

Computer Communications and Networks

Jacek Rak

Resilient Routing in Communication Networks

A Systems Perspective

Second Edition

 Springer


Computer Communications and Networks

Series Editors

Jacek Rak, Department of Computer Communications, Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Gdansk, Poland

A. J. Sammes, Cyber Security Centre, Faculty of Technology, De Montfort University, Leicester, UK

Editorial Board Members

Burak Kantarci , School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada

Eiji Oki, Graduate School of Informatics, Kyoto University, Kyoto, Japan

Adrian Popescu, Department of Computer Science and Engineering, Blekinge Institute of Technology, Karlskrona, Sweden

Gangxiang Shen, School of Electronic and Information Engineering, Soochow University, Suzhou, China

The **Computer Communications and Networks** series is a range of textbooks, monographs and handbooks. It sets out to provide students, researchers, and non-specialists alike with a sure grounding in current knowledge, together with comprehensible access to the latest developments in computer communications and networking.

Emphasis is placed on clear and explanatory styles that support a tutorial approach, so that even the most complex of topics is presented in a lucid and intelligible manner.

Jacek Rak

Resilient Routing in Communication Networks

A Systems Perspective

Second Edition



Springer

Jacek Rak
Department of Computer Communications
Faculty of Electronics Telecommunications,
and Informatics
Gdansk University of Technology
Gdansk, Poland

Chair of Computer Networks and Computer
Communications
Faculty of Computer Science
and Mathematics
University of Passau
Passau, Germany

ISSN 1617-7975 ISSN 2197-8433 (electronic)
Computer Communications and Networks
ISBN 978-3-031-64656-0 ISBN 978-3-031-64657-7 (eBook)
<https://doi.org/10.1007/978-3-031-64657-7>

1st edition: © Springer International Publishing Switzerland 2015

2nd edition: © The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

To my parents

Preface

Since the introduction of the Internet in the 1970s, the concept of global communications has notably changed our daily activities. Communication networks and networked systems, in general, providing access to various services at any time and location, have become the key elements of the critical infrastructure on which our everyday lives depend. Therefore, they are expected to offer uninterrupted service in the presence of various challenges.

As the effective capacity of networks is continuously increasing to accommodate the more-or-less exponentially growing demand volumes, the cost of failures of network elements is rising as well. The same observation refers to the increase in frequency, intensity, and scale of failures, particularly those triggered by natural disasters (such as fires, floods, hurricanes, volcano eruptions, or earthquakes), technology-related massive failures, and malicious human activities. All these factors undoubtedly call for the deployment of adequate mechanisms of networked systems resilience to ensure that these systems can maintain an acceptable level of service in failure scenarios.

Network resilience is undoubtedly a complex issue. For any network architecture, a proper understanding of network challenges, including natural threats and malicious human activities, is necessary to introduce the appropriate preventive mechanisms related to end-to-end communications resilience—the topic addressed in this book. A particular focus of this book is on mechanisms of resilient routing that are able to maintain the ability of a network to provide communication services in the presence of disruptions. This book addresses resilient routing from the perspective of a networked system, i.e., a system composed of interconnected elements (including servers and computing units) providing storage, computation, and communication services.

Compared to the first edition of this book, this edition extends the first two chapters of the former edition into seven chapters, now highlighting a comprehensive set of aspects of networked systems resilience and almost doubling the size of the first edition. The last three chapters are meant to serve as representative case studies illustrating the utilization of the related resilience mechanisms in differentiated scenarios.

The book is divided into three parts. Part I “Introduction to Networked Systems Resilience” consists of three chapters and serves as the introduction. Chapter 1 elaborates on resilience and its importance for networks. A particular focus of Chap. 1 is on a detailed classification of failure scenarios, the definition of network resilience and its desired properties, the explanation of consecutive phases of recovery of services in failure scenarios, a discussion of the impact of human and organizational issues on the resilience of networked systems followed by the analysis of costs, benefits, and challenges to the deployment of resilience mechanisms.

Chapter 2 discusses the taxonomy of challenges and faults, errors, and failures, and describes the disciplines of resilience referring to network design approaches to provide service continuity (such as survivability, fault tolerance, traffic tolerance, and disruption tolerance mechanisms), as well as measurable characteristics, including the attributes of network dependability (such as reliability and availability), security, or performability. The later part of Chap. 2 explains the techniques for evaluating and improving the total availability and reliability of serial, parallel, and mixed architectures of systems.

The objective of Chap. 3 is to discuss the most important metrics useful in evaluating resilience. It starts with the analysis of properties of metrics dedicated to single elements of the system. Next, it explains the properties of the essential structural metrics and discusses the reasons for the diverse characteristics of system elements, the related irregular characteristics of the system topology, and the resulting potential challenges. After that, it elaborates on the major functional metrics, i.e., the ones useful for evaluating system performance at the network level and the packet level, as well as the subjective metrics referring to the satisfaction of users. The final part of Chap. 3 discusses the practical applications of the analyzed metrics in everyday use (e.g., in the configuration of routing protocols).

Part II of this book, entitled “Schemes of Resilient Routing,” consists of four chapters discussing the strategies and algorithms of resilient routing. In particular, Chap. 4 focuses on mechanisms of multi-hop resilient communications in connection-oriented systems. In particular, it first outlines the architectural properties of ring networks together with the related resilience mechanisms. It next highlights the major schemes of resilient routing in mesh networks. In this context, it presents a taxonomy of resilient routing strategies according to six main criteria: backup path setup method, failure model, scope of recovery procedure, usage of recovery resources, and application of recovery schemes to multidomain and multilayer architectures of networked systems. Finally, it explains the components of the total recovery time for common architectures of communication networks.

The objective of Chap. 5 is to discuss the properties of resilience mechanisms in packet-switched networks necessary to reduce the long convergence time experienced in default configurations in these systems. In this context, it first analyzes the properties of the spanning tree protocol (STP) characteristic of Layer-2 Ethernet networks and investigates its major variants aimed at ensuring fast recovery of affected spanning trees. Next, it explains the properties of the selected Layer-3 fast recovery mechanisms in IP and IP-MPLS networks.

The main focus of Chap. 6 is on modeling the optimization problems related to resilient routing involving preplanned protection mechanisms in connection-oriented communication systems. In this context, several optimization models are defined to obtain the shortest sets of disjoint working and protection paths. The next part of Chap. 6 explains the properties of optimization models related to sharing of backup path resources and using other forms of protection structures such as p -cycles. The final part of Chap. 6 explains the properties of the most relevant mathematical methods that can be used to solve the optimization problems addressed in this chapter.

Chapter 7 discusses the properties of computationally efficient methods for determining the shortest sets of k end-to-end disjoint communication paths crucial in assuring protection against simultaneous failures of $k-1$ network elements. For this purpose, it starts with explaining the details of the common Dijkstra's algorithm for calculating the shortest path between a particular pair of end nodes, as this algorithm plays an essential role in the operation of other algorithms discussed in this chapter. Next, it explains and illustrates the most representative schemes to determine a shortest set of disjoint paths in single-cost networks, namely, Suurballe's and Bhandari's algorithms. Finally, it discusses the properties of the k -Penalty algorithm designed to determine the set of k end-to-end disjoint paths in networks with different costs assigned to links for calculation of different paths (i.e., the so-called "multi-cost" network case).

Part III ("Case Studies") includes three chapters presenting example deployment scenarios for resilience mechanisms in selected network configurations. In particular, Chap. 8 investigates the resilience of Future Internet communications. It starts with the scheme of resource provisioning for the Future Internet architecture defined in terms of three Integer Linear Programming (ILP) models and equivalent heuristics that allow for fair allocation of network resources using the concept of virtualization. The latter part of Chap. 8 investigates the resilience of content-oriented networking. In this context, it presents the extension of the anycast routing concept in a way that also protects against failures of destination nodes (which is impossible for the common unicast transmission scheme). The chapter also discusses the anycast routing technique to substantially reduce the number of affected flows due to malicious activities targeted at high-degree nodes.

Protection against regional failures is addressed in Chap. 9, which presents the solutions for Wireless Mesh Networks commonly formed by stationary mesh routers interconnected by wireless links. High-frequency communications (e.g., using the 71–86 GHz band) is the reason for the vulnerability of WMNs to weather-based disruptions, particularly to intensive precipitation. Therefore, heavy rainfalls may seriously reduce (or even completely degrade) the available link capacity in a given area. In this context, Chap. 9 introduces a set of measures of WMN resilience to region-based disruptions and proposes a transmission scheme that allows the preparation of the WMN topology to adverse weather conditions in advance by using the dynamic antenna alignment features according to forecasts of heavy rainfalls.

The last communications scenario is related to the resilience of end-to-end routing in VANETs and is presented in Chap. 10. This novel concept of wireless mobile networks organized in an ad hoc manner encounters link availability problems due to vehicles' high mobility. The problem becomes even more complicated if the stability of end-to-end multi-hop paths is concerned. Chapter 10 presents two approaches to resilient end-to-end routing in VANETs that help remarkably increase the lifetime of end-to-end communication paths. The first one is designed to provide differentiated protection paths based on investigated classes of service. The second scheme extends the concept of anypath routing to improve the probability of end-to-end message delivery by utilizing a new metric of link costs to select stable links in message forwarding decisions.

This book can be used at the university level for differentiated courses on communication networks and networked systems. For instance:

- The content of selected Chaps. 1 through 5 can be applied to introductory courses on communication networks and networked systems.
- Chapters 1 through 7 can serve as the content of a one-semester course on network resilience.
- A complete set of Chaps. 1 through 10 was prepared for advanced modeling-oriented courses on network resilience.

Finally, this book can be used as a reference material by researchers and professionals interested in the resilience of communication technologies.

A remarkable part of the content of this book is inspired by discussions during my scientific visits, scholarships, invited lectures, and seminar talks over the last 15 years, e.g., at Ghent University–iMinds, Belgium; Concordia University, Montreal, Canada; Technical University of Munich, Germany; University of Iceland, Reykjavik, Iceland; University of Passau, Germany; National Institute of Information and Communications Technology (NICT) Tokyo, Japan; Osaka University, Japan; Norwegian University of Science and Technology, Trondheim, Norway; University of Coimbra, Portugal; Halmstad University, Sweden; Lund University, Sweden; Lancaster University, United Kingdom.

This book was prepared in part during my research fellowship visit (PICAIS Research-in-Residence Fellowship) in 2024 at the University of Passau, Germany (the Chair of Computer Networks and Computer Communications led by Hermann de Meer). I want to extend special thanks to Teresa Gomes from the University of Coimbra, Portugal; David Hutchison from Lancaster University, United Kingdom; Mariusz Mycek from Warsaw University of Technology, Poland; Michal Pióro from Warsaw University of Technology/Gdansk University of Technology, Poland; Stefan Schmid from Technical University of Berlin, Germany, Hermann de Meer, Alexander Kilian and Armin Stocker from University of Passau, Germany, for their valuable comments beneficial in improving the technical content of the current edition of this book.

Contents

Part I Introduction to Networked Systems Resilience

1	Fundamentals of Resilience of Communication Networks and Networked Systems.....	3
1.1	A Classification of Failures of Network Elements.....	5
1.2	Network Resilience.....	7
1.3	Recovery of Services.....	10
1.4	The Impact of Human and Organizational Issues on the Resilience of Networked Systems.....	13
1.5	Deployment of Resilience Mechanisms: Costs and Benefits.....	14
1.6	Other Challenges to the Deployment of Resilience Mechanisms...	16
1.7	Summary.....	17
	References.....	18
2	Resilience of Networked Systems: A Taxonomy of Challenges, Faults, Disciplines, and Attributes.....	21
2.1	Challenges.....	23
2.2	Faults.....	28
2.3	Errors and Failures.....	32
2.4	Resilience Disciplines.....	33
2.4.1	Survivability and Fault Tolerance.....	34
2.4.2	Disruption Tolerance.....	36
2.4.3	Traffic Tolerance.....	36
2.4.4	Trustworthiness and Its Attributes.....	37
2.5	Evaluation and Improvement of System Total Availability.....	45
2.5.1	Total Availability for Serial Systems.....	45
2.5.2	Total Availability for Parallel Systems.....	47
2.5.3	Total Availability for Mixed Systems.....	48
2.5.4	A General Strategy to Determine the Total Availability of Complex Systems.....	50
2.5.5	Improvement of Networked Systems Availability.....	51
2.6	Evaluation of System Reliability.....	52

- 2.7 Summary..... 54
- References..... 55
- 3 System- and Element-Related Metrics Useful in the Evaluation of Resilience..... 59**
 - 3.1 The Formal Representation of Networked Systems Architecture... 61
 - 3.2 Centrality Metrics for Evaluation of Resilience of Single System Elements..... 64
 - 3.3 Structural Metrics for Evaluation of Resilience of Networked Systems Architectures..... 69
 - 3.4 Reasons for Diverse Characteristics of System Elements..... 76
 - 3.5 Functional Metrics for Networked Systems Resilience Evaluation..... 78
 - 3.5.1 Network-Level Metrics..... 78
 - 3.5.2 Packet-Level Metrics..... 81
 - 3.5.3 Subjective Metrics..... 82
 - 3.6 Selected Examples for Adaptation of Metrics in Networked Systems..... 83
 - 3.7 Summary..... 85
 - References..... 86

Part II Schemes of Resilient Routing

- 4 Strategies and Concepts for Resilient Routing in Circuit-Switched Networked Systems..... 91**
 - 4.1 Resilient Routing in Ring Networks..... 93
 - 4.2 The Need for Resilience Differentiation in Mesh Networks..... 96
 - 4.3 Schemes for Backup Path Resources Reservation in Mesh Networks..... 98
 - 4.3.1 Backup Path Setup Method..... 99
 - 4.3.2 Failure Model..... 100
 - 4.3.3 Scope of Recovery Procedure..... 102
 - 4.3.4 Usage of Recovery Resources..... 103
 - 4.3.5 Protection Cycles..... 106
 - 4.3.6 Domain of Recovery Operation..... 109
 - 4.3.7 Layer of Recovery Operations..... 111
 - 4.4 Analysis of Recovery Time in the Optical Layer..... 113
 - 4.5 Recovery Time in the IP-MPLS Layer..... 117
 - 4.6 Summary..... 119
 - References..... 121
- 5 Resilience Schemes for Fast Recovery in Packet-Switched Communication Systems..... 125**
 - 5.1 Link-Layer Recovery Mechanisms in Packet-Switched Networks..... 126
 - 5.1.1 Spanning Tree Protocol..... 126

5.1.2	Rapid Spanning Tree Protocol.....	128
5.1.3	Multiple Spanning Trees.....	129
5.2	Mechanisms of Fast Recovery in IP Networks.....	131
5.3	IP-MPLS Mechanisms for Fast Recovery.....	134
5.3.1	Proactive Schemes of Resilient Routing in MPLS Networks.....	135
5.3.2	Reactive Approaches to Resilient Routing in MPLS Networks.....	137
5.4	Summary.....	138
	References.....	139
6	Optimization Methods for Resilient Routing in Connection-Oriented Communication Networks.....	143
6.1	Network Flows.....	144
6.2	The Network Model Applied in This Chapter.....	147
6.3	Finding the Set of Working Paths.....	148
6.4	Finding the Set of Pairs of Disjoint Working and Protection Paths.....	152
6.4.1	Nodal Disjointness of Working and Protection Paths.....	153
6.4.2	Link Disjointness of Working and Protection Paths.....	155
6.5	Optimization Model for the “A Priori” Sharing of Backup Path Resources.....	156
6.6	Optimization Model for the “A Posteriori” Sharing of Backup Path Resources.....	159
6.7	Protection Cycles (<i>p</i> -Cycles).....	163
6.8	Mathematical Methods Used to Solve the Resilient Routing Optimization Problems.....	165
6.8.1	Simplex Method.....	167
6.8.2	Branch-and-Bound Method.....	171
6.8.3	Column Generation Method.....	173
6.9	Conclusions.....	174
	References.....	175
7	Efficient Methods to Determine Disjoint Paths for Single Demands.....	177
7.1	Dijkstra’s Algorithm.....	180
7.2	Suurballe’s Algorithm.....	184
7.3	Bhandari’s Algorithm.....	188
7.4	<i>k</i> -Penalty Algorithm to Determine Sets of Disjoint Paths in Multi-cost Networks.....	193
7.5	Summary.....	196
	References.....	197

Part III Case Studies

8	Resilience of Future Internet Communications	201
8.1	Key Research Topics and Requirements for the Future Internet Architecture.....	204
8.2	Network Resource Provisioning Concepts in the “System IIP” Future Internet Architecture.....	207
8.3	Fault Tolerance of Content-Oriented Networking.....	212
8.3.1	The Concept of Survivable Anycasting.....	213
8.3.2	Shared Protection for Survivable Anycasting.....	221
8.3.3	Protection of Information-Centric Communications Against Intentional Failures.....	226
8.4	Summary.....	234
	References.....	234
9	Resilience of Wireless Mesh Networks	239
9.1	Measures of Wireless Mesh Networks Survivability.....	242
9.1.1	Network Model.....	244
9.1.2	Proposed Measures to Evaluate the Survivability of WMNs.....	245
9.1.3	Method of a WMN Survivability Evaluation.....	248
9.1.4	Analysis of Modeling Results and Conclusions.....	250
9.2	A New Approach to the Design of Weather Disruption-Tolerant Wireless Mesh Networks.....	256
9.2.1	Proposed Approach.....	257
9.2.2	ILP Formulation of Weather-Resistant Links Formation Problem (WRLFP).....	259
9.2.3	Computational Complexity of WRLFP Problem.....	260
9.2.4	Analysis of Modeling Results and Conclusions.....	263
9.2.5	Appendix—Rain Radar Maps Used in Simulations.....	267
9.3	Summary.....	269
	References.....	269
10	Disruption-Tolerant Routing in Vehicular Ad Hoc Networks	273
10.1	Reliability Requirements of VANET Applications.....	277
10.2	Network Layer Addressing and Routing Issues.....	279
10.2.1	Unicast Routing with Fixed Addressing.....	281
10.2.2	Unicast Routing with Geographical Addressing.....	281
10.2.3	Multicast Routing with Geographical Addressing.....	282
10.2.4	Broadcast Multi-hop Message Dissemination.....	282
10.3	Improving the Resilience of End-to-End V2V Communications by Multipath Routing.....	284
10.3.1	Probability of V2V Transmission Availability.....	285
10.3.2	Provisioning of Multiple Availability Classes.....	288
10.3.3	Analysis of Modeling Results and Conclusions.....	292

- 10.4 A New Approach to Anypath Forwarding Providing
 - Long Path Lifetime..... 294
 - 10.4.1 Long-Lifetime Anypath (LLA) Concept..... 296
 - 10.4.2 Analysis of Modeling Results and Conclusions..... 300
- 10.5 Summary..... 304
- References..... 305
- Conclusions..... 309**
- Glossary..... 313**
- Index..... 337**

List of Major Symbols

A	Set of directed arcs used to represent directional network links
A_T	Total system availability
a_h	Directed arc
$\hat{a}_{i,j}$	Element of the adjacency matrix
bc_i	Betweenness centrality coefficient for node i
b_h	Amount of capacity to be reserved for backup paths at arc a_h under backup capacity sharing
$b_{r,h,g}$	Binary variable to indicate whether for r -th demand, the failed primary path traverses arc a_g , and the corresponding backup path traverses arc a_h
$b_{h,g}$	Integer variable representing the total capacity needed for backup paths at arc a_h in the case of shared protection provided for working paths traversing the failed arc a_g
$b(i)$	Supply/demand of node i
cc_i	Closeness centrality coefficient for node i
c_h	Total capacity of arc a_h
c_r	Volume of demand d_r
\overline{c}_h	The unused capacity of arc a_h
D	Set of demands
$D^{UN} (D^{AN})$	Set of unicast (anycast) demands
$D^{DS} (D^{US})$	Set of anycast downstream (upstream) demands
d_{avg}	Average node degree
d_i	Degree centrality coefficient for node i
d_{min}	Minimum node degree
E	Set of edges used to represent bidirectional network links
ec_i	Eigenvector centrality coefficient for node i
G	Graph representing the network structure
h	Arc index
$h_{i,j}$	The number of hops for the shortest path between nodes i and j
(i, J)	VANET hyperlink between vehicle i and the set of forwarding vehicles J

J	Set of forwarding vehicles in VANET anypath communications
k	Degree of a network node
l_i	Label of vertex i
N	Set of nodes representing network nodes
$N \setminus T$	Set of edge nodes
P	Set of pre-computed paths
$P(\hat{r}_n)$	Probability of node n failure as a function of the distance \hat{r}_n between node n and the failure epicenter
$P(\delta)$	Probability of occurrence of a failure scenario δ
p	Index of a path from the set P of pre-computed paths
p_i	Index of a predecessor node of node i on a shortest path between the source node and node i
$p_{i,j}$	Layer 2 probability of packet delivery via VANET link (i, j)
$p_{k,h}$	Binary variable indicating whether arc a_h belongs to p -cycle k
$p(r_{i,j})$	Probability density function of inter-vehicle distance
$p_{\Psi}(\psi)$	Probability density function of percentage ψ of flows surviving the regional failure
$R_T(t)$	Total system reliability at time t
r	Demand index
\hat{r}	Radius of a failure region
\hat{r}_n	Distance between node n and the failure epicenter
$S_i(t_0, \Delta t)$	Movement vector of vehicle i in time interval $(t_0, t_0 + \Delta t)$
s_h	Number of channels reserved at arc a_h for protection cycles
$sh_h^{(r)}$	The capacity reserved so far at a_h that may be shared with respect to the backup path of r -th demand
$s_{i,j}$	Stability index of a VANET link between nodes i and j
$s_i^x(t_0, \Delta t)$	Movement of vehicle i in time interval $(t_0, t_0 + \Delta t)$ along X axis
$s_i^y(t_0, \Delta t)$	Movement of vehicle i in time interval $(t_0, t_0 + \Delta t)$ along Y axis
s_r	Source node of demand d_r
T	Set of transit (forwarding) nodes
t_r	Destination node of demand d_r
u_k	Number of units of link capacity needed for p -cycle k
V	Set of vertices of a graph
$v_i(t)$	Velocity vector of vehicle i at time t
$v_i^x(t)$	Velocity of vehicle i at time t along X axis
$v_i^y(t)$	Velocity of vehicle i at time t along Y axis
$v_{r,n}$	Binary variable indicating whether a replica server located at node n is selected as a backup replica of r -th anycast demand; 0 otherwise
w_h	Number of channels already reserved at arc a_h for working paths
$w_{i,j}$	Probability of node j being the forwarding node of a packet received from vehicle i
x_h	The amount of flow served by arc a_h
$x_{k,h}$	Binary variable indicating whether a working path on arc a_h can be protected (i.e., is protectable) by p -cycle k

$x_{r,h}^l$	Binary variable indicating utilization of l -th channel at arc a_h by a working path of r -th demand
$x_{r,h}$	Binary variable indicating utilization of arc a_h by a working path of r -th demand
$x_{r,p}$	Binary variable indicating utilization of path p as a transmission path for demand d_r
(\hat{x}, \hat{y})	X and Y axis coordinates of the failure epicenter
(\hat{x}_n, \hat{y}_n)	X and Y axis coordinates of node n
$y_{r,h}$	Binary variable indicating utilization of arc a_h by a backup path of r -th demand
δ	Region failure scenario given by the set of non-operational nodes (after the outage)
$\delta_{h,r,p}$	Binary variable indicating utilization of arc a_h by path p to serve demand d_r
ζ_h	Unitary cost of arc a_h in backup path computations under backup path sharing
$\kappa(G)$	Vertex connectivity of graph G
$\kappa_{r,n}$	Binary variable indicating whether a replica server located at node n is selected as a working replica of r -th anycast demand
Λ_h	The number of channels available at an optical link a_h
$\lambda(G)$	Edge connectivity of graph G
λ	Failure rate
λ_i	Wavelength of an optical channel associated with channel i at the optical link
μ	Repair rate
ξ_h	Unitary cost of arc a_h in working path computations
$\hat{\xi}_i$	The cost of the shortest path from the source node of a given path to node i
ρ_h	Probability that two vehicles are connected by link a_h at any time t
σ_{deg}	Standard deviation of node degrees
$\sigma_{i,J}$	Cost of a VANET hyperlink (i, J)
$\sigma_{i,t}$	Cost of a VANET path between nodes i and t
σ_J	Cost of a VANET anypath from set J to the destination node
$\sigma_{s,t}$	The number of shortest paths between different node pairs s and t
$\sigma_{s,t,i}$	The number of shortest paths between different node pairs s and t traversing node i
$\tau(r)$	Index of a demand associated with demand d_r
$\nu_{r,p}$	Cost of a pre-computed path p to serve demand d_r
$\varphi(x)$	Objective function
$\Psi(\delta)$	Random variable referring to the percentage ψ of flows delivered in scenario δ
$\omega_h(t)$	Estimated signal attenuation due to rainfalls for arc a_h at time t

Acronyms

ACA	Average Content Accessibility
ACS	Access Station
ADM	Add-Drop Multiplexer
AODV	Ad hoc On-demand Distance Vector
APF	Active Path First
APS	Automatic Protection Switching
ATM	Asynchronous Transfer Mode
ATTR	Average Two-Terminal Reliability
B&B	Branch-and-Bound
BC	Betweenness Centrality
BER	Bit Error Rate
BGP	Border Gateway Protocol
BLP	Binary Linear Programming
BLSR	Bi-directional Line Switched Ring
C2C-CC	Car-to-Car Communications Consortium
CAM	Cooperative Awareness Message
CAN	Content-Aware Networking
CC	Closeness Centrality
CCH	Control Channel
CCN	Content-Centric Networking
CDN	Content Delivery Network
CON	Content-Oriented Networking
CoS	Class of Service
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CIST	Common and Internal Spanning Tree
CST	Common Spanning Tree
D ² R ² + DR	The resilience strategy involving the defend, detect, remediate, recover, diagnose, and refine phases
DC	Degree Centrality
DDoS	Distributed Denial of Service

DoS	Denial of Service
DNS	Domain Name System
DSRC	Dedicated Short Range Communications
DSS	Data Stream Switching
DTN	Delay Tolerant Networking
DWDM	Dense Wavelength Division Multiplexing
E/O	Electrical-to-Optical
EC	Eigenvector Centrality
EMP	Electromagnetic Pulse (attack)
ENISA	The European Union Agency for Network and Information Security
ETSI	European Telecommunications Standards Institute
ETT	Expected Transmission Time
EXT	Expected Transmission Count
FCC	Federal Communications Commission
FI	Future Internet
FIA	Future Internet Assembly
FIT	Failures in Time
FSO	Free-Space Optical
FTP	File Transfer Protocol
Gb/s	Gigabit per second
GHz	Gigahertz
GMPLS	Generalized Multiprotocol Label Switching
GPS	Global Positioning System
HCI	Human-Computer Interaction
HTTP	Hypertext Transfer Protocol
ICN	Information-Centric Networking
IETF	Internet Engineering Task Force
IFIP	International Federation for Information Processing
ILP	Integer Linear Programming
InP	Infrastructure Provider
IoT	Internet of Things
IP	Internet Protocol
IPDV	IP packet Delay Variation
IPER	IP packet Error Ratio
IPFRR	IP Fast-Reroute
IPLR	IP packet Loss Ratio
IPTD	IP packet Transfer Delay
IPv6	Internet Protocol version 6
IS-IS	Intermediate System to Intermediate System
ISP	Internet Service Provider
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector
IVC	Inter-Vehicular Communications
LOS	Line of Sight
LF	Largest First

LFA	Loop-Free Alternates
LP	Linear Programming
LSA	Link State Advertisement
LSP	Label Switched Path
μ -ACA	Mean Content Accessibility
MAC	Media Access Control
MANET	Mobile Ad hoc NETWORK
Mb/s	Megabit per second
MCFP	Minimum Cost Flow Problem
MDT	Mean Downtime
MFlop	Mega Floating-point operations
MIMO	Multiple-Input Multiple-Output
MILP	Mixed Integer Linear Programming
MIVC	Multi-hop Inter-Vehicular Communications
MOS	Mean Opinion Score
MPLS	Multiprotocol Label Switching
MSTI	Multiple Spanning Tree Instance
MSTP	Multiple Spanning Tree Protocol
MTBF	Mean Time Between Failures
MTBI	Mean Time Between Interruptions
MTBM	Mean Time Between Maintenance
MTFF	Mean Time to First Failure
MTRS	Mean Time to Restore Service
MTTF	Mean Time to Failure
MTTR	Mean Time to Recovery/Repair
MUT	Mean Uptime
NDO	Named Data Object
NLP	Non-Linear Programming
NNI	Network-Network Interface
NSF	National Science Foundation
NVE	Network Virtualization Environment
O/E	Optical-to-Electrical
O/E/O	Optical-to-Electrical-to-Optical
OBU	On-Board Unit
OSPF	Open Shortest Path First
OTI	Organizational, Technological, and Individual
OTN	Optical Transport Network
OXC	Optical Cross Connect
P2P	Peer-to-Peer
PDR	Packet Delivery Ratio
PER	Packet Error Rate
PHY	Physical Layer
PIA	Percent of IP service Availability
PIU	Percent of IP service Unavailability
PLR	Packet Loss Ratio

POI	Point of Interest
PWCE	Protected Working Capacity Envelope
PSY	Path Symmetry
QoE	Quality of Experience
QoR	Quality of Resilience
QoS	Quality of Service
QLRM	Qualitative Robustness Metric
QNRM	Quantitative Robustness Metric
rLCC	Relative Size of the Largest Connected Component
RBD	Reliability Block Diagram
RIP	Routing Information Protocol
RFC	Request for Comments
RMP	Restricted Master Problem
RREP	Route Response
RREQ	Route Request
RSTP	Rapid Spanning Tree Protocol
RSU	Road-Side Unit
RTT	Round Trip Time
SCH	Service Channel
SDH	Synchronous Digital Hierarchy
SDN	Software-Defined Networking
SHR	Self-Healing Ring
SIVC	Single-hop Inter-Vehicular Communications
SLA	Service Level Agreement
SLB	Service Loss Block
SNR	Signal-to-Noise Ratio
SONET	Synchronous Optical Network
SOS	Standard Deviation of Opinion Scores
SP	Service Provider
SPP	Shortest Path Problem
SPT	Shortest Path Tree
SRLG	Shared Risk Link Group
STP	Spanning Tree Protocol
TCP	Transmission Control Protocol
TDM	Time Division Multiplexing
UNI	User-Network Interface
UPSR	Unidirectional Path-Switched Ring
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VANET	Vehicular Ad hoc NETWORK
VLAN	Virtual Local Area Network
VN	Virtual Network
VoIP	Voice over IP
VPN	Virtual Private Network
VRU	Vulnerable Road User

VSCC	Vehicle Safety Communications Consortium
WDM	Wavelength Division Multiplexing
Wi-Fi	Wireless Fidelity
WMD	Weapon of Mass Destruction
WMN	Wireless Mesh Network
WSN	Wireless Sensor Network

Part I
Introduction to Networked Systems
Resilience

Chapter 1

Fundamentals of Resilience of Communication Networks and Networked Systems



Communication networks have become essential for the everyday operation of our society [2]. In particular, the Internet, considered an indispensable part of the critical information infrastructure for several personal and business applications, is now expected to be *always available* [29]. Availability of network services is thus an essential aspect of Service Level Agreements (SLAs) between service providers and customers, apart from typical quality of service (QoS) parameters such as throughput, latency, jitter (packet delay variation), or packet losses [11, 17]. Any disruption of end-to-end routing, even lasting for a short time, commonly induces severe economic losses and remarkably affects the network provider's reputation.

This chapter elaborates on the resilience aspect and its importance for networks. Our analysis is dedicated primarily to communication networks (particularly infrastructure elements such as routers, switches, and communication links) and their services (e.g., IP communications, VoIP, or Wi-Fi). These investigations are also valid for more general architectures of *networked systems* composed of interconnected elements (including servers and computing units) providing storage, computation, and communication services. Therefore, the overall objective of this chapter is to highlight crucial aspects of the resilience of communication networks and, generally, of networked systems. For the sake of simplicity, the term *network* is used in this book in both contexts.

A considerable amount of content exchanged in the core part of a communication infrastructure requires high-capacity storage, processing, and transmission capabilities. In scenarios of failures of network nodes/links, thousands of flows may be affected, and a significant amount of data (measured in terms of terabits) may be lost [25]. For instance, an OC-48 optical link downtime equal to 10 s causes about three million packets of the average size of 1 kB to be dropped [28].

Table 1.1 shows the upper thresholds of Quality of Service (QoS) parameters, as identified for Internet Protocol (IP) networks by International Telecommunication Union—Telecommunication Standardization Sector (ITU-T) in Y.1540 and Y.1541 recommendations for different service classes expressed in terms of IP packet Loss

Table 1.1 Values of QoS parameters for different ITU-T service classes based on [34]

Class of service	Examples of applications	IPLR	IPER	IPTD	IPDV
Class 0	Real time, jitter-sensitive, highly interactive	1×10^{-3}	1×10^{-4}	100 ms	50 ms
Class 1	Real time, jitter-sensitive, interactive	1×10^{-3}	1×10^{-4}	400 ms	50 ms
Class 2	Transaction data, highly interactive	1×10^{-3}	1×10^{-4}	100 ms	ND
Class 3	Transaction data, interactive	1×10^{-3}	1×10^{-4}	400 ms	ND
Class 4	Low loss only (e.g., short transactions, bulk data, video streaming)	1×10^{-3}	1×10^{-4}	1 s	ND
Class 5	Traditional applications of default IP networks	ND	ND	ND	ND

ND—not defined

Ratio (IPLR), IP packet Error Ratio (IPER), IP packet Transfer Delay (IPTD), and IP packet Delay Variation (IPDV).

As discussed in [5], SLAs commonly include requirements on:

- High network availability (e.g., of 99.99%, or higher—like 99.999% availability for telemonitoring, or telesurgery applications, also called the “five nines” property [26]).
- Short time of service recovery after a failure. In particular, for stringent services, it is necessary to provide service recovery time below 50 ms [36] (which is compliant with the respective value of IPDV parameter for service classes 0 and 1 from Table 1.1).

However, due to continuous technological progress resulting in both increasing the transmission ratio and reducing the transmission delay (even as low as 1 ms, as in the case of 5G networks [3]), the values shown in Table 1.1 may now be out of date for services and requiring appropriate updating.

Failures of network elements, as well as failures of services provided by them, are inevitable mainly due to the complexity of elements themselves, numerous flaws likely to be incorporated into the elements either in their design, deployment, or maintenance phases, as well as a large set of external factors (e.g., natural disasters or malicious attacks) posing a significant threat to the proper functioning of networks.

The remaining part of this chapter is organized in the following way. We start our investigations by analyzing in Sect. 1.1 common scenarios for failures of network elements and provide their detailed classification referring to the number, location, duration, and extent of failures. Next, we focus in Sect. 1.2 on defining network resilience and its desired properties. Our discussion continues in Sect. 1.3, where we explain typical consecutive phases of recovery of services in scenarios of failures of network elements. The impact of human and organizational issues on the resilience of networked systems is discussed in Sect. 1.4. After that, in Sect. 1.5, we analyze

the potential costs and comment on evident benefits following the deployment of resilience mechanisms. Section 1.6 provides a discussion of other significant challenges to the deployment of resilience mechanisms in networks, while the summary of the chapter is given in Sect. 1.7.

1.1 A Classification of Failures of Network Elements

This section provides a classification of failures of network elements referring to the four main aspects: the number of elements affected by these failures, the location of failed elements, and the duration and extent of failure events. A detailed classification of failures is illustrated in Fig. 1.1.

Concerning the number of affected elements, failure events can be broadly classified into *single failures* (i.e., referring to failures of single communication links or single network nodes—such as single switches or servers at a time) or *multiple failures* denoting simultaneous failures of several system elements. It is important to note that a failure of multiple elements does not necessarily mean that these failures occur at exactly the same time. In the literature, it is also expected to classify as a multiple failure a scenario in which a failure of a next element occurs before the physical repair of the previously failed element has been completed [39].

Based on the nature of failures, we can distinguish either *random failures* (i.e., failures occurring accidentally in areas that do not depend on the external characteristics of the environment) or *regional failures* confined to specific geographic regions [9]. Concerning the duration aspect, failures can be divided into *transient failures* [28], i.e., failures which tend to subside when the factor affecting the element ceases to exist, as, for instance, in the case of a temporal impact of dense fog on decreasing the capacity of wireless links in Free-Space Optical networks [37] and *non-transient failures* (i.e., permanent failures requiring a physical repair of failed elements). Indeed, about 50% of failures are identified as transient and last less than a minute [28]. It refers, for example, to disruptions of communication paths observed in IP networks where routing protocols (e.g., Open Shortest Path First—OSPF) can reroute the traffic reactively upon a failure.

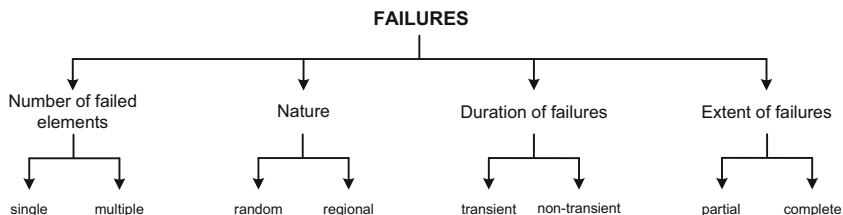


Fig. 1.1 Classification of failures of network elements

Other major scenarios of relatively short-lasting failures are attributed to wireless networks. For instance, Wireless Mesh Networks (WMNs) with stationary nodes connected by high-frequency wireless links often encounter time-varying weather-based disruptions partially or completely degrading the available link capacity (e.g., as a result of a heavy rain storm) [20]. In mobile Vehicular Ad hoc NETWORKS (VANETs), the lifetime of communication links (and thus also the availability of links and end-to-end communication paths) is commonly measured in seconds due to the high mobility of vehicles [21, 23].

Finally, failure scenarios can be divided based on the extent of failures into those referring to *partial failures* [32] denoting failures of some parts of a network element, e.g., some ports of a switch or *complete failures* of certain elements.

Failures of network nodes and links interrupting a network's normal functioning are relatively common for various reasons. Following [24], 20% of the failures emerge from scheduled maintenance activities (for instance, updates of the network architecture). About 55% of the other failure events denote random failures of single links caused unintentionally by third parties (e.g., cuts of links located underground during dig-ups). Quite often, such events simultaneously affect more than one link at a time (particularly if several links placed together in a duct are cut simultaneously).

The remaining 25% of failures of links/nodes mainly refer to disaster-based failures identified in [9, 29] as following from:

- Natural disasters including, e.g., earthquakes, floods, or fires
- Malicious attacks aimed at causing severe losses at minimum cost (often implying failures of high-degree nodes or high-capacity links serving most of the traffic)
- Technology-related disasters triggered by technological events such as power blackouts or faults incorporated into the structure of system elements at various phases of their lifetime

About half of multiple failure scenarios are related to links not connected to the same node [22]. Also, the risk of large-scale failures due to natural disasters (or human-made disasters) is rising.

Disaster-based failures are far more dynamic and broader in scope than classical random failures. They commonly result in simultaneous failures of network elements located in specific geographic regions [16]. For instance, every year, tens of hurricanes worldwide are responsible for power outages, disrupting communications on a massive scale for a long time (10 days, on average) [14]. Hurricane Katrina, which caused severe losses in Louisiana and Mississippi in the Southeastern USA in August 2005 [35], is only one of them.

Earthquakes are the reasons for even greater destruction due to long times of manual repair actions. A notable example is the 7.1-magnitude earthquake in December 2006 in Southern Taiwan, which resulted in simultaneous failures of seven submarine links and visibly affected the Internet connectivity between Asia and North America for weeks. As a result, international communications to China, Taiwan, Hong Kong, Korea, and Japan immediately became impossible [35]. Similarly, the Greatest Japan Earthquake of 9.0 magnitude on March 11, 2011,

caused wide-scale damage to undersea cables and impacted about 1500 telecom switching offices due to power outages [9].

Based on the occurrence of disaster symptoms, disasters can be either predictable (e.g., hurricanes or floods) or non-predictable (e.g., earthquakes) [27]. An important observation is that disaster events often lead to *cascading failures*, meaning that the failure of a particular element of a networked system (e.g., due to the earthquake) can next trigger correlated failures in other parts of the network (e.g., due to power outages after the earthquake) [9].

Another essential reason for failures of network elements refers to intentional human activities, e.g., bombing, use of weapons of mass destruction (WMD) [2], or electromagnetic pulse (EMP) attacks. Such activities affect the ability of networks to fulfill their QoS requirements. As reported in [29, 38], EMP events can be triggered by human activities using a nuclear explosion at a high altitude in the atmosphere or using nonnuclear equipment utilizing powerful radio frequency (RF) devices. Solar storms can also trigger EMP events, for instance, the solar storm in Quebec, Canada, in 2018 [29]. Other examples of malicious human activities refer to events aimed at disturbing the operation of selected elements of a network that are particularly important for delivering services (such as *Distributed Denial of Service—DDoS* attacks) [1] or attacks leading even to physical destruction of specific network elements [29].

Technology-related failures are reported to happen primarily due to internal events referring to faults of the software associated with certain network elements or faults of elements incorporated at various stages of their lifetime—either at the design, deployment, or maintenance (configuration) phase [29]. The consequences of such events can be remarkable. As an example, a 5-min outage that occurred at Google in August 2013 was responsible for a reduction of the overall Internet traffic by 40% due to the unavailability of primary Google services such as Google Drive, Google Search, and YouTube [13]. External events such as power blackouts or solar storms can also trigger technology-related failures. Power blackouts can indeed cause massive failures in networks, such as the blackout in 2006 that was harmful to 10 million users in Europe, the 2009 blackout affecting 87 million users in Brazil and Paraguay, or the 2021 blackout in India affecting 670 million users [31].

1.2 Network Resilience

Among several definitions of network resilience available in the literature, the most adequate and widely accepted seems to be the one by Sterbenz et al. from [35] referring to *the ability of the network to provide and maintain an acceptable level of service in the face of various faults and challenges to normal operation*. This definition was one of the research outcomes of the FP7 EU ResumeNet project [30] and was next adopted by ENISA—The European Union Agency for Network and Information Security [11].

As discussed in [11], in this definition:

- *Acceptable level of service* is characterized by service parameters measured in time and commonly specified in the SLA, such as service availability, throughput, delay, delay variation (jitter), and packet losses.
- *Provide* refers to the delivery of services according to the SLA under normal conditions.
- *Maintain* focuses on the assurance of service delivery at an acceptable service level in scenarios of failures of system elements by utilizing appropriate resilience mechanisms aimed at minimizing the negative consequences of failures as well as restoring the affected services (i.e., services that became no longer available due to failures, or services for which the level of delivery happened to decrease below the accepted threshold).
- *Faults* denote flaws (imperfections) that arise at various stages of design, implementation, and maintenance of system elements and services, which can be potential obstacles/disturbances to their proper functioning. These can be in hardware or software (in the latter case, the flaws are usually termed “bugs”).
- *Challenges* represent risks to the network and its services, comprising events such as large-scale disasters, malicious human activities, hardware destruction, failures of service providers, operational mistakes, or a rapid increase of legitimate traffic above the level acceptable by the system.

Figure 1.2 presents a typical scenario of the service level degradation (for instance, due to a failure of a network element), illustrating the periods when the service is provided, as well as periods when the service can be considered as either maintained (when its level is still above the minimum service level) or not delivered.

Proper delivery of services in a normal scenario and in challenging conditions requires continuously measuring service properties using relevant metrics. Such measurements should be quantifiable (i.e., based on quantitative measurements), repeatable (i.e., provide the same result in consecutive trials), and comparable to obtain accurate and usable information [11].

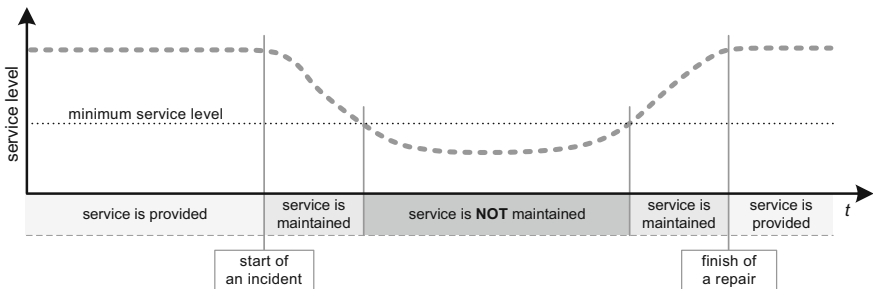


Fig. 1.2 An example illustration of periods of service delivery in a failure scenario

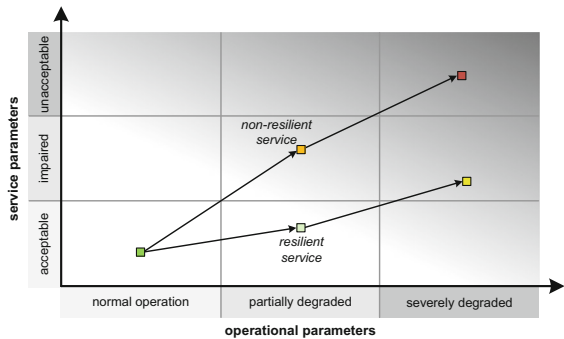
To reduce the negative consequences of failures, it is essential to identify potential risks, assess them quantitatively (or qualitatively), and determine their likelihood of occurrence and possible impact on network services.

A significant subset of risk factors may be controlled by deploying schemes based on mechanisms of *redundancy* typically applied at three main physical levels:

1. The hardware level. In this case, the internal architecture of network elements comprises duplication of specific circuits/components. As a result, a remarkable number of internal failures of elements are perceived as not impacting the correct operation of a network (the related internal failures of elements are not manifested by them in the network).
2. The network architecture level. It can be achieved by the multiplication of some network elements to increase the operational resilience of certain parts of the network. For example, connectivity with the Internet may be provided by two links (instead of one link) to lower the likelihood of Internet inaccessibility.
3. The network services level. Examples include, e.g., schemes of resilient routing using *alternate (backup) paths* to redirect the traffic upon the failure of a network element affecting the *primary (working) paths*. Traditionally, to protect against failures of single links/nodes, any backup path should not traverse the transit links/nodes used by the primary path being protected.

The perception of the level of services by the end users depends on the operational characteristics of the network as well as the related service parameters, as illustrated in Fig. 1.3 and discussed in [11]. Under normal network operation, providing services at an acceptable level is relatively easy, regardless of deploying resilience mechanisms. However, in scenarios of minor/major degradation of network parameters (e.g., due to failures of network elements), service parameters tend to change toward impaired and even unacceptable. The dynamicity of these changes naturally depends on the implementation (or not) of network resilience mechanisms, leading to two potential transitions of the service level perception, as illustrated in Fig. 1.3. It follows from a general conclusion that for network architectures not equipped with resilience mechanisms, the perception of the level of services by the

Fig. 1.3 A distinction of a resilient and a non-resilient service



end users in failure scenarios tends to drop much quicker than when relying on resilience mechanisms.

1.3 Recovery of Services

System element failures may harm numerous communication, storage, and computation services. In this section, we explain typical phases of service recovery for a communication service provided by network elements. However, it should be noted that the respective phases for recovery of services of other categories (i.e., computation and storage) may have a similar meaning. Figure 1.4 illustrates the consecutive phases described in [6, 8, 25] needed to recover the communication service—e.g., a multi-hop communication path affected due to a failure of a network link/node traversed by that path.

Communications typically become no longer possible along a given primary (working) path if this path traverses a failed network link (or node). To initiate the recovery procedure after the occurrence of a fault in the system, *fault detection* needs to be performed by the respective mechanisms of either the control plane or the data plane [7]. Fault detection in the control plane is possible by the elements of a network localized in a neighborhood of a faulty element by identifying the Loss of Clock, Loss of Modulation, Loss of Signal, or noticing the increased signal-to-noise ratio (SNR) [8]. Concerning fault detection in the data plane, the related analysis may refer to the values of Bit Error Ratio (BER), cyclic redundancy check (CRC), and TCP checksum verification, as well as be based on the identification of incorrect values of the end-to-end quality of service parameters such as, e.g., increased delay or lower throughput [7].

Fault detection is followed by *fault localization* aimed at the determination of the location of a failed element needed to identify a network node being a neighbor of a failed element traversed by the affected primary path, at which further transmission along this path should be suspended [8], marked as operation (1) in the example Fig. 1.5. For local rerouting schemes, precise identification of such a node is even more critical, as it is expected to serve as the beginning of a local detour path around a failed element. Faulty elements in communication networks can be localized using

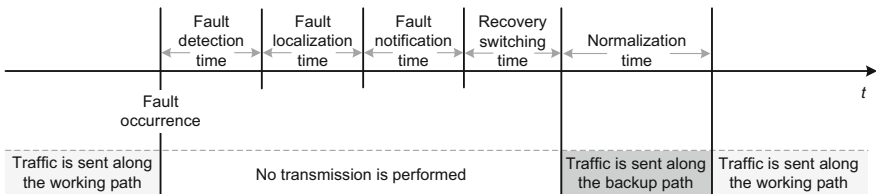


Fig. 1.4 The timeline of recovery steps for a communication path affected by a network element failure

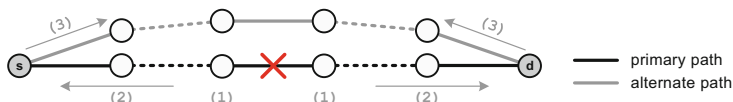


Fig. 1.5 Illustration of recovery steps in a scenario of a failure of a network element affecting the primary path

either passive methods (which are based on the analysis of alarm signals received from the monitoring agents) or active schemes involving sending dedicated packets (such as, e.g., ping messages) by probing stations to actively localize a point of failure [10].

During the *fault notification* phase, the respective messages informing about the failure are sent by nodes neighboring to the faulty element toward the respective end nodes of the protected segment of the primary path to initiate the activation of the backup path [4] (see operation (2) in Fig. 1.5). As fault notification time depends on the size of a segment of the working path protected by a given backup path, it has a minor impact on the total recovery time for local recovery schemes.

In the *recovery switching* phase, the flows originally served by the affected working paths are redirected onto the corresponding backup paths, illustrated as operation (3) in Fig. 1.5. It is important to note that for transmission schemes based on the reactive determination of backup paths after the occurrence of failures of network elements, the recovery switching phase takes visibly more time, as it additionally includes calculation of the related backup paths and reservation of link resources [4] performed during the recovery switching phase. Such computations of backup paths are, in turn, not needed in the recovery switching phase for proactive schemes, where backup paths are determined at the time of setting up the related working paths.

The period of using the backup paths after the recovery switching phase is naturally not endless. During this period, often called the *normalization* phase, the restoration of a network to its normal operational state (involving, e.g., a physical repair/replacement of failed elements) takes place, and, as soon as it is completed, the redirected flows are reverted onto the original (i.e., working) paths [6, 7].

Depending on a given failure scenario, recovering all the affected network flows at the original QoS levels is often impossible. This is because, in a post-failure period, it may turn out that the total available network capacity (which was reduced as a result of the failure) is lower than the total requested capacity (as illustrated in Fig. 1.6) and will return to its original level only after completing the operations of physical repair/replacement of the affected network elements.

This problem can be even more visible in the case of disaster scenarios, where the network is likely to become affected even harder (due to multiple failures likely to occur) as well as due to a probable rapid increase of network traffic driven by the post-disaster behavior of users intensifying their efforts in getting access to disaster-related information as well as communicating more intensively with their relatives [29], as illustrated in Fig. 1.7.

Fig. 1.6 An example relation between the total requested capacity and the available network capacity in a failure scenario

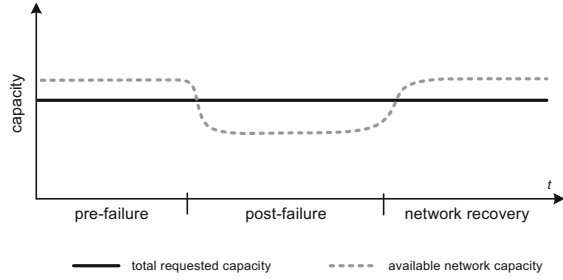
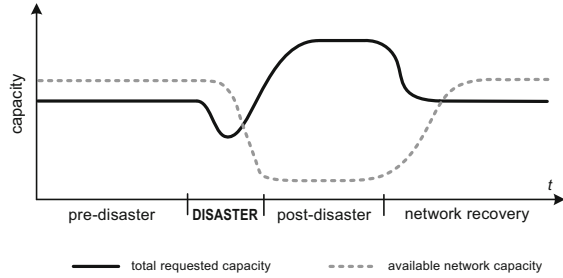
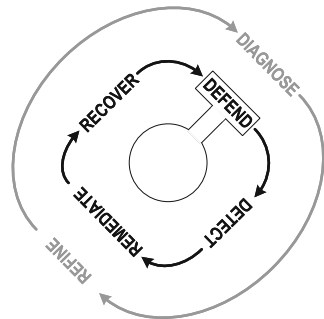


Fig. 1.7 An example relation between the total requested capacity and the available network capacity in a disaster scenario



To reduce the risk of the unsuccessful recovery of some services due to events of unexpected type or volume, deploying a strategy of continuous evaluation and improvement of resilience mechanisms to prepare the network better for incoming disruptions is necessary. The D^2R^2+DR strategy from [33] is particularly appropriate for this purpose. This strategy, illustrated in Fig. 1.8, consists of two loops that operate in almost real time. The inner loop refers to the four consecutive operations: defend, detect, remediate, and recover. This loop is used in the diagnosis and recovery of the network following the occurrence of a given disruption. In particular, the *defend* phase involves analyzing a network’s functioning and setting up the network resilience mechanisms (e.g., redundant elements or backup paths) in advance. The objective of the *detect* phase is to recognize anomalies by analyzing symptoms or effects of a disruption. During the *remediate* phase, actions such as redirection of the affected traffic onto the related backup paths are performed, while

Fig. 1.8 D^2R^2+DR resilience strategy from [33]



the *recover* phase is to finalize the (physical) restoration of the network to its normal state (i.e., from before the occurrence of a disruption).

The outer loop, in turn, operates in a longer timescale, presumably offline, and involves both computational tools and human experts. Its objective is to evaluate the efficiency of network recovery mechanisms after finalizing the operations from the inner loop following each disruption encountered by the network (the *diagnose* phase) and, next, to apply the necessary updates in the configuration of the network recovery schemes (the *refine* phase), so that the network can handle the incoming future disruptions more effectively.

1.4 The Impact of Human and Organizational Issues on the Resilience of Networked Systems

As discussed in Chapter 32 of [29], issues of resilience in networked systems are not only driven by technical aspects typically resolved by the related technical procedures. As noticed by Hutchison and Sterbenz in [19], when modeling the resilience mechanisms for complex networks, there is a strong need to address the role and involvement of people as system components in this analysis.

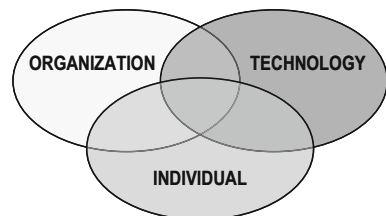
Indeed, apart from *the technology viewpoint* already discussed in this chapter, people involved in the operation of every networked system pose challenges related to:

- *The organizational viewpoint* with the related policies, processes, and procedures determined in the organization for joint activities of people working together for a common purpose, i.e., defining the operations of groups of people
- *The individual viewpoint* representing the ways individual people behave under certain circumstances

These three aspects together form the so-called OTI scheme covering organizational, technological, and individual points of view [15, 29] illustrated in Fig. 1.9.

Apart from issues that can be classified as representing certain viewpoints exclusively (e.g., technical aspects of the network structure), a large subset of aspects refers to more than one viewpoint. For instance, following the recent COVID-19 pandemic, it is becoming increasingly common that employees frequently work

Fig. 1.9 A mutual interaction of organizational, technological, and individual viewpoints



remotely using mobile devices, creating threat scenarios that go far beyond the challenges of individual viewpoints.

The OTI analysis is useful to increase awareness of nontechnical risks, especially in the context of human and organizational aspects, to deploy more resilient strategies [29]. In this context, the analysis of only technical risks is highly insufficient. In particular, when analyzing the nontechnical aspects of the organization, it is essential to focus on exploring the policies used within the organization and on understanding the behavior of employees (e.g., what are the ways people conduct their everyday work tasks), their roles, and responsibilities, as well as the social relations among people [19, 29].

Any networked system is indeed as weak as its weakest component. Unfortunately, this observation relatively often applies to people. However, it is essential to note that people can also act as sources of strength for a networked system [19], for instance, in conducting physical repair operations following failures of system elements. Therefore, people's impact on a networked system's resilience is undoubtedly complex. As such, it requires a detailed analysis and should be considered in a networked system's design, deployment, and maintenance phases.

1.5 Deployment of Resilience Mechanisms: Costs and Benefits

Deployment of resilience mechanisms typically implies using alternative means for provisioning network services in scenarios of failures of network elements originally utilized to deliver these services. This, in turn, raises the need to install and use certain redundant elements (such as routers and servers) as well as to reserve more network resources (link capacities) to set up additional (backup) paths. These aspects imply that the resilience of networks comes at an extra cost. This is true, provided the related redundant elements and structures are deployed exclusively for resilience. However, in many network configurations, the cost of resilience can be reduced significantly (if not entirely) [18]. In this context, the most important observations referring to the *cost of resilience* are as follows:

- Certain forms of redundancy often already exist in networked systems, whether or not increasing their resilience was the main reason for their implementation. Examples include, e.g., deploying multiple copies of servers in a networked system with their original purpose to ensure the scalability of the related services (via load balancing and reduction of the overall transmission delay). In such configurations, there is only one step ahead toward the resilience of such an architecture: to set up the resilient anycast communication scheme, where one of many servers is meant to respond to the user's request (regardless of how many of them are operational).
- Link capacity reserved for backup paths can be shared between these paths as long as they are designed to protect working paths that are not expected to

fail simultaneously. This, in turn, can reduce the amount of excessive network resources needed for backup paths.

- Due to a variety of network traffic characteristics (allowing for grouping the traffic into classes related to the quality and priority of service), the overall cost of resilience can be significantly lowered, e.g., if low-priority traffic is served during normal operation of a system using link capacities reserved originally for backup paths.
- Concerning the operational costs of the network mainly following from the energy consumption, deployment of resilience schemes can be enhanced by several techniques to reduce the overall energy consumption, including utilization of renewable energy sources, all-optical communication solutions to avoid energy-inefficient signal conversions between the optical and electrical domains, or the use of sleep mode for backup paths.

Many examples of network failures, particularly those triggered by disasters, show that the right approach is not to judge the need to implement resilience mechanisms only from the perspective of their cost of deployment. Numerous failures in real environments indicate that resilience should be viewed as a property that can pay for itself [18]. Indeed, in a non-resilient network, the adverse financial and societal outcomes of even one such failure event can be far more severe than the cost of implementing the related resilience mechanisms.

It is clear that investing in network resilience mechanisms can provide remarkable and indisputable benefits. In particular:

- Resilience allows to avoid substantial financial losses for both users and service providers in failure scenarios, especially those referring to massive failures leading to long-lasting problems of disconnectivity and unavailability of services outlined in Sect. 1.1 of this chapter. This is indeed a severe issue since scenarios of multiple failures, particularly those following natural disasters, are continuously increasing in number, intensity, and scale.
- Mechanisms of resilience can also reduce the risk of improper functioning of networks and services provided by them in scenarios of malicious human activities (attacks) occurring frequently, targeted at the most critical elements of networked systems (see, e.g., DDoS attacks), aimed at degrading the performance of a system or disrupting the proper functioning of services and conducted for reasons ranging from economical to even political ones.
- As architectures of communication networks and networked systems, as well as of the internal structures of network elements and the related software, are becoming increasingly complex, the frequency of technology-induced failures is also increasing. Interdependencies additionally magnify the problem among systems of different types (e.g., a mutual dependence between a communication network and a power grid), which can be responsible for even a total collapse of such interdependent systems, even though the related root cause of such a collapse might be considered minor (e.g., a failure of a single element in one of these systems). The related mechanisms of resilience, able to mitigate the effects

of such interdependent failures, are thus critical for the proper functioning of contemporary highly interdependent systems.

- As public communication networks often turn out to be the only means of communication in disaster scenarios, their proper functioning in challenging conditions can even save human lives. Indeed, communication networks and networked systems have become an integral part of critical infrastructures [12]. The authorities often use public networks in disaster scenarios (such as those caused by fires, floods, earthquakes, or volcano eruptions) when their own communication systems become affected by a failure event or to reach civilians broadly with emergency messages. With proper resilience mechanisms, the efficiency of such operations could be significantly improved. For instance, a lack of adequate mechanisms of network resilience made communications (also of rescue team members) during the 2018 fires in the Attica region in Greece in the affected areas hardly possible.
- Communication networks and networked systems provide end users with an extensive range of services, including remote work possibilities. In this context, the importance of the correct functioning of public communication networks anytime and anywhere (especially in periods of increased legitimate network traffic) has become more critical during the recent COVID-19 pandemic and is expected to remain at least as important in the future.

A conclusion following this analysis is that the profits from deploying resilience mechanisms significantly exceed the costs of their implementation. Some of the gains seen, e.g., in scenarios of rescue operations, are almost incalculable. Therefore, it is clear that resilience mechanisms should be treated as an integral aspect of the design of any network architecture.

1.6 Other Challenges to the Deployment of Resilience Mechanisms

Despite obvious benefits from the use of resilience mechanisms in networks and various ways of reducing the cost of resilience highlighted in the previous section, a number of factors still seem to limit the scale of their deployment. In particular:

- It is often assumed that networks provide a satisfactory level of service, whatever challenges they encounter. A large group of engineers involved in network design is usually of the opinion that an increase in expenditure on enhancing the level of resilience is largely unjustified. However, examples of real-world network failures (as described in Chapter 1 of [29]) strongly contradict this assumption. Indeed, contemporary networks fail more and more frequently due to new and growing problems. Basic resilience mechanisms often turn out to be inefficient in scenarios of more demanding failure events.

- Despite the availability in the literature (especially since the 1990s) of many methods for resilient provisioning of services, in particular, including those referring to multi-hop transmission, few of them have been implemented in real network systems due to the limitations mainly imposed by the methodology in use at the time for management of networks and configuration of their elements. However, along with the growing popularity of programmable networks (e.g., using the idea of Software-Defined Networking) in many environments, implementing resilience mechanisms has recently become easier than ever.
- During the phase of physical repair of damaged network elements (e.g., wired links that have been burnt due to fire), it is relatively common that operators face difficulties in accessing the affected areas immediately, mainly due to land ownership rights issues and other local legal conditions. This, in turn, can lead to a visible extension of the time needed to complete the physical repair phase and an increase in the unavailability of services. Therefore, it remains crucial to develop independent standards for repair procedures in failure scenarios, allowing the network equipment owners to operate without unnecessary additional delays. This, in turn, requires the involvement of representatives from network operators, equipment vendors, scientists, and regulators in standardization works.
- In contracts between service providers and users (i.e., SLAs), resilience is represented most by service availability and the level of packet losses. Although these parameters are included in the SLAs, there are often several exclusions added to these agreements, e.g., related to ignoring the periods of service unavailability caused by occurrences of natural disasters in the calculation of the availability value. Such exclusions, in turn, increase the risk of not implementing a significant set of resilience mechanisms. Therefore, there is a clear need to work on standardization for such contracts, particularly regarding their validity in a broad spectrum of failure scenarios. Otherwise, the networks may not cope with many failure events, which seems crucial for networks expected to play a socially important role in the critical communication infrastructure.

1.7 Summary

Communication networks and, generally, networked systems are susceptible to many challenges that may often lead to failures of their elements or seriously limit the ability of networks to provide services to users.

In this introductory chapter, we first highlighted the most common failure scenarios in networks, provided the definition of network resilience and its desired properties, and explained typical phases of service recovery in failure scenarios. Next, we explained the need to focus also on other (nontechnical) challenges to the resilient operation of networks and their services (following from the organizational and individual viewpoints). The later part of the chapter investigated the costs and benefits of deploying resilience mechanisms in networks and analyzed other challenges to deploying resilience mechanisms.

Communication systems that support almost all our daily activities and often enable differentiated critical services (and thus serve as critical infrastructures) undoubtedly need to be resilient. Assuring their resilience, in turn, requires a systematic approach. Also, it is risky to assume that networks will continue to provide services at an acceptable level without adequate investments in resilience mechanisms. Indeed, given a rich set of challenges to the resilient operation of networks, as motivated in the following chapters of this book, resilience needs to be regarded as an internal property of any networked system.

? Questions

1. Characterize possible scenarios for failures in networks.
2. Explain the term “network resilience” and discuss its meaning in the context of service level degradation in a failure scenario.
3. Discuss the importance of redundancy in increasing network reliability. Characterize possible levels of its application in networks.
4. Describe the consecutive phases of the recovery procedure for services affected by failures of network elements.
5. Explain the need for a continuous adaptation of resilience mechanisms and discuss a helpful strategy for this purpose.
6. Discuss the impact of human and organizational issues on the resilience of networked systems.
7. Characterize the benefits following the deployment of resilience mechanisms in networks.
8. Describe possible ways of reducing the cost of introducing resilience to networks.
9. Explain the challenges to the deployment of resilience mechanisms in networks.

References

1. A10: Five Most Famous DDoS Attacks and Then Some: <https://www.a10networks.com/blog/5-most-famous-ddos-attacks/>. Accessed 14 Apr 2023
2. Agarwal, P.K., Efrat, A., Ganjugunte, S.K., Hay, D., Sankararaman, S., Zussman, G.: The resilience of WDM networks to probabilistic geographical failures. *IEEE/ACM Trans. Netw.* **21**(5), 1525–1538 (2013)
3. Agiwal, M., Roy, A., Saxena, N.: Next generation 5G wireless networks: A comprehensive survey. *IEEE Commun. Surv. Tutorials* **18**(3), 1617–1655 (2016)
4. Autenrieth, A.: Recovery time analysis of differentiated resilience in MPLS. In: Proceedings of 4th International Workshop on Design of Reliable Communication Networks (DRCN’03), pp. 333–340 (2003)
5. Bonaventure, O., Filsfils, C., Francois, P.: Achieving sub-50 milliseconds recovery upon BGP peering link failures. *IEEE/ACM Trans. Netw.* **15**(5), 1123–1135 (2007)

6. Chiesa, M., Kamiński, A., Rak, J., Rétvári, G., Schmid, S.: A survey of fast-recovery mechanisms in packet-switched networks. *IEEE Commun. Surv. Tutorials* **23**(2), 1253–1301 (2021)
7. Cholda, P., Jajszczyk, A.: Recovery and its quality in multilayer networks. *IEEE/OSA J. Lightwave Technol.* **28**(4), 372–389 (2010)
8. Cholda, P., Mykkeltveit, A., Helvik, B.E., Wittner, O.J.: A survey of resilience differentiation frameworks in communication networks. *IEEE Commun. Surv. Tutorials* **9**(4), 32–55 (2007)
9. Dikbiyik, F., Tornatore, M., Mukherjee, B.: Minimizing the risk from disaster failures in optical backbone networks. *IEEE/OSA J. Lightwave Technol.* **32**(18), 3175–3183 (2014)
10. Dusia, A., Sethi, A.S.: Recent advances in fault localization in computer networks. *IEEE Commun. Surv. Tutorials* **18**(4), 3030–3051 (2016)
11. ENISA: Measurement frameworks and metrics for resilient networks and services: Technical report (2011)
12. ENISA: Methodologies for the identification of critical information infrastructure assets and services: Guidelines for charting electronic data communication networks (2014)
13. CNET: Google goes down for 5 minutes, Internet traffic drops 40%: <https://www.cnet.com/tech/services-and-software/google-goes-down-for-5-minutes-internet-traffic-drops-40/>. Accessed 14 Apr 2023
14. Gościński, R., Walkowiak, K., Klinkowski, M., Rak, J.: Protection in Elastic Optical Networks. *IEEE Netw.* **29**(6), 88–96 (2015)
15. Gouglidis, A., Green, B., Busby, J., Rouncefield, M., Hutchison, D., Schauer, S.: Threat awareness for critical infrastructures resilience. In: Proceedings of the 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM'16), pp. 196–202 (2016)
16. Habib, M.F., Tornatore, M., De Leenheer, M., Dikbiyik, F., Mukherjee, B.: Design of disaster-resilient optical datacenter networks. *IEEE/OSA J. Lightwave Technol.* **30**(16), 2563–2573 (2012)
17. Hutchison, D., Hjalmtysson, G., Sterbenz, J.P.G., Ventre, G., Vicente, J.: Would self-organized or self-managed networks lead to improved QoS? In: de Meer, H., Bhatti, N. (eds) Quality of Service—IWQoS 2005. Lecture Notes in Computer Science, vol. 3552, pp. 17–18 Springer, Berlin (2005)
18. Hutchison, D., Pezaros, D., Rak, J., Smith, P.: On the importance of resilience engineering for networked systems in a changing world. *IEEE Commun. Mag.* **61**(11), 200–206 (2023)
19. Hutchison, D., Sterbenz, J.P.G.: Architecture and design for resilient networked systems. *Comput. Commun.* **131**, 13–21 (2018)
20. Jabbar, A., Rohrer, J.P., Oberthaler, A., Cetinkaya, E.K., Frost, V., Sterbenz, J.P.G.: Performance comparison of weather disruption-tolerant cross-layer routing algorithms. In: Proceedings of the 28th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM'09), pp. 1143–1151 (2009)
21. Karagiannis, G., Altintas, O., Ekici, E., Heijenk, G., Jarupan, B., Lin, K., Weil, T.: Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards, and solutions. *IEEE Commun. Surv. Tutorials* **13**(4), 584–616 (2011)
22. Kini, S., Ramasubramanian, S., Kvalbein, A., Hansen, A.F.: Fast recovery from dual link or single-node failures in IP networks using tunneling. *IEEE/ACM Trans. Netw.* **18**(6), 1988–1999 (2010)
23. Li, F., Wang, Y.: Routing in vehicular ad hoc networks: A survey. *IEEE Veh. Technol. Mag.* **2**(2), 12–22 (2007)
24. Markopoulou, A., Iannaccone, G., Bhattacharyya, S., Chuah, C.-N., Ganjali, Y., Diot, C.: Characterization of failures in an operational IP backbone network. *IEEE/ACM Trans. Netw.* **16**(4), 749–762 (2008)
25. Mas, C., Tomkos, I., Tonguz, O.K.: Failure location algorithm for transparent optical networks. *IEEE J. Sel. Areas Commun.* **23**(8), 1508–1519 (2005)
26. Menth, M., Martin, R.: Network resilience through multi-topology routing. In: Proceedings of the 5th International Workshop on Design of Reliable Communication Networks (DRCN'05), pp. 271–277 (2005)

27. Mukherjee, B., Habib, M.F., Dikbiyik, F.: Network adaptability from disaster disruptions and cascading failures. *IEEE Commun. Mag.* **52**(5), 230–238 (2014)
28. Nelakuditi, S., Lee, S., Yu, Y., Zhang, Zh.-L., Chuah, Ch.-N.: Fast local rerouting for handling transient link failures. *IEEE/ACM Trans. Netw.* **15**(2), 359–372 (2007)
29. Rak, J., Hutchison, D. (eds.): *Guide to Disaster-Resilient Communication Networks*. Springer, Berlin (2020)
30. ResumeNet: Resilience and survivability for future networking: Framework, mechanisms and experimental evaluation (2011). <http://www.resumenet.eu/>. Accessed 18 Jan 2024
31. Reuter, C.: Power outage communications: Survey of needs, infrastructures and concepts. In: *Proceedings of 10th ISCRAM Conference*, pp. 1–5 (2013)
32. Segovia, J.S.: *Robustness against large-scale failures in communications networks*. PhD Thesis, University of Girona, Spain (2011)
33. Smith, P., Hutchison, D., Sterbenz, J.P.G., Schöller, M., Fessi, A., Karaliopoulos, M., Lac, C., Plattner, B.: Network resilience: A systematic approach. *IEEE Commun. Mag.* **49**(7), 88–97 (2011)
34. Stankiewicz, R., Chołda, P., Jajszczyk, A.: QoX: What is it really? *IEEE Commun. Mag.* **49**(4), 148–158 (2011)
35. Sterbenz, J.P.G., Hutchison, D., Cetinkaya, E.K., Jabbar, A., Rohrer, J.P., Schoeller, M., Smith, P.: Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Comput. Netw.* **54**(8), 1245–1265 (2010)
36. Vasseur, J.-P., Pickavet, M., Demeester, P.: *Network Recovery: Protection and Restoration of Optical, SONET-SDH, and MPLS*. Morgan Kaufmann, Los Altos (2004)
37. Wang, Y., Ma, Ch., Li, X., Zhao, Y., Zhang, Y.: Node protection method with content-connectivity against disaster in disaster recovery center networks. In: *Proceedings of the 13th International Conference on Optical Communications and Networks (ICOON'14)*, pp. 1–4 (2014)
38. Wilson, C.: CRS report for congress: High altitude electromagnetic pulse (HEMP) and high power microwave (HPM) devices: Threat assessments. Order Code RL32544, pp. 1–27 (2008)
39. Zhang, J., Zhu, K., Mukherjee, B.: Backup reprovisioning to remedy the effect of multiple link failures in WDM mesh networks. *IEEE J. Sel. Areas Commun.* **24**(8), 57–67 (2006)

Chapter 2

Resilience of Networked Systems: A Taxonomy of Challenges, Faults, Disciplines, and Attributes



Failures in networked systems are inevitable. As indicated in the previous chapter, they may occur due to various challenges, including forces of nature (such as hurricanes or earthquakes), human errors (e.g., cable cuts), or malicious attacks, to mention a few. Despite the visible diversity of their characteristics, they share a common feature: there is no way to eliminate them entirely.

Concerning scenarios of potential failures, qualitative measures are necessary to evaluate the performance of the affected networked systems. Also, as failures are expected to occur relatively rarely (however, often with notable negative consequences), a sufficiently long observation time horizon should be assumed for the related analysis.

A networked system can generally be defined as a system consisting of elements that work together within the constraints imposed by its architecture. Each element of the system can be considered a system itself. Services provided to users by networked systems can be broadly classified into three main groups: (a) communications, (b) storage, and (c) computations. They can be offered to users by either single system elements (for instance, concerning data storage or computations by high-performance units) or a group/chain of system elements (e.g., data transmission: switching/routing, or computations in the cloud). The related service failures can thus follow from faults affecting either single elements of a system or a subset of system elements in the context of either hardware, software, or operational (human-related) issues. Therefore, the resilience of networked systems comprises the resilience of the infrastructure (hardware), services (e.g., transmission services provided by the networking equipment), and software.

As this book primarily addresses the aspect of resilient routing, the analysis of events leading to service failures and the related resilience properties and schemes are presented and analyzed here from the perspective of communication services

of networked systems or, simply speaking, of communication networks. This also follows from the observation that communication services, apart from their default role, are also crucial for the correct functioning of storage and computation services, e.g., due to enabling sending the (input) data to storage/computation units, as well as the requested information/results back to the users.

It is true that our daily routines, becoming more and more dependent on communication services, are responsible for the exponential growth of exchanged information. Consequently, emerging failures of networked systems links (or nodes) bring about significant data and revenue losses. With the continuously observed extension of communication networks' impact on supporting almost all activities of our society, the negative consequences of failures of network elements are only expected to increase.

Link failures in communication networks can sometimes last several days/weeks (especially in wide-area networks) and, therefore, imply a remarkable degradation of the network performance over a long time. Indeed, localization of faults followed by necessary physical repair operations (e.g., of undersea optical cables) can take days to weeks, causing noticeable disruptions to network-dependent services.

The problem of link availability gets even more complicated in wireless networks due to the time dependency of link characteristics on various factors, including weather-based disruptions. However, in local area networks with wired links, the share of node failures over all failures is commonly more significant than in wide-area networks due to the possibility of providing better physical protection of shorter links.

In this chapter, we provide a taxonomy of challenges and faults and characterize the disciplines and attributes of the resilience of networked systems with a particular focus on communication services. The remaining part of this chapter is organized in the following way. To deal with faults of elements of networked systems, it is necessary to analyze first the challenges responsible for their occurrence. This is the main aim of Sect. 2.1 presenting the classification of challenges, followed by spatial and temporal analysis of their influence and analysis of their correlation with various challenge categories. However, the diversity of challenge characteristics makes the task of real-time challenge identification rather complex and often requires a multistage approach, as described later in Sect. 2.1.

It is also crucial to identify the intermediate events occurring before any service failure (i.e., faults and errors referring to network elements), which is indispensable to provide the real-time response of recovery mechanisms. In particular, Sect. 2.2 provides the taxonomy of faults as well as their correlation with the occurrence of challenges, while Sect. 2.3 characterizes the successive errors and failures, i.e., the last components of the "challenge \rightarrow fault \rightarrow error \rightarrow failure" chain.

The diversity of communication systems technologies and that of challenges triggering differentiated failure scenarios are the reasons for the existence of a multitude of resilience disciplines described in detail in Sect. 2.4 referring to network design approaches to provide service continuity (in particular including survivability, fault tolerance, traffic tolerance, and disruption tolerance mechanisms). Analysis of network resilience can be, in turn, performed using measurable

characteristics, i.e., attributes of network dependability (such as reliability and availability), security, or performability—all related to the perceived service quality and included in recommendations of the International Telecommunication Union—Telecommunication Standardization Sector (ITU-T) and Internet Engineering Task Force (IETF), as described in Sect. 2.4. The later part of the chapter highlights the techniques for evaluation and improvement of the system total availability (Sect. 2.5) and reliability (Sect. 2.6) for serial, parallel, and mixed architectures of systems. The summary and conclusions are provided in Sect. 2.7.

2.1 Challenges

Communication networks and, generally, networked systems are subject to a large group of challenges, recognition of which is crucial for network design and planning. As discussed by Çetinkaya and Sterbenz in [10], a *challenge* can be defined as a characteristic/condition that may occur as an event affecting the normal operation of a network. Major challenges for networks are shown in Fig. 2.1.

Human errors are commonly seen as non-malicious (i.e., non-intentional) activities. Examples include human activities leading to accidental failures of network elements—e.g., network link cuts by a digger or misconfiguration/construction errors (for instance of BGP routing protocol or DNS service) as a result of human incompetence [10] often responsible for the so-called *technology-related failures* [51]. However, the range of negative consequences for communication systems can be broad: from minor and short-lasting events to even catastrophic failures (for instance, a fire accidentally initiated by humans or when ignoring early warnings in system operation).

Large-scale disasters are mostly caused by forces of nature (referred to as *natural disasters*). They comprise the following *predictable disasters* resulting in significant disruptions of communication links, as well as system nodes [51]:

- Floods, including the 2013 flood in central Europe due to a rapid swelling of the Danube and Elbe rivers.
- Fires such as the 1998 Hinsdale fire in the USA [34] responsible for massive failures in the Chicago metropolitan network (about 40,000 lines served by the Hinsdale switch inactive for five weeks until a complete restoration) or the 2018 fire in Greece (Attica region) making communications in the impacted areas (also among rescue team members) significantly reduced [64].

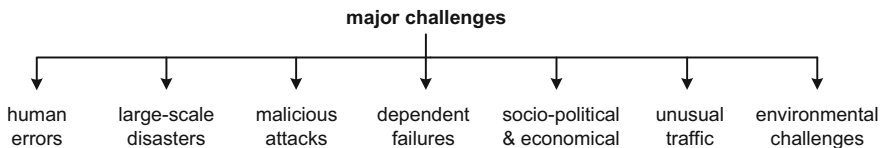


Fig. 2.1 Major challenges identified in [10]

- Heavy rainfalls, such as the rainfall in 2004 in Croatia responsible for a breakdown of the critical communication infrastructure, which, in turn, implied a failure of the Croatian flight control system lasting for several hours.
- Hurricanes and tornadoes. A notable example of these events is the 2005 hurricane Katrina in the USA, which resulted in power outages switching off multiple network nodes for over a week (limiting the overall network availability to about 85% [35]) or the 2017 hurricane Maria (Latin America, Mexican Gulf) bringing a long-lasting unavailability of Internet and cellular communications in Dominicana as well as no cellular communications in 95% of cases in Puerto Rico [49].

An essential subset of natural disasters refers to *non-predictable disasters*, among which we can mainly identify earthquakes. Their negative impact on the communication infrastructure can be severe, primarily in the context of failures of long-haul communication links (typically of undersea optical cables characterized by a long time needed for a physical repair). For instance:

- The 2006 Taiwan earthquake (a magnitude of 7.1) caused failures of seven submarine optical links, which suspended Internet connectivity between Asia and North America for weeks and disrupted international communications to China, Hong Kong, Japan, Korea, Singapore, and Taiwan [33].
- The 2008 Wenchuan earthquake triggered damage to nearly 4,000 telecom offices and a failure of about 30,000 km of optical links [50].
- The 2008 earthquake in the Mediterranean Sea area caused failures of the undersea optical links, implying long-lasting unavailability of Internet communications between Europe and Africa [63].
- The 2011 Greatest Japan earthquake of 9.0 magnitude [17, 62] was responsible for massive power outages, failures of undersea optical links, failures of over 2,000 switching offices (the results of the main shock on March 11, 2011, and the aftershock on April 7, 2011) and damaged about 1,500 cellular base stations [53].

It is worth noticing that the effects of natural disasters can sometimes be interconnected since the occurrence of certain disasters can magnify the impact of other disasters. For instance, an earthquake can trigger a tsunami, causing flooding of large areas of land and triggering failures of the inland communication infrastructure (for example, as in the case of the 2011 Greatest Japan earthquake). Similarly, a heavy wind can notably increase the negative outcome of a fire by contributing to its wider and faster spreading. Apart from terrestrial or meteorological causes, natural disasters can happen due to cosmological events, e.g., geomagnetic storms [30], the occurrence of which is also hard to predict.

The statistical information related to the vulnerability of regions to natural disasters can be found, e.g., in [45]. A general conclusion from observations over the last two decades is that the number, intensity, and scale of natural disasters are notably increasing.

Malicious attacks is another group of challenges referring to deliberate actions designed to cause significant disruption, commonly by targeting the network

infrastructure's most important software/hardware elements. Examples include, among others, the Distributed Denial of Service (DDoS) attacks aimed at exhausting resources of multiple elements of a networked system at the same time, e.g., the DDoS attack targeted at Amazon Web Services in 2020 with a duration of three days and a peak rate of malicious traffic of 2.3 Tbps, the 2020 attack oriented at Google which lasted for six months with a peak rate of 2.5 Tbps or the 2018 attack targeted at GitHub by a malicious traffic at 1.35 Tbps and lasting for about 20 minutes [1].

Dependent failures refer to challenges that may trigger a cascade of failures—for instance, after a failure of a system (or its part) providing services to another system [10]. Examples include power grids assuring power supply for the Internet infrastructure (see, e.g., the impact of Hurricane Katrina, which caused long-lasting power blackouts and failures of network nodes also due to depletion of fuel supplies of energy generators caused by a prolonged duration of a disaster event).

An important observation is that systems' dependency can often be mutual, which increases the potential scale of losses even more. For instance, a mutual dependency between communication networks and power grids (with power grids providing power supply and communication networks assuring the control functions for power grids) is a good example of *interdependent systems*. In such mutually dependent architectures, a failure of even a single element in one system can lead to a total collapse of both systems due to a cyclic propagation of failures [38].

Socio-political and economical challenges include deliberate activities (also acts of terrorism) affecting networks, even if they are not a primary target (e.g., a bombing attack harmful to a communication system located in the vicinity of the event occurrence). Results of other activities aimed at threatening the operation of communication systems directly include Internet outages triggered to achieve advantages on economic markets or as a response to political decisions (mainly performed at a national scale) [16].

Unusual traffic can be problematic if its volume exceeds the limits assumed during the network design phase. Such extra traffic can be inserted into the network, e.g., after the occurrence of a catastrophic event not necessarily disrupting the network infrastructure itself but resulting in a significant increase in the number of simultaneous requests to get information (often by an order of magnitude greater, as in the case of the 9/11 terrorist attack in New York in 2001).

Environmental challenges depend on the properties of communication environments. They are related, e.g., to mobility aspects in wireless networks, in particular to time-varying characteristics of wireless links as, for instance, in vehicular ad-hoc networks—VANETs [55] or to patterns of node mobility (e.g., in marine scenarios) where communications often need to comply with the rules of delay-tolerant networking—DTN [9]. However, environmental challenges can also be linked to adverse weather conditions, which temporarily (i.e., in a transient way) but relatively frequently affect the available capacity of wireless links (as in the case of heavy rain for radio frequency—RF links or heavy fog/dense clouds concerning the free-space optical wireless communications—FSO [29, 51]).

The crucial aspects of challenges of all types refer to characteristics measurable in space and time. As shown in Table 2.1, the impact of a disruption on

Table 2.1 Spatial and temporal characteristics of challenges summarized in [10]

Examples of challenges	Spatial region		Duration	
	Challenge	Impact	Challenge	Impact
Earthquake	100s km ²	100s km ²	Seconds	Days+
Fire	100s m ²	10s km ²	Hours	Days
Hurricane	100s km ²	100s km ²	Hours	Days+
Malicious attack	Node	Global	Hours	Hours
Misconfiguration	Node	Global	Seconds	Minutes
Pandemic	Global	Global	Days	Months
Policy-related	N/A	Regional/global	N/A	Years
Power blackout	100s km ²	Regional	Minutes	Hours
Solar storm	1000s km ²	1000s km ²	Minutes	Days+
Terrorism	100s m ²	Global	Hours	Hours+

networked system performance can often be significantly different from the original scope/duration of a challenge. For instance, an attack targeted at a single node may influence the entire system’s performance. Similarly, a challenge expected to last for a short time (e.g., a solar storm occurring within several minutes) can trigger consequences experienced in a much longer period of several days.

According to [5, 10], any challenge can be further described based on additional criteria summarized in Fig. 2.2. Among them, we can identify their cause (either natural, human-made, or challenge-dependent). Another aspect refers to the challenge boundaries (being either internal—referring to challenges identified inside the considered system such as, e.g., misconfiguration issues or external—where the source of a challenge is located outside the system—as in scenarios of natural disasters or attackers operating from outside the system). The “target” property of a challenge can be considered as denoting either a potential direct victim of a challenge (e.g., in the case of malicious software) or collateral (a system not targeted directly by a challenge—for instance, in a scenario of a terrorist attack).

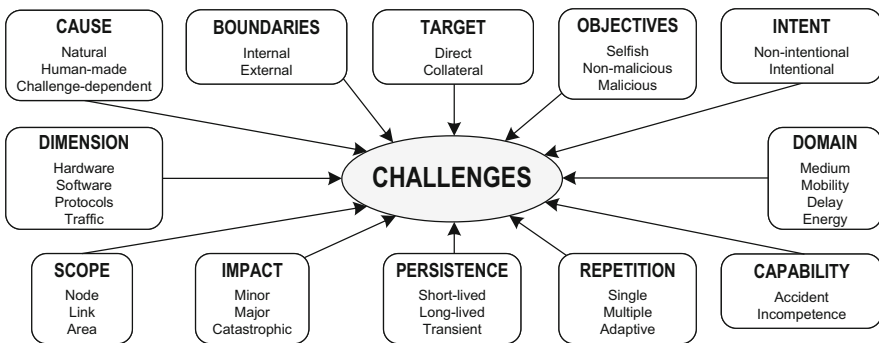


Fig. 2.2 A detailed classification of characteristics of challenges based on [10]

Based on the objectives, challenges can be divided into selfish (referring to the behavior of a component of a system), non-malicious, or malicious. Human operations can be classified in terms of their intent into non-intentional (non-deliberate) or intentional (deliberate). The dimension of a challenge defines the aspect of a networked system architecture it refers to, being either hardware, software, protocols, or referring to traffic. Domain, in turn, denotes the aspect of a system operation potentially threatened by a challenge (either medium, mobility, delay, or energy).

Further characteristics of challenges define their scope (either particular nodes/links of a system or a specific area, for instance, in the case of natural disasters affecting system elements located in a given region), impact (minor, major, or catastrophic), persistence characterizing duration (short-lived, long-lived, or transient), as well as repetition (single, multiple, or adaptive—able to adjust their properties in a sequence of events, as in a sequence of attacks). Finally, the “capability” property of a challenge is another feature related to the activities of humans, which divides human-related challenges into accident- or incompetence-related.

A detailed correlation of these challenge categories with major challenges listed earlier in this chapter in Fig. 2.1 (following from the respective one proposed in [5] for computer systems by the International Federation for Information Processing (IFIP) Working Group 10.4) can be found in [10]. Recognition of challenges often requires a multistage approach illustrated in Fig. 2.3 [19].

It includes detection of challenge symptoms (i.e., which may lead to the recognition of a challenge onset), identification of the root cause of a challenge, and determination of a potential impact on the system. However, to be cost-efficient, any remediation action should be preceded by assessing the challenge impact versus the cost of remediation [19]. Challenge detection mechanisms, typically invoked in a distributed manner, should be as lightweight as possible in order not to use resources unnecessarily (which is an essential requirement for resource-limited networks) and not to disturb the normal operation of a networked system [19].

Mitigating the potential impacts of challenges in real time is difficult, especially when they share several symptoms. For instance, the observed increased traffic can be an implication of a Distributed Denial of Service attack attempt or simply the legitimate overload caused by flash crowds.

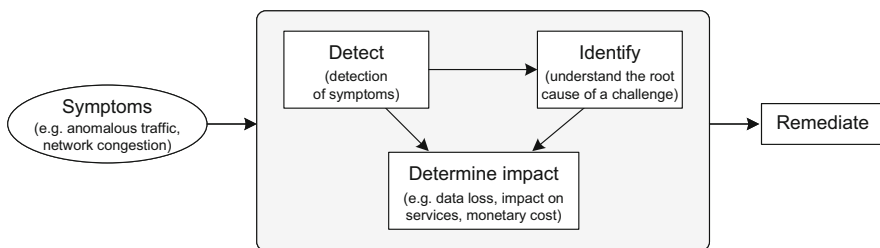


Fig. 2.3 Aspects of challenge identification from [19]

2.2 Faults

Faults can be broadly defined as flaws (imperfections) in the system that are likely to arise at various stages of the design, implementation, and maintenance of system elements and services. They may appear as potential obstacles/disturbances to the proper operation of the system and its services [59]. Examples include, e.g., software bugs, hardware flaws (generally, system architecture/element problems), or operational flaws such as physical damages of nodes (e.g., due to a fire) or damages of given ports of a switch due to an electromagnetic pulse (EMP). Faults can manifest themselves in either unprotected or protected systems. Their activation can also take place in the normal operational state of a system and even without the occurrence of challenges.

Similar to challenges, faults can be broadly classified based on their nature, component, origin, extent, and persistence. A taxonomy of faults referring to their specifics extending the one from [36] is provided in Fig. 2.4.

In particular, concerning the nature of faults, we can identify either *intentional faults* (being results of deliberate activities being either malicious or non-malicious) or *accidental faults*, which were created (or appeared) fortuitously.

The component aspect divides the set of faults into *software faults*, i.e., flaws in the design or development of software, commonly termed “bugs” such as *Bohrbugs* (hard faults, easily detected), *Mandelbugs* (characterized by complex underlying causes, chaotic and even nondeterministic behavior), *Heisenbugs* (elusive faults whose behavior often alters while being researched) [21], or *aging-related bugs* (software faults that get accumulated over time), as well as *hardware faults* referring to the hardware elements of a system.

The origin of faults covers three aspects:

- Creation/occurrence phase referring to either *development faults* (flaws that arose during the phases of system design, deployment, and modification as well as when defining the respective procedures for operating the system) or *operational faults* related to the exploitation phase of a system
- Phenomenological causes referring to the matter triggering faults being either *human-made faults* (due to human imperfections) or *physical faults* (due to unfavorable physical phenomena)

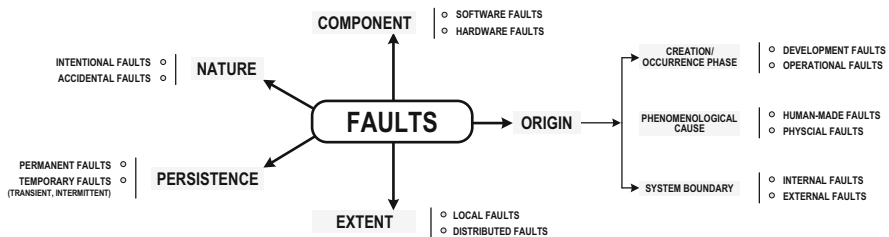


Fig. 2.4 A taxonomy of faults

- System boundary defining either *internal faults*—referring to parts of the system state leading to errors when invoked by computational activities or *external faults* originating from interactions with the physical or human environment of a networked system

The extent of faults representing the coverage of faults refers to either *local faults* (e.g., faults at a given location of a system) or *distributed faults* spread across multiple locations.

The attribute of persistence of faults, referring to the fault lifetime, comprises three variants. The first one refers to *permanent faults* (whose presence is not limited in time to certain internal/external conditions, and often requiring physical repair actions). The second variant denotes *temporary faults* (the ones that occur during a certain limited period and can be terminated/cleared without any interrupting operation). It can be further decomposed into *transient faults* occurring due to specific time-varying properties of the physical (external) environment that tend to subside when the factor affecting the network element ceases to exist and *intermittent faults*—a subset of temporary internal faults due to malfunctioning of devices following, e.g., from changes of parameter values of hardware components such as their temperature.

It is worth noting that any fault related to the design and deployment of a networked system (as well as the implementation of the related software) is commonly considered human-made, as humans naturally deploy such systems.

As discussed earlier in this chapter, challenges can be responsible for creating or activating faults. This is illustrated in Table 2.2, presenting a correlation between the major types of challenges provided in Sect. 2.1 and the identified classes of faults.

As given in Table 2.2, faults implied by human errors can be considered either intentional (e.g., not eliminated due to the cost constraints of system deployment) or accidental (for instance, as a result of incompetence). Human errors may lead to both software faults (e.g., bugs) and hardware faults (e.g., cuts of communication links by a third party); they can also be related to either the development or the operational phase (as both phases include the involvement of humans). They can thus result in human-made faults and can lead to both internal faults (e.g., caused by programming issues) and external faults (due to actions by humans performed during the exploitation of a system). The scale of potential faults triggered by these challenges ranges from local faults to distributed faults. Such faults can also be permanent (for instance, the already mentioned cable cut by a digger operated by a human).

Faults implied by the occurrence of large-scale disasters are obviously accidental and affect the hardware part of the system. They can manifest themselves during the operation of a system and can lead to physical faults being external concerning their source, both local (single/regional faults) and distributed regarding their extent, as well as permanent in the context of their adverse effects for elements of a networked system.

Malicious attacks can trigger intentional faults that can affect the software and hardware components during the operation of a system and its services. They are

Table 2.2 Correlation between challenges and faults

	Human errors	Large-scale disasters	Malicious attacks	Dependent failures	Socio-political & economical	Unusual traffic	Environmental challenges
Intentional faults	+		+		+	+	
Accidental faults	+	+		+		+	+
Software faults	+		+		+		
Hardware faults	+	+	+	+	+		
Development faults	+					+	
Operational faults	+	+	+	+	+	+	+
Human-made faults	+		+		+	+	
Physical faults		+		+			+
Internal faults	+		+		+		
External faults	+	+	+	+	+	+	+
Local faults	+	+	+	+	+	+	+
Distributed faults	+	+	+	+	+	+	+
Permanent faults	+	+	+	+	+		
Temporary faults			+	+	+	+	+

made by humans (attackers). It is worth noting that malicious attacks can result in internal faults, e.g., by injecting a malicious code—often referred to as “Trojan horses” or external (e.g., due to intrusion) [36]. The extent of faults caused by attacks can range from local (e.g., injection of a malicious code at a single node) to distributed. Similarly, the effect of malicious attacks on the persistence of faults can be considered either permanent (e.g., if an attack causes physical damage to the system element) or temporal (ending with the termination of the attack).

Dependent failures are typically responsible for accidental faults of hardware in the context of the exploitation period (e.g., a failure of a power grid switching off system nodes—an example of an operational fault). The phenomenological causes of the related faults are mainly physical (e.g., a power cut). Since dependent failures occur in scenarios of interaction of multiple systems, the respective faults are typically external for each considered system. Concerning the coverage area, the triggered faults can be either local or distributed, as well as either permanent (e.g., physical destruction of a system element), or temporal (for instance, determined by the end of the power-cut period).

Since socio-political and economical challenges often share similar motivations with malicious attacks, the characteristics of faults they trigger are commonly comparable.

The unusual traffic (i.e., traffic exceeding the expected volume) can visibly reduce the ability of certain elements of a networked system to fulfill their mission. This, in turn, can activate faults when the system is not prepared properly to handle high traffic volumes. This feature may be not implemented in the system either accidentally (an accidental fault) or by intentionally limiting the deployment costs of a system (an intentional fault). Although the occurrence of unusual traffic is not expected to trigger either software or hardware faults, it can indeed be responsible for operational faults in the system due to its inability to serve the excessive traffic based on certain assumptions made during the development of a system (a development fault). Faults triggered by the unusual traffic are naturally human-made and can be considered external since their source is indeed the people who deploy the system as well as the people using it. The unusual traffic can even result in the unavailability either of certain nodes/links in a given location (a local fault) or of elements distributed across the system (a distributed fault) due to a time-varying volume of traffic (a temporal fault).

Finally, environmental challenges can lead to accidental faults (see, e.g., adverse weather conditions limiting the effective capacity of wireless links even to zero). They can trigger faults during the exploitation of a system (operational faults), which are typically physical (following from the time-varying properties of the environment), and happen due to the interaction of a system with the environment (external faults) in a given area (local or distributed faults). The changing properties of the environment are also responsible for the temporal characteristics of the related faults.

A fault needs to be detected in real time in either the physical layer (for example, due to loss of signal, loss of modulation, or loss of clock) using signal degradation recognition (e.g., increased bit error rate—BER) or quality of service deterioration

(indicated by decreased throughput or increased transmission delay). After fault detection, it is essential to localize the point of fault to distribute fault notifications necessary to remediate the adverse effects of the fault on the network performance [12, 19]. A complete return of a system to its normal operational state can be achieved later only if the root causes of the fault are eliminated.

For any challenge, apart from evaluating its *impact* on the networked system performance, it is essential to identify the probability of a challenge occurrence (*challenge_prob*), as well as the likelihood (*fail_prob*) that a particular challenge will result in a fault (since not all challenges necessