00	0.07	4.00	4.11
	mMRC-2	0.30	3.48	3.69	0.27	3.86	4.07	0.17	3.99	4.16
P_3	MRC	0.59	4.46	4.80	0.80	5.50	5.87	0.92	5.94	6.34
	mMRC-1	0.22	3.55	3.76	0.14	3.90	4.06	0.17	4.09	4.27
	mMRC-2	0.24	3.63	3.85	0.30	3.97	4.20	0.24	4.29	4.49
P_5	MRC	0.43	4.39	4.66	0.41	4.88	5.13	0.54	5.24	5.53
	mMRC-1	0.26	3.47	3.68	0.22	3.72	3.91	0.10	3.91	4.04
	mMRC-2	0.29	3.43	3.64	0.23	3.77	3.96	0.15	3.91	4.06
P_7	MRC	0.21	3.91	4.10	0.23	4.21	4.40	0.19	4.18	4.35
	mMRC-1	0.25	3.41	3.60	0.23	3.53	3.72	0.10	3.84	3.97
	mMRC-2	0.15	3.11	3.26	0.24	3.29	3.48	0.25	3.42	3.61

(b) σ^2 and confidence intervals

lowest heterogeneity levels, respectively, similar to the case in Fig. 6.5a. mMRC-1 performs significantly better than mMRC-2 on the most heterogenous topology pool P_3 while the performance of mMRC-2 is much higher on the least heterogeneous topology pool P_7. For example, the ave_{minVT} values of the MRC for P_3 vary as 4.63, 5.68, and 6.14 while the ave_{minVT} values of mMRC-1 for the same pool change as 3.65, 3.98, and 4.18 that result in the corresponding IP values of 21.03%, 29.88%, and 31.84%. We show in [89] that a network becomes more heterogeneous when n increases. Since our heuristics perform better when the heterogeneity increases, we believe that reducing the $minVT$ by these ratios is an important scalability result especially for networks with a large number of nodes. On the other hand, the ave_{minVT} values of the MRC on P_7 vary as 4.01, 4.31, and 4.27 while the ave_{minVT} values of mMRC-2 are 3.19, 3.39, and 3.52, which result in the corresponding IP values of 20.44%, 21.34%, and 17.56%. Note that, on $G^{Waxman}_{n=300}$, mMRC-2 provides an IP value of 17.56%, which is more than two times higher than mMRC-1's IP value of 8.43%. These results show that the performance of our heuristics closely depends on the heterogeneity level of network topologies so that mMRC-1 provides higher improvement percentages as the heterogeneity level of networks increases while mMRC-2 performs better than mMRC-1 in the least heterogeneous networks.

Table 6.5b presents the variance (σ^2) of each $minVT$ distribution and confidence intervals for each ave_{minVT} with 95% confidence using the same topology pools in Table 6.5a. The upper limit of each confidence interval reported for our heuristics is always smaller than the lower limit of the corresponding confidence interval for the MRC, which indicates that our heuristics requires a smaller $minVT$ than the MRC with 95% confidence with respect to n. For example, for $G^{GLP}_{n=200}$ of P_3, the confidence intervals for mMRC-1, mMRC-2, and the MRC are $(3.90, 4.06)$, $(3.97, 4.20)$, and $(5.50, 5.87)$, respectively.

Table 6.6 shows the $minVT$ of mMRC-1, mMRC-2 and the MRC for real networks, which are from sndlib.zib.de (Nobel), [90] (Labnet03), topology-zoo.org (Rediris, AGIS, InternetMCI), and simula.no (COST239, SprintUS). Similar to [62], a minimum number of nodes, namely 1, 9 and 1, are removed from Rediris, AGIS, and InternetMCI, respectively, to make them 2-connected since only 2-connected networks can provide full coverage against single failures. Our heuristics provide a smaller $minVT$ in 5 networks while they perform as efficient as the MRC for the remaining 2 networks. For example, both mMRC-1 and

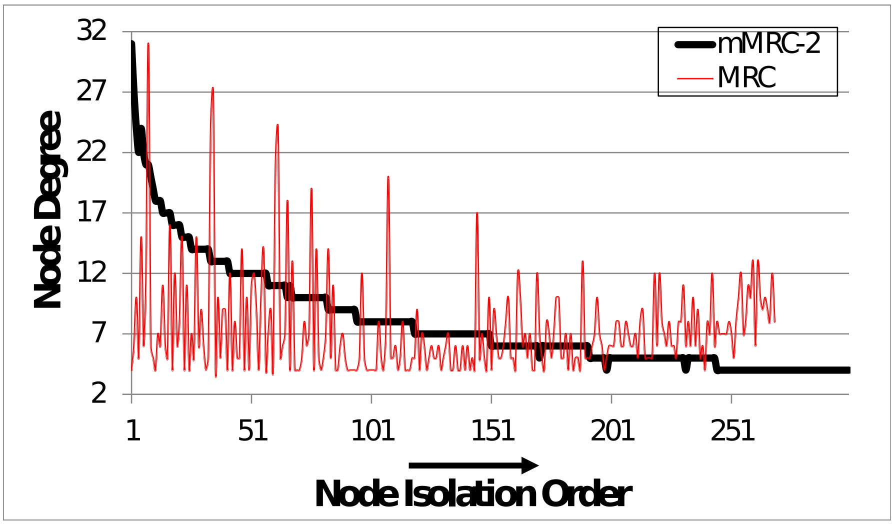
Table 6.6. The comparison of minimum VT requirement using real networks

Network	$\lvert V\rvert$	$\lvert E\rvert$	H	$minVT$ (mMRC-1)	$minVT$ (mMRC-2)	$minVT$ (MRC)
Nobel	28	41	0.288	4	4	5
Labnet03	20	53	0.422	2	3	3
Rediris	18	30	0.608	3	3	4
AGIS	16	21	0.353	5	5	6
InternetMCI	18	32	0.452	4	3	4
COST239	11	26	0.130	2	2	2
SprintUS	32	64	0.556	4	4	4

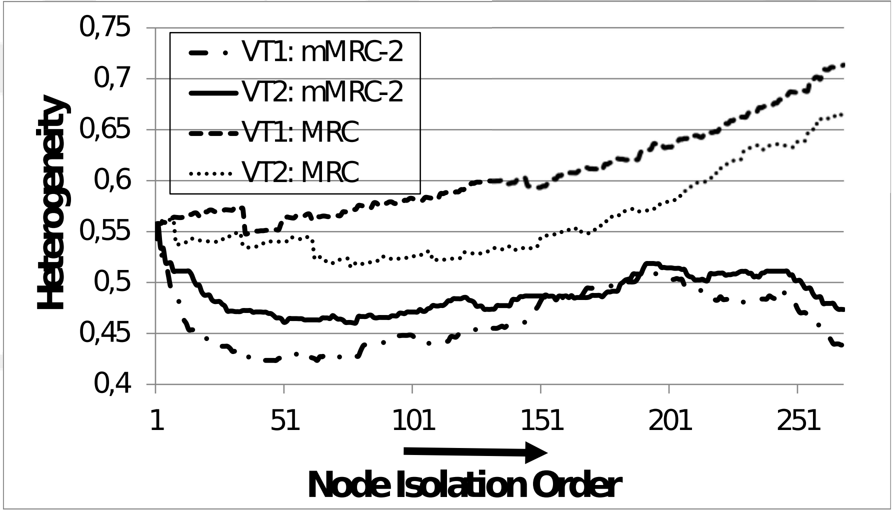
mMRC-2 provide a $minVT$ of 3 in Rediris while $minVT$ for the MRC is 4, which corresponds to an IP of 25%. The correlation between the H and $minVT$ values for real networks indicates that using the H metric to reduce $minVT$ works for 4 network topologies. However, while the H value for SprintUS is relatively high, there is no improvement for $minVT$ in case of mMRC-2. This result indicates that there may be other metrics than H to better measure the robustness of the VTs against partitioning.

6.2.2. Effect of Isolation Order On Heterogeneity

We analyzed the operation of mMRC-2 on an example 300-node topology in $G_{m=4}^{Waxman}$ subset of P_8 to demonstrate the impact of isolation order on the heterogeneity levels of VTs. Fig. 6.6a shows the degree distributions of the isolated nodes for mMRC-2 and the MRC when they are both configured to construct 2 VTs. Since the MRC gives up constructing the configured number of VTs after the isolation of the 269th node, we cannot provide the degree distribution of the MRC for the remaining isolations. As shown in Fig. 6.6a, the degree distribution for mMRC-2 exhibits a decreasing trend while the distribution for the MRC fluctuates since mMRC-2 pre-processes the nodes and links to arrange them in descending order according to the node degrees, but the MRC's selection of nodes and links is arbitrary. Fig. 6.6b shows the heterogeneity distributions of the VTs being constructed by mMRC-2 and the MRC up to the 270th node. The MRC cannot isolate this node in any VT because its isolation results in partitioning both VTs. The heterogeneity levels of the VTs are initially the same for mMRC-2 and the MRC, namely 0.557. Successive isolations of nodes cause the heterogeneity distributions of both VTs to exhibit an increasing trend in the MRC



(a) Degree distributions of the isolated nodes



(b) Heterogeneity distributions of VTs

Figure 6.6. Distributions of isolated node degrees and heterogeneity of VTs

up to the isolation of the 270th node. On the other hand, the heterogeneity distributions of mMRC-2 for both VTs never exceed 0.518. The possible reason for the success of mMRC-2 in building 2 VTs is that the probability of partitioning the VTs is smaller since mMRC-2 keeps the heterogeneity levels of the VTs within certain limits. Although the heterogeneity level of a topology may not always accurately indicate the probability of partitioning, our experiment results demonstrate that heterogeneity might be a good indicator for the partitioning probability of VTs, especially in Waxman topologies.

6.2.3. Evaluation of Alternate Path Lengths

Table 6.7 presents the values for *RAPL* of mMRC-1, mMRC-2, the MRC, and the MRT (low-point version), that is defined in Section 5.3.1, using the topology pool P_3. *RAPL* values for the MRC$_{\text{Des}}$, the variant of the MRC that arranges the nodes in the Q_n in descending order according to their degrees, but selects VTs in a round-robin fashion as in the MRC, are also reported. Table 6.7 shows that both mMRC-1 and mMRC-2 significantly reduce the minimum VT requirement of the MRC at the expense of only slightly increasing the alternate path lengths. The *RAPL* values reported for mMRC-1 are higher than the *RAPL* values for mMRC-2 for all subsets since mMRC-1 achieves a higher *IP* on P_3 (Table **??**). Note that, in general, the *RAPL* values increase as $minVT$ decreases. The *RAPL* values for the MRC$_{\text{Des}}$ also decrease with respect to *n*, and are very close to the *RAPL* values of the MRC. The *RAPL* values for the MRC are the lowest since all the variants of the MRC mostly achieve a smaller $minVT$ than the MRC. The *RAPL* values for the MRT are the highest since it uses only two VTs, which are constructed using a procedure different from the MRC. These results show that all variants of the MRC generate scalable alternate paths close to the optimum SPF routing whereas the MRT's alternate path lengths do not scale well with respect to the network size.

Table 6.7. RAPL values computed using P_3

Subset	mMRC-1	mMRC-2	MRC$_{\text{Des}}$	MRC	MRT
$G^{GLP}_{n=100}$	112.29	109.87	109.02	109.02	225.30
$G^{GLP}_{n=200}$	110.80	109.01	107.65	107.05	249.18
$G^{GLP}_{n=300}$	110.06	108.09	106.83	106.40	284.21

7. MULTI TOPOLOGY ROUTING BASED IPFRR FOR SOFTWARE DEFINED NETWORKS

In the previous chapters, we studied the inner workings of MT-IPFRR approaches, and focused on the MRC since it generates scalable alternate paths with respect to network size and density. We investigated the topological dynamics behind the performance of the MRC. Inspired by our topological analysis results, we introduced our topology-aware heuristics which reduce the number of VTs used by the MRC, and, hence, decrease its operational complexity.

In this chapter, we propose a new MT-IPFRR technique for SDN to fast recover from the network failures in the data plane by defining all the tasks which should be performed by the controller during the failure recovery process. The central architecture of SDN makes it an ideal platform to deploy MT-IPFRR since the central computation of the VTs by the controller guarantees the consistency among the alternate routing tables. Our technique relies on the restoration approach where the alternate routing tables are pro-actively computed by the controller in advance of the failure while the resources for the alternate paths are allocated upon the detection of the failure [24]. Our technique utilizes the MRC algorithm to construct VTs which provide protection against single link/node failures in the data plane, and can be used to build a self-recovering SDN from the network failures. Our experimental results show that our approach considerably reduces the recovery time from the failures compared to the reactive recovery in SDN. Note that the topological properties of the VTs used for the failure recovery do not have an impact on the recovery time. This fact is the reason why we chose the MRC algorithm to construct the VTs for our experiments rather than using its variants proposed in this thesis since the version of the MRC in use does not make any difference in terms of recovery time. The rest of this chapter is organized as follows. Section 7.1 describes the fundamental concepts of SDN. Section 7.2 introduces the workflow for the application of MT-IPFRR to SDN. Section 7.3 provides our experimental results.

7.1. Software Defined Networks

SDN physically separates the control plane of a communication network from the forwarding plane as opposed to the traditional networks where both the control and forwarding functions are built into each forwarding device. Therefore, the deployment of new protocols and applications in traditional networks is difficult since it requires to become familiar with the interfaces provided by each forwarding device. The centralized management capability offered by SDN makes the underlying network easily programmable.

A typical SDN architecture has three planes, namely, *data*, *control*, and *application* [91]. The data plane consists of hardware devices such as switches that are physically connected to form the network topology. Each switch incorporates a flow table which contains the rules regarding how the packets of incoming flows should be treated. Each flow rule has the format <*Header, Statistics, Actions*> where *Header* defines the flow to match the flow rule, *Statistics* keep the number of the packets that match the flow rule, and *Actions* define how the matching packet should be processed by the forwarding device. When a packet arrives at a switch, the packet is compared against the entries in the flow table. If one of the flow entries matches with the header of the incoming packet, the action of the corresponding entry is performed. If there is no match, the packet is forwarded to the controller located at the control plane which will decide a routing path for the incoming packet, and add new entries into the flow tables of the switches to activate this path. The control plane contains the *controller* software which is responsible for the management of the switches in the data plane. The hardware devices in the data plane communicate with the control plane using a communication interface such as the OpenFlow protocol which was standardized by the Open Networking Foundation [92]. The controller periodically communicates with the devices in the data plane to maintain the network state information such as the network topology and traffic statistics. It also communicates with the application plane to provide services to applications which play a key role in this architecture by providing network management and security functions.

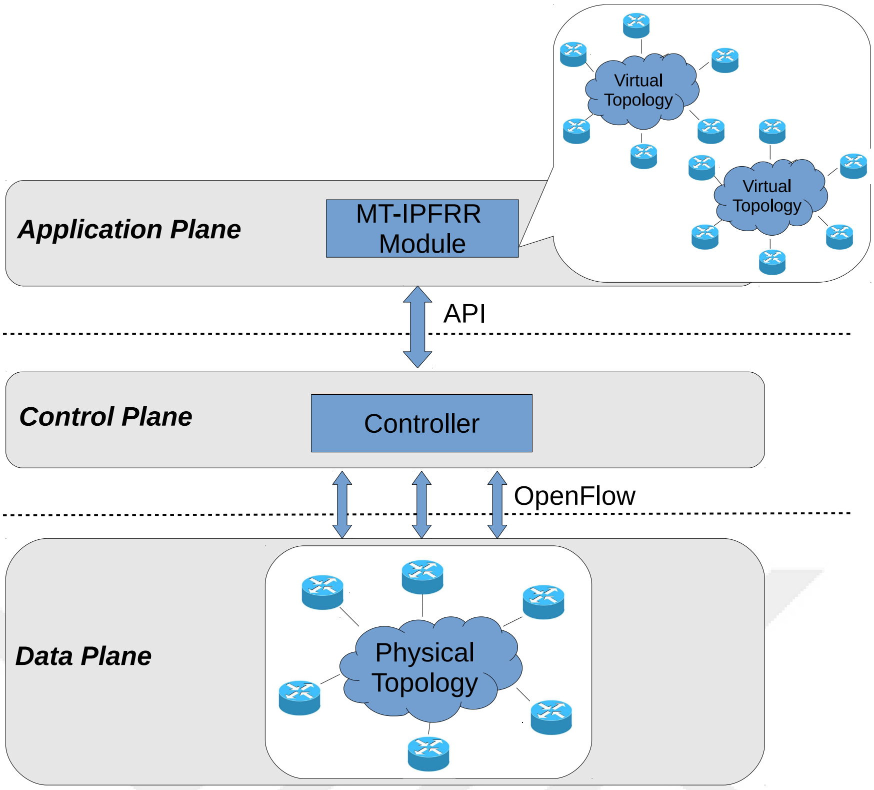


Figure 7.1. SDN architecture incorporating MT-IPFRR

7.2. Architecture for MT-IPFRR in SDN

Fig. 7.1 shows the SDN architecture which incorporates an MT-IPFRR module which is responsible for the failure recovery in addition to the determination of the primary routes for the flows. The module constructs a pre-configured number of VTs based on the physical topology in the data plane. It interacts with the controller via the API (Application Programming Interface) functions provided by the controller to perform the tasks including the discovery of the underlying physical topology, keeping track of the failures in the data plane and the modification of the flow tables of the switches to activate a primary path in no-failure scenario or an alternate path upon detection of a failure. The controller in turn interacts with the switches in the data plane via OpenFlow protocol [92]. Note that the MT-IPFRR module is not responsible for the detection of the failures, but relies on the failure notifications received from the controller which are generated based on the underlying failure detection mechanisms such as Loss of Signal (LOS) or Bidirectional Forwarding Detection (BFD) [24].

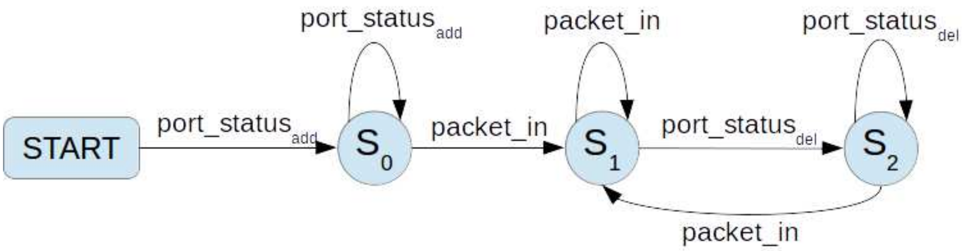


Figure 7.2. State machine to perform MT-IPFRR in SDN

Table 7.1. States versus actions

State	*Index*	*Action*
S_0	1	The new link is stored in a topology database
	2	Primary routing tables are computed
	3	VTs and alternate routing tables are computed

(a) Actions performed during network initialization

State	*Index*	*Action*
S_1	1	The primary route for the new flow is computed, and then activated in the data plane by updating the flow tables
	2	The flow information along with its primary route is stored in a data structure to track the active flows in the data plane

(b) Actions performed upon the detection of a new flow

State	*Index*	*Action*
S_2	1	The VT which isolates the failed component is determined
	2	The active flows which are disrupted by the failure are determined
	3	The alternate routes for the disrupted flows are computed and then activated in the data plane
	4	The primary routing tables are re-computed
	5	The VTs and the alternate routing tables are re-computed based on the physical topology without the failed component

(c) Actions performed upon the detection of a failure

Fig. 7.2 depicts the state machine for the MT-IPFRR module in Fig. 7.1 while the actions performed in each state are listed in Table 7.1 where the order of execution is indicated by the indices. Note that the actions specified in Table 7.1 perform the failure recovery in SDN using the restoration method. Our module performs its tasks by processing the OpenFlow messages from the switches including *packet_in* and *port_status*. *START* is the initial state which makes a transition to S_0 if a $port\_status_{add}$ message, which corresponds to a *port_status* message for a link addition, is received. During the network initialization, upon the receipt of each $port\_status_{add}$ message, some preliminary tasks are carried out in S_0 as listed in Table 7.1a. Both the VTs along with the routing tables are repeatedly computed each time a new link is added to the topology, not only for once at the end of the topology discovery process. This is due to the fact that there is no way to decide the end of the discovery process until the first *packet_in* message arrives. Note that MT-IPFRR can successfully construct the VTs only if the underlying topology is 2-connected [8]. When a new traffic flow is detected via a *packet_in* message, a transition from S_0 to S_1 is made. As specified in Table 7.1b, the route to transmit the new flow is decided in S_1 based on the primary routing table, which was already computed in S_0. If the new flow is an ARP (Address Resolution Protocol) request, no primary route can be computed for the flow since the network address of the destination is not known at this stage. Therefore, in such a case, the ARP request is flooded to the entire network. Upon the receipt of a $port\_status_{del}$ message which corresponds to a *port_status* message for a link deletion, a transition from S_1 to S_2 is made. As listed in Table 7.1c, the alternate paths for the disrupted flows are determined, and the primary and alternate routing tables are re-computed each time a failure is detected in S_2. A transition back to S_1 is made when a new flow is detected.

MT-IPFRR using the restoration method is explained using the topology in Fig. 7.3 whose corresponding VTs are illustrated in Fig. 7.4. The primary path from S_2 to S_7 is selected to be the shortest one, namely S_2-S_4-S_7, by the controller which inserts the corresponding flow entries into S_2, S_4 and S_7 to activate the primary path. After the link (S_4, S_7) fails, the controller calculates the recovery path S_2-S_3-S_5-S_8-S_7 using VT_3 shown in Fig. 7.4 since the link (S_4, S_7) is isolated in VT_3. The controller activates this recovery path

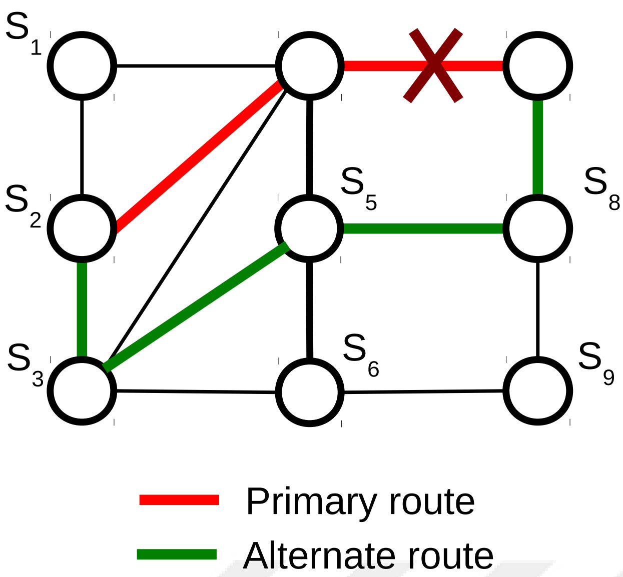


Figure 7.3. An example scenario for the restoration method

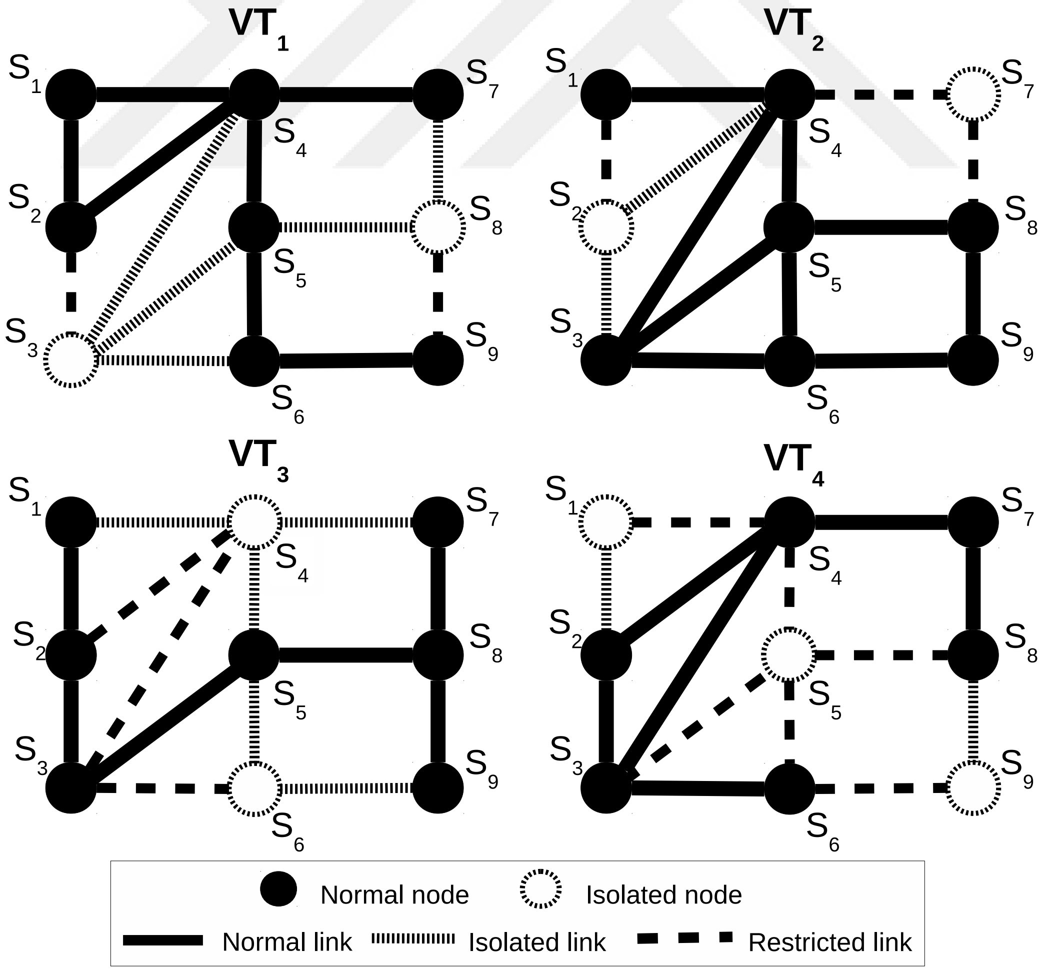


Figure 7.4. Example virtual topologies

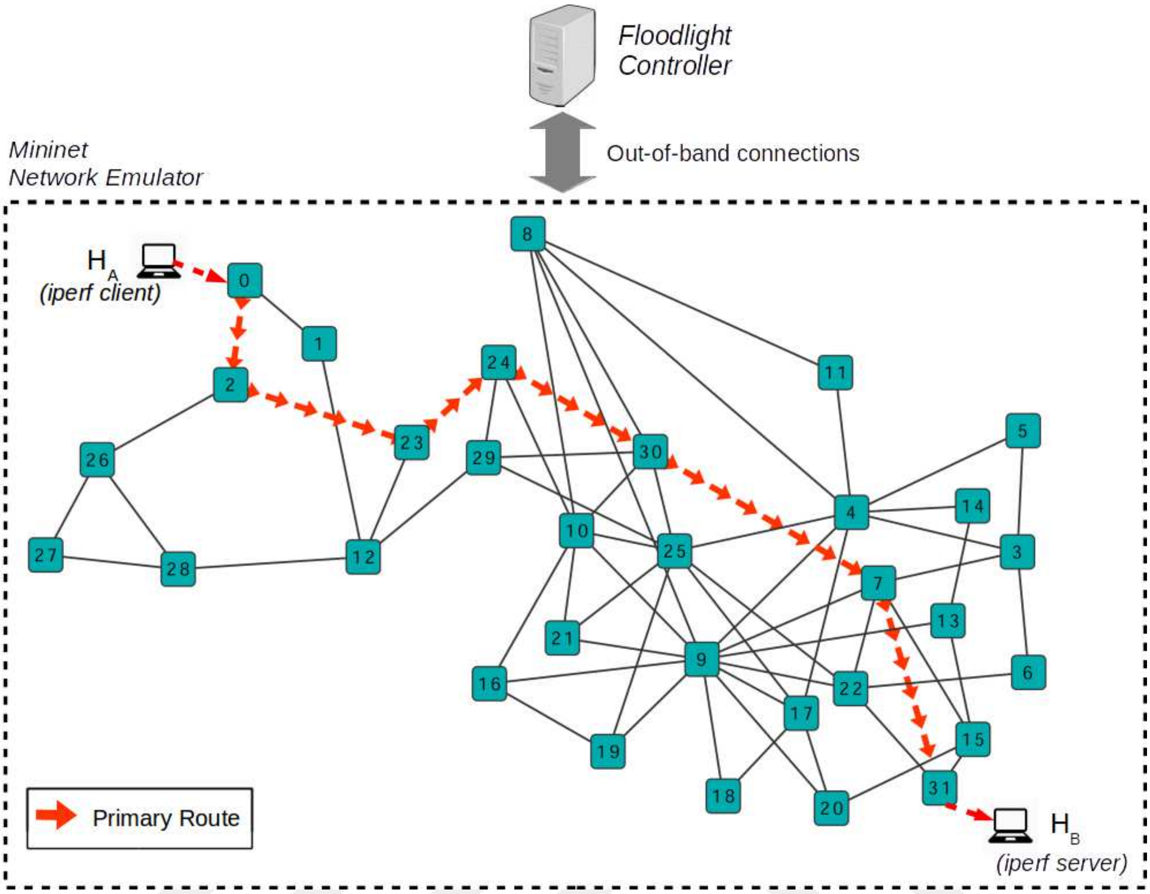


Figure 7.5. Experimental setup

by deleting the flow entries of the failed primary path in S_2, S_4 and S_7, and adding the new entries into the flow tables of the switches on the recovery path.

7.3. Experimental Results

We implemented the MT-IPFRR module presented in Fig. 7.1 which has all the functionalities in Table 7.1. The module uses the MRC algorithm to construct the VTs. In this section, we analyze the performance of MT-IPFRR for failure recovery in SDN using the SprintUS topology with 32 switches and 64 links as the data plane. Both the MRC software and the SprintUS topology were obtained from the Simula Research Laboratory web site (*http://simula.no*).

Fig. 7.5 demonstrates our experimental setup where the data plane containing the SprintUS topology was created within the Mininet network emulator [93] on a computer with Intel i5 processor running at 2.4 GHz and 4 GB main memory. Each switch in the data plane is connected to the Floodlight controller (*http://www.projectfloodlight.org*), running

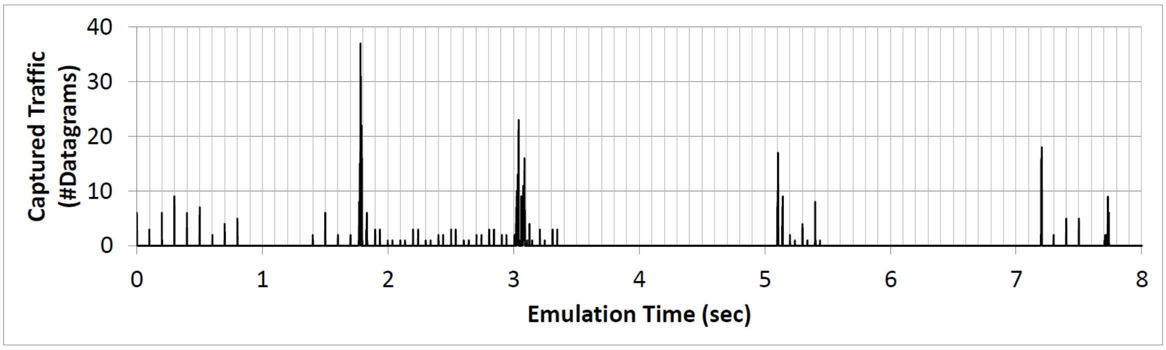


Figure 7.6. Traffic captured at the controller during the failure recovery process

external to Mininet, through an out-of-band control link. Each link in the data plane is assumed to have unit weight. A UDP (User Datagram Protocol) traffic flow between H_A and H_B computers, which are connected to the switches 0 and 31, respectively, was created through the *iperf* tool. The flow has a bandwidth of 10 Mbit/s and is composed of UDP datagrams each of which is 1470 bytes long. The primary route of the flow is indicated by the red arrows in Fig. 7.5. We evaluate the failure recovery time (Ω) in the case that one of the links on the primary route fails. Ω is defined to be the time that it takes to activate an alternate path in the data plane which does not use the failed component. The recovery time for a failure is computed using the formula $\Omega = \rho - \gamma$ where ρ is the capture time of the first datagram received by H_B following the emergence of the failure while γ is the capture time of the last datagram received by H_B prior to the failure. *Wireshark* tool is run on H_B to determine the capture times of the received datagrams. In our experiments, we used OpenFlow 1.1 and OpenvSwitch 2.0.2, and relied on the default failure detection mechanism employed by the OpenvSwitch 2.0.2. Note that the failures in the data plane in our experiments are triggered through the commands that Mininet provides.

Fig. 7.6 shows the amount of traffic exchanged between the Floodlight controller and the switches during the transmission of the UDP flow in Fig. 7.5, which was captured on the controller via Wireshark. The only messages sent and received by the controller up to the 1.76th second of the emulation are *echo_request* and *echo_reply* which verify the liveness of the switches [94]. From 1.76th to 3.01st second, *packet_out* and *packet_in* messages containing the LLDP (Link Layer Discovery Protocol) frames [95], which are used to discover the links in the data plane, are exchanged by the controller with a peak for *packet_out* messages at 1.78th second. Since a UDP flow is initiated between H_A and

H_B at 3.01st second, a *packet_in* message from the switch 0, to which H_A is connected, is received by the controller at this point. This message contains an ARP query for the network address of H_B, which is in turn flooded to the entire network via a *packet_out* message. Up to 3.03rd second, the controller keeps receiving *packet_in* messages with the ARP queries for H_B which are then flooded to the entire network. At 3.06th second, the *packet_in* message containing the first UDP datagram of the flow is received by the controller. The primary route is activated in the data plane via *flow_mod* messages up to 3.09th second. Note that the MT-IPFRR module successfully completes the computation of all the VTs along with the primary and alternate routing tables prior to the activation of the primary route. No messages other than *echo_request* and *echo_reply* are exchanged between the controller and switches up to 7.705th second when the link $(0,2)$ is failed in Mininet environment. This failure triggers a *port_status* message to be sent to the controller. Up to 7.74th second, the entries regarding the disrupted flow are deleted from the the flow tables of the switches on the primary route, and the alternate route determined based on the VTs is activated on the data plane by installing the new flow entries to the switches.

Table 7.2. Performance comparison of MT-IPFRR with the reactive recovery

Failed Link	*IPFRR*			*Reactive Recovery*		
	Ω (ms)	*#LD*	*APL*	Ω (ms)	*#LD*	*APL*
$(0,2)$	44.877	36	7	101.733	84	6
$(2,23)$	32.967	25	8	63.745	50	6
$(23,24)$	37.193	30	7	84.350	69	6
$(24,30)$	41.055	27	8	52.810	43	6
$(30,7)$	51.311	41	8	63.402	51	6
$(7,31)$	35.947	26	7	64.423	53	6

Table 7.2 shows the comparison results, namely Ω, the number of lost datagrams (#LD) and the alternate path length (*APL*), for MT-IPFRR and the reactive recovery in the case that one of the links on the primary route depicted in Fig. 7.5 fails. Ω for MT-IPFRR is reported for the restoration case. Note that MT-IPFRR determines the alternate path based on the alternate routing table which is computed prior to the actual failure using the Shortest Path First (SPF) tree of the VT which isolates the failed link. On the other hand, for the reactive recovery, Ω is reported for the case in which both the alternate routing tables and the alternate

paths are reactively computed upon the detection of the failure based on the SPF tree of the physical topology which excludes the failed link. Table 7.2 shows that MT-IPFRR achieves a smaller recovery time for each failure scenario compared to the reactive recovery. For example, for the failure of the link $(0,2)$, Ω varies as 44.877 ms and 101.733 ms for IPFRR and the reactive recovery, respectively, while #LD varies as 36 and 84. Table 7.2 also shows that the *APL* values are slightly higher in the case of MT-IPFRR, which is due to the fact that the path diversity of the virtual topologies used by IPFRR is smaller compared to the reactive case. For example, the *APL* is 7 in the case of IPFRR while it is 6 for the other case. These results show that MT-IPFRR significantly reduces the failure recovery time compared to the reactive recovery.

8. DISCUSSION

In this thesis, we used a large number of topologies with diverse structural properties to analyze the performance of the MRC under varying topological conditions. We evaluated topological metrics other than the network size (n) and density (m) since it has been discovered in Section 3 that n and m can not always be correlated with the minimum VT requirement. We chose BRITE as our topology generation tool since it is widely used by the networking community to randomly generate topologies for the experiments. Our performance evaluation in this thesis is comprehensive since we used all the topology generation models supported by BRITE, namely, Waxman, BA, BA2, and GLP. We used the Cytoscape's Network Analyzer in combination with BRITE to analyze the topological dependency of the MRC, and showed that the topologies in our experiments are structurally diverse such that they exhibited trends in H, C, and CC. In addition to our experiments using the synthetic topologies, we also performed experiments on real networks which provided similar results to the ones for the synthetic topologies of BRITE. The experimental results showed that our topology-aware algorithms reduce the minimum VT requirement of the MRC for the real network deployments.

Cytoscape's Network Analyzer can also be used to evaluate the topology characterizing metrics other than H, C, and CC which may also be correlated with the minimum VT requirement. Such an evaluation may inspire the development of other algorithms which further reduces the minimum VT requirement by relying on other topological metrics. In the case that Waxman, BA, BA2, and GLP models do not yield structurally diverse topologies in terms of the topological metric to be evaluated, other topology generation models in the literature may need to be used to generate topologies with the requested topological diversity.

We believe that, taking the large structural diversity of the topologies used in our experiments into consideration, choosing a topological metric with a low computational overhead which measures the robustness of the VT backbones against partitioning sufficiently well under all circumstances is a complex issue. Therefore, designing a topology-aware algorithm to reduce the minimum VT requirement under all topological conditions is a challenging task. Our experimental results show that an MRC variation may achieve the highest reduction in the minimum VT requirement in a certain topology while the same variation may perform worse than the other variations in another topology. If a topology-aware algorithm considers multiple topology characterizing metrics in combination, it may perform more efficiently under varying topological conditions. However, the computation of these metrics should be fast enough to satisfy the operational needs of a real network deployment.

When the number of VTs in MT-IPFRR gets higher, the processing time and state requirements of routers increase [19–22]. The amount of reductions in the number of VTs reported in Section 5 and Section 6 leads to the reductions in *(i)* the time required to compute alternate routing tables, and *(ii)* the amount of required FIB storage. In MT-IPFRR, alternate routing tables are computed by constructing the VTs and their corresponding SPF trees when the network is initialized, and this computation is repeated each time a network failure is detected. Therefore, a reduction in the number of VTs leads to a reduction in *(i)* that speeds up the preparation of the alternate next hops for the anticipated failures in a topology. This is an important result which helps maintaining QoS in a network where the time interval between successive failures is short. For example, reducing the minimum VT requirement of the MRC from 5 to 4 for a topology such as Geant in Table 5.4 leads to 20% improvement in the SPF computation time for the VTs. The number of alternate next-hops that needs to be stored in FIB decreases as well when the number of VTs for MT-IPFRR is reduced. The numerical experiments show that our heuristics reduces *(ii)* by around 30% compared to the conventional MRC. This is a significant result to speed up the activation of the alternate next hops, which decreases the reaction time to the failures. Moreover, this reduction decreases the cost of having a large FIB.

The trend in our numerical results indicates that the improvement percentages of our heuristics may get higher as the number of nodes (n) increases. Fig. 6.5 in Section 6.2.1 shows that both mMRC-1 and mMRC-2 provide higher IP values as the heterogeneity level of the networks increases. Since we observed in [89] that, excluding the Waxman topologies, the heterogeneity level of networks in our experiments increases as n gets higher, our heuristics provide a scalable solution. The selection between mMRC-1 and mMRC-2 in a real network deployment can be made according to the heterogeneity level of the network which is the main reason for the performance difference between these heuristics.

We suggest that our automated analysis tool in Section 4 can be used offline to provide guidance to the network operator for the selection between mMRC-1 and mMRC-2 in a real network deployment. The selection process can also be fully automated by running both heuristics and selecting the one that provides a fewer number of VTs. However, it takes longer time to run both heuristics to determine their VT requirements for a given topology, which delays the preparation of alternate routing tables to recover from the network failures, and hence, increases the reaction time to the failures. Although the mechanism to make a selection between mMRC-1 and mMRC-2 is not within the scope of this thesis, we plan to investigate the tradeoff between these offline and automated selection approaches in our future work.

Even though mMRC-1 and mMRC-2 are likely to lead to the construction of a fewer number of VTs for a topology compared to the conventional MRC, it can not guarantee that the fewest number of VTs will be constructed since a selection to guarantee the fewest number of VTs requires a generic threshold for the heterogeneity level to be defined. This is not possible since the heterogeneity is not a metric which under all circumstances represent the robustness of the VTs against partitioning sufficiently well.

9. CONCLUSIONS

In this thesis, we first describe the operational principles of the MRC and the MRT algorithms using example networks. Both mechanisms pre-compute VTs for MT-IPFRR which can be effectively used to provide full protection against single network failures. The MRC requires the number of VTs to be calculated as an input to its algorithm whereas the MRT always yields only two VTs. We performed experiments using randomly generated topologies to evaluate the alternate path lengths. The alternate path lengths achieved by the MRC relative to the OSPF's converged paths ranged from 111.6% to 133.64% while the alternate path lengths of the MRT varied between 224.35% and 525.39%. These performance results showed that the MRC with a configurable number of VTs provides scalable alternate paths whereas the MRT's alternate path lengths significantly increase with respect to network size and density. We also investigated the alternate path lengths and the minimum number of VTs required by the MRC for a complete failure protection using realistic ISP-level topologies. We observed that an increase in network density usually leads to a reduction in the alternate path lengths. We also discovered that the topological properties other than network size and density which may affect the minimum VT requirement should be investigated since the trends in the number of nodes in a topology as well as the density can not always be correlated with the trends in the minimum VT requirement.

We implemented an automated topology generation and analysis framework to analyze the impact of varying topological conditions on the performance of the MRC. We generated 11,500 topologies with diverse clustering, heterogeneity, and centralization levels by varying the parameters of various topology generation models including BA, GLP, BA2, and Waxman. The numerical results showed that, when the number of nodes in a topology gets higher, the generated topologies become more heterogenous whereas the increase in network density leads to less heterogenous networks. Through extensive experiments, we demonstrated that, if a network becomes more heterogenous (i.e., certain nodes have much higher number of neighbors than the other nodes), the MRC requires a higher number of VTs to provide full alternate path coverage. We believe that our framework will significantly contribute not only to the evaluation of IPFRR mechanisms but also to the network planning and management studies requiring the usage of various topologies with diverse characteristics. Inspired by the correlation between the minimum VT requirement and the heterogeneity, we proposed a new heuristic algorithm enhancing the MRC which takes the node degree information into account. Numerical experiments showed that our heuristic significantly reduces the complexity in terms of processing time and state requirements of routers by decreasing the number of VTs which should be constructed by the MRC. The analysis results confirmed the effectiveness of our systematic approach in performing the topological analysis of networking algorithms.

We proposed two topology-aware algorithms which take the heterogeneity and the link density information into consideration to decrease the operational complexity of the MRC by reducing its minimum VT requirement. We performed extensive experiments on 3200 topologies with diverse structural properties showing that our algorithms significantly reduce the minimum VT requirement. The results showed that the alternate path lengths are kept within acceptable limits. mMRC-1 reduces the VT requirement of the MRC up to 31.84%, and achieves higher improvement percentages than mMRC-2 as the heterogeneity level of the networks increases. On the other hand, mMRC-2 performs better than mMRC-1 when the heterogeneity decreases, and provides an improvement of up to 28.44%. Our heuristics provided a smaller number of VTs in 5 out of 7 real networks while they performed as efficient as the MRC for the remaining networks. Numerical results also showed that there may be other metrics for certain topologies to better measure the robustness of the VTs against partitioning. The alternate path lengths of our heuristics are very close to the optimum SPF routing while they are slightly higher than the alternate path lengths of the MRC.

We also defined the workflow to perform the MT-IPFRR in SDN. We proposed a new MT-IPFRR technique, which was implemented as a restoration method, for the failure recovery in SDN. Our approach provided alternate paths in case of single link/node failures in the data plane. Our experimental results showed that our approach considerably reduces the time to recover from network failures in the SprintUS topology by up to 55% compared to the reactive recovery in SDN. These results indicated that the MT-IPFRR is a promising approach to construct fault tolerant SDN. In our future work, we plan to implement our approach as a protection method, and compare its performance results with the restoration based implementation.

10. RECOMMENDATIONS

The future work may concentrate on investigating the impact of the topological metrics other than *H*, *C*, and *CC* on the performance of the MRC. This requires the generation of a topology pool with the requested topological diversity for the experiments which may neccesitate the usage of the topology generation models other than Waxman, BA, BA2, and GLP. A topology-aware algorithm considering only a single topological metric to reduce the minimum VT requirement may yield satisfactory results in a certain topology while it may perform poorly in another topology. Therefore, a topology-aware algorithm combining multiple topological metrics can be designed to improve the overall performance. However, the computation of these metrics should be fast enough to satisfy the operational needs of a real network deployment. Finally, MT-IPFRR can be implemented as a protection method to further reduce the failure recovery time in SDN.

11. REFERENCES

1. Francois, P., Filsfils, C., Evans, J. and Bonaventure, O., Achieving Sub-Second IGP Convergence in Large IP Networks, ACM SIGCOMM Computer Communication Review, 35, 2 (2005) 35-44.

2. Tagarelli, A. and Interdonato, R., Who's Out There?: Identifying and Ranking Lurkers in Social Networks, IEEE/ACM Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2013, Canada, 215–222.

3. Ahn, Y.Y., Han, S., Kwak, H., Moon, S. and Jeong, H., Analysis of Topological Characteristics of Huge Online Social Networking Services, 16th International Conference on World Wide Web, 2007, Canada, 835-844.

4. Doncheva, N.T., Assenov, Y., Domingues, F.S. and Albrecht, M., Topological Analysis and Interactive Visualization of Biological Networks and Protein Structures, Nature Protocols, 7, 4 (2012) 670-685.

5. Su, G., Morris, J.H., Demchak, B. and Bader, G.D., Biological Network Exploration with Cytoscape 3, Current Protocols in Bioinformatics, 47 (2014) 1-24.

6. RFC 5714, IP Fast Reroute Framework, Internet Engineering Task Force (IETF), USA, 2010.

7. RFC 5286, Basic Specification for IP Fast Reroute: Loop-Free Alternates, Internet Engineering Task Force (IETF), USA, 2008.

8. Kvalbein, A., Hansen, A.F., Cicic, T., Gjessing, S. and Lysne, O., Multiple Routing Configurations for Fast IP Network Recovery, IEEE/ACM Transactions on Networking, 17, 2 (2009) 473-486.

9. Enyedi, G. and Retvari, G., Finding Multiple Maximally Redundant Trees in Linear Time, Periodica Polytechnica, 54 (2011) 29-40.

10. Internet Draft, Algorithms for Computing Maximally Redundant Trees for IP/LDP Fast-Reroute, Internet Engineering Task Force (IETF), USA, 2014.

11. Internet Draft, An Architecture for IP/LDP Fast-Reroute Using Maximally Redundant Trees, Internet Engineering Task Force (IETF), USA, 2014.

12. RFC 6981, A Framework for IP and MPLS Fast Reroute Using Not-Via Addresses, Internet Engineering Task Force (IETF), USA, 2013.

13. Internet Draft, Loop-Free IP Fast Reroute Using Local and Remote LFAPs, Internet Engineering Task Force (IETF), USA, 2008.

14. Cevher, S., Chen, T., Hokelek, I., Kang, J., Kaul, V., Lin, Y. J., Pang, M., Rodoper, M., Samtani, S., Shah, C., Bowcock, J., Rucker, G.B., Simbol, J.L. and Staikos, A., An Integrated Soft Handoff Approach to IP Fast Reroute in Wireless Mobile Networks,

2nd International Conference on Communication Systems and Networks (COMSNETS), 2010, Bangalore, India, 1-10.

15. Internet Draft, IP Fast Re-Route with Fast Notification, Internet Engineering Task Force (IETF), USA, 2012.

16. Internet Draft, Remote LFA FRR, Internet Engineering Task Force (IETF), USA, 2013.

17. RFC 4915, Multi-Topology (MT) Routing in OSPF, Internet Engineering Task Force (IETF), USA, 2007.

18. RFC 5120, M-ISIS: Multi Topology (MT) Routing in Intermediate System to Intermediate Systems (IS-ISs), Internet Engineering Task Force (IETF), USA, 2008.

19. Apostolopoulos, G., Using Multiple Topologies for IP-only Protection Against Network Failures: A Routing Performance Perspective, Institute of Computer Science (ICS) of the Foundation for Research and Technology - Hellas (FORTH), Tech. Rep. TR377, 2006.

20. Menth, M. and Martin, R., Network Resilience through Multi-Topology Routing, 5th International Workshop on Design of Reliable Communication Networks, 2005, Naples, Italy, 271-277.

21. Cicic, T., Hansen, A.F. , Kvalbein, A., Hartman, M., Martin, R., Menth, M., Gjessing, S. and Lysne, O., Relaxed Multiple Routing Configurations: IP Fast Reroute for Single and Correlated Failures, IEEE Transactions on Network and Service Management, 6, 1 (2009) 1-14.

22. Kamamura, S., Miyamura, T., Pelsser, C., Inoue, I. and Shiomoto, K., Minimum Backup Configuration-Creation Method for IP Fast Reroute, Global Telecommunications Conference (GLOBECOM), 2009, Hawaii, USA, 1-6.

23. Huck, A., ‘Independent Trees in Graphs, Graphs and Combinatorics, 10, 1 (1994) 29–45.

24. Sharma, S., Staessens, D., Colle, D., Pickavet, M. and Demeester, P., OpenFlow: Meeting Carrier-Grade Recovery Requirements, Computer Communications (Elsevier), 36, 6 (2013) 656-665.

25. Cevher, S., Ulutas, M. and Hokelek, I., Trade-off Analysis of Multi Topology Routing Based Ip Fast Reroute Mechanisms, IEEE INFOCOM 16th Global Internet Symposium, 2013, Turin, Italy, 157-162.

26. Cevher, S., Ulutas M. and Hokelek, I., Performance Evaluation of Multiple Routing Configurations, 21st Signal Processing and Communications Applications Conference (SIU), 2013, Cyprus, 1-4.

27. Cevher, S., Ulutas, M. and Hokelek, I., Enhancing Multiple Routing Configurations Through Systematic Analysis of Topological Characteristics, International Journal of Network Management, 26, 3 (2015) 176-198.

28. Medina, A., Lakhina, A., Matta, I. and Byers, J., BRITE: An Approach to Universal Topology Generation, 9th International Symposium in Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), 2001.

29. Smoot, M.E, Ono, K., Ruscheinski, J., Wang, P.L. and Ideker, T., Cytoscape 2.8: New Features for Data Integration and Network Visualization, Bioinformatics, 27, 3 (2011) 431-432.

30. Dong, J. and Horvath, S., Understanding Network Concepts in Modules, BMC Syst Biol, 1, 24 (2007).

31. Cevher, S., Ulutas, M. and Hokelek, I., Topology-Aware Multiple Routing Configurations for Fault Tolerant Networking, Journal of Network and Systems Management, doi: 10.1007/s10922-015-9358-4 (2015).

32. Cevher, S., Ulutas, M., Altun, S. and Hokelek, I., Multiple Routing Configurations for Fast Re-Route in Software Defined Networks, 24th Signal Processing and Communications Applications Conference (SIU), 2016, Zonguldak.

33. Cevher, S., Ulutas, M., Altun, S. and Hokelek, I., Multi Topology Routing Based IP Fast Re-Route for Software Defined Networks, 21st IEEE Symposium on Computers and Communications (ISCC), 2016, Messina. (Accepted)

34. Radoslavov, P., Tangmunarunkit, H., Yu, H., Govindan, R., Shenker, S. and Estrin, D., On Characterizing Network Topologies and Analyzing Their Impact on Protocol Design, Technical Report, USC-CS-TR-00-731, 2000.

35. Lee, D.S and Kalb, J.L., Network Topology Analysis, Technical Report, SAND2008-0069, 2008.

36. Mahadevan, P., Krioukov, D., Vahdat, A. and Fall, K., A Basis for Systematic Analysis of Network Topologies, Technical Report, Cooperative Association for Internet Data Analysis (CAIDA), 2006.

37. Labovitz, C., Ahuja, A., Wattenhofer, R. and Venkatachary, S., The Impact of Internet Policy and Topology on Delayed Routing Convergence, IEEE INFOCOM, 2001, Anchorage, USA, 537-546.

38. Li, Z. and Mohapaira, P., The Impact of Topology on Overlay Routing Service, IEEE INFOCOM, 2004, Hong Kong, .

39. Medem, A., Magnien, C. and Tarissan, F., Impact of Power-Law Topology on IP-Level Routing Dynamics: Simulation Results, NetSciCom, 2012, Orlando, USA, 220-225.

40. Verma, K.P. and Hu, Z., Impact of Topology on the Performance of Communication Networks, National Conference on Communications (NCC), 2011, Bangalore, 1-5.

41. Calcada, T., Cortez. P. and Ricardo, M., Using Data Mining to Study the Impact of Topology Characteristics on the Performance of Wireless Mesh Networks, IEEE Wireless Communications and Networking Conference (WCNC), 2012, Shanghai, 1725-1730.

42. Kamiyama, N., Kawahara, R., Mori, T., Harada, S. and Hasegawa, H., Impact of Topology on Parallel Video Streaming, IEEE Network Operations and Management Symposium (NOMS), 2010, Osaka, 607-614.

43. Xi, F. and Liu. Z., Small World Topology-Aware Geographic Routing in Wireless Sensor Networks, International Conference on Communications and Mobile Computing (CMC), 2009, Yunnan, 116-120.

44. Yigitel, M.A., Incel, O.D. and Ersoy, C., QoS vs. Energy: A Traffic-Aware Topology Management Scheme for Green Heterogeneous Networks, Computer Networks, 78 (2015) 130-139.

45. Yakine, F. and Idrissi, A., Energy-Aware Topology Control And Qos Routing In Ad Hoc Networks, Procedia Computer Science, 56 (2015) 309-316.

46. Deniz, F., Bagci, H., Korpeoglu, I. and Yazici, A., An Adaptive, Energy-Aware and Distributed Fault-Tolerant Topology-Control Algorithm for Heterogeneous Wireless Sensor Networks, Ad Hoc Networks, 44 (2016) 104-117.

47. Thulasiraman, P. and White, K.A., Topology Control of Tactical Wireless Sensor Networks Using Energy Efficient Zone Routing, Digital Communications and Networks, 2 (2016) 1-14.

48. Hegde, R., Kumar, S. and Gurumurthy, K.S., The Impact of Network Topologies on the Performance of the In-Vehicle Network, International Journal of Computer Theory and Engineering, 5, 3 (2013) 405-409.

49. Hauge, M., Brose, M.A., Sander, J. and Andersson, J., Multi-Topology Routing for QoS Support in the CoNSIS Convoy MANET , Military Communications and Information Systems Conference (MCC), 2012, Gdansk, 1-8.

50. Bae, S. and Henderson, T.R., Traffic Engineering with OSPF Multi-Topology Routing, IEEE Military Communications Conference (MILCOM), 2007, Orlando, 1-7.

51. Xiantai, G., Hongchao, Y., Feng, Y., Gang, L. and Qian, W., Modeling and Simulation of Small Satellite Constellation Networking Using Multi-Topology Routing, International Conference on Computer Application and System Modeling (ICCASM), 2010, Taiyuan Shanxi, 143-147.

52. Xie, L., Zhang, X. and Wang, S., An Energy Efficient Algorithm Based on Multi-Topology Routing in IP Networks, 23rd International Conference on Computer Communication and Networks (ICCCN), 2014, Shanghai, 1-8.

53. Neumann, A., Lopez, E., Cerda-Alabern, L. and Navarro, L., Securely-Entrusted Multi-Topology Routing for Community Networks, 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS), 2016, Cortina d’Ampezzo, 1-8.

54. Imahama, D., Fukushima, Y. and Yokohira, T., A Reroute Method Using Multiple Routing Configurations for IP Network Recovery, 19th Asia-Pacific Conference on Communications (APCC), 2013, Denpasar, 433-438.

55. Harada, Y., Hui, W., Fukushima, Y. and Yokohira, T., A Reroute Method to Recover Fast From Network Failure, International Conference on Information and Communication Technology Convergence (ICTC), 2014, Busan, 903-908.

56. Takahashi, R., Tembo, S., Yukimatsu, K., Kamamura, S., Miyamura, T. and Shiomoto, K., Dispersing Hotspot Traffic in Backup Topology for IP Fast Reroute, IEEE International Conference on Communications (ICC), 2011, Kyoto, 1-5.

57. Lee, S.S.W., Li, K. and Chen, A., Energy Efficient Multi-Topology Routing Configurations for Fast Failure Reroute in IP Networks, Global Telecommunications Conference (GLOBECOM), 2012, Anaheim, USA, 2785-2790.

58. Gardner, M.T., May, R., Beard, C. and Medhi, D., A Geographic Multi-Topology Routing Approach and Its Benefits During Large-Scale Geographically Correlated Failures, Computer Networks, 82 (2015) 34-49.

59. Imahama, D., Fukushima, Y. and Yokohira, T., Decreasing of the Number of Backup Tables in Multiple Routing Table Method for Fast IP Network Recovery, 28th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC), 2013, Yeosu.

60. Cicic, T., An Upper Bound on the State Requirements of Link-Fault Tolerant Multi-Topology Routing, IEEE International Conference on Communications, 2006, Istanbul, 1026-1031.

61. Gjessing, S., Implementation of Two Resilience Mechanisms Using Multi Topology Routing and Stub Routers, Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT-ICIW’06), 2006, Guadelope.

62. Menth, M. and Braun, W., Performance Comparison of Not-Via Addresses and Maximally Redundant Trees (MRTs), IEEE International Symposium on Integrated Network Management, 2013, Ghent, 218-225.

63. Kuang, K., Wang, S. and Wang, X., Discussion on the Combination of Loop-Free Alternates and Maximally Redundant Trees for IP Networks Fast Reroute, IEEE International Conference on Communications (ICC), 2014, Sydney, 1131-1136.

64. Bejerano, Y. and Koppol, P.V., Optimal Construction of Redundant Multicast Trees in Directed Graphs, IEEE INFOCOM, 2009, Rio de Janeiro, 2696 - 2700.

65. Bejerano, Y., Jana, S. and Koppol, P.V., Efficient Construction of Directed Redundant Steiner Trees, IEEE 37th Conference on Local Computer Networks (LCN), 2012, Clearwater, 119-127.

66. Braun, W. and Menth, M., Scalable Resilience for Software-Defined Networking Using Loop-Free Alternates with Loop Detection, IEEE Network Softwarization, 2015, London, 1-6.

67. Adrichem, N.L.M., Asten, B.J. and Kuipers, F.A., Fast Recovery in Software Defined Networks, European Workshop on Software Defined Networks, 2014, Budapest, 61-66.

68. Kempf, J., Bellagamba, E., Kern, A., Jocha, D., Takacs, A. and Skoldstrom, P., Scalable Fault Management for OpenFlow, IEEE International Conference on Communications (ICC), 2012, Ottawa, 6606-6610.

69. Sgambelluri, A., Giorgetti, A., Cugini, F., Paolucci, F. and Castoldi, P., OpenFlow-Based Segment Protection in Ethernet Networks, Journal of Optical Communications and Networking, 5, 9 (2013) 1066-1075.

70. Lee, S., Li K.Y., Chan K.Y., Lai, G.H. and Chung, Y.C., Path Layout Planning and Software Based Fast Failure Detection in Survivable OpenFlow Networks, 10th International Conference on the Design of Reliable Communication Networks (DRCN), 2014, Ghent, 1-8.

71. Kuzniar, M., Peresini, P., Vasic, N., Canini, M. and Kostic., D., Automatic Failure Recovery for Software-Defined Networks, HotSDN, 2013, Hong Kong, 159-160.

72. Chu, C.Y., Xi, K., Luo, M. and Chao, H.J., Congestion-Aware Single Link Failure Recovery in Hybrid SDN Networks, IEEE INFOCOM, 2015, Kowloon, 1086-1094.

73. Ros, F.J. and Ruiz, P.M., On Reliable Controller Placements in Software-Defined Networks, Computer Communications, 77 (2016) 41-51.

74. Beheshti, N. and Zhang, Y., Fast Failover for Control Traffic in Software-Defined Networks, IEEE GLOBECOM, 2012, Anaheim, 2665-2670.

75. Hu, Y., Wendong, W., Xiangyang, G., Liu, C.H., Que, X. and Cheng, S., Control Traffic Protection in Software-Defined Networks, IEEE Global Communications Conference, 2014, Austin, 1878-1883.

76. Li, C., Li, X., Li, K., Zhang, H., Shi, Y. and Chen, S., Toward Software Defined AS-Level Fast Rerouting, The Journal of China Universities of Posts and Telecommunications, 21, 6 (2014) 100-108.

77. Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., Zolla, J., Holzle, U., Stuart, S. and Vahdat, A., B4: Experience with a Globally-Deployed Software Defined WAN, SIGCOMM, 2013, Hong Kong, 3-14.

78. Heckmann, O., Piringer, M., Schmitt, J. and Steinmetz, R., Generating Realistic ISP-Level Network Topologies, IEEE Communications Letter, 7, 7 (2003) 335-336.

79. Medina, A., Lakhina, A., Matta, I. and Byers, J., BRITE: Universal Topology Generation From a User’s Perspective, User Manual, BUCS-TR-2001-003, Boston University, 2001.

80. Freeman, L., Centrality in Social Networks: Conceptual Clarification, Social Networks, 1, 3 (1978) 215-239.

81. Watts, D.J. and Strogatz, S.H., Collective Dynamics of Small World Networks, Nature, 393 (1998) 440-442.

82. Barabasi, A.L. and Albert, R., Emergence of Scaling in Random Networks, Science, 286, 5439 (1999) 509–512.

83. Albert, R. and Barabasi, A., Topology of Evolving Networks: Local Events and Universality, Physical Review Letters, 85, 24 (2000).

84. Bu, T. and Towsley, D., On Distinguishing Between Internet Power Law Topology Generators, IEEE INFOCOM, 2002, New York, 638-647.

85. Wang, Y., Xing, L. and Wang, H., Reliability of Scale-Free Complex Networks, Reliability and Maintainability Symposium (RAMS), 2013, Orlando, 1-6.

86. Mahanti, A., Carlsson, N., Mahanti, A., Arlitt, M. and Williamson, C., A Tale of the Tails: Power-Laws in Internet Measurements, IEEE Network, 27, 1 (2013) 59-64.

87. Garey, M. and Johnson, D., Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, San Francisco, 1979.

88. Zukerman M., Introduction to Queueing Theory and Stochastic Teletraffic Models, City University of Hong Kong, Hong Kong, 2014.

89. Cevher, S., Ulutas, M. and Hokelek, I., An Automated Topological Analysis of Multiple Routing Configurations, 36th IEEE Sarnoff Symphosium, 2015, Newark, 99-104.

90. Menth, M., Efficient Admission Control and Routing in Resilient Communication Networks, Ph.D. Thesis, University of Würzburg, Faculty of Computer Science, 2004.

91. TR-504, SDN Architecture Overview 1.1, Open Networking Foundation (ONF), USA, 2014.

92. TR-016, OpenFlow Management and Configuration Protocol 1.2, Open Networking Foundation (ONF), USA, 2014.

93. Heller, B., Reproducible Network Research with High-Fidelity Emulation, Ph.D. Thesis, Stanford University, 2013.

94. TS-020, OpenFlow Switch Specification 1.5.0, Open Networking Foundation (ONF), USA, 2014.

95. 802.1AB, Station and Media Access Control Connectivity Discovery, IEEE, USA, 2009.

CURRICULUM VITAE

Selçuk Cevher was born in Trabzon, Turkey in 1978. He graduated from the Fatih High School in 1996. He received his B.Sc. and M.Sc. from the Department of Computer Engineering at Karadeniz Technical University, Trabzon, Turkey in 2000 and 2004, respectively. From 2005 to 2008, he worked as a research assistant at Department of Computer Science at City University of New York, NY, USA. From 2007 to 2008, he worked as a graduate co-op in Telcordia Technologies, Inc., New Jersey. From 2009 to 2010, he worked as a software engineer in CTech, Kocaeli, Turkey. From 2010 up to present, he has been working as a lecturer in the Department of Computer Engineering at Karadeniz Technical University. His research interests include IP networks and network reliability. Among his publications are:

S. Cevher, M. Ulutas, and I. Hokelek, "Enhancing Multiple Routing Configurations Through Systematic Analysis of Topological Characteristics", International Journal of Network Management, Wiley, 2016, doi: 10.1002/nem.1926

S. Cevher, M. Ulutas, and I. Hokelek, "Topology-Aware Multiple Routing Configurations for Fault Tolerant Networking", Journal of Network and Systems Management, Springer, 2015, doi: 10.1007/s10922-015-9358-4

S. Cevher, M. Ulutas, and I. Hokelek, "Multi Topology Routing Based IP Fast Re-Route for Software Defined Networks", ISCC, Messina, Italy, 2016 (Accepted)

S. Cevher, M. Ulutas, and I. Hokelek, "Automated Topological Analysis of Multiple Routing Configurations", IEEE SARNOFF, NJ, USA, 2015

S. Cevher, M. Ulutas, and I. Hokelek, "Trade-off Analysis of Multi Topology Routing Based IP Fast Re-route Mechanisms", INFOCOM Global Internet Workshop, Turin, Italy, 2013

S. Cevher, T. Chen, I. Hökelek, J. Kang, V. Kaul, et.al, “An Integrated Soft Handoff Approach to IP Fast Reroute in Wireless Mobile Networks”, COMSNETS, Bangalore, India, 2010

S. Cevher, M. U. Uyar, M. Fecko, J. Sucec, and S. Samtani, "Interest-Aware and Bandwidth Efficient Multicast Group Planning", MILCOM 2008, San Diego, CA, USA, Nov 2008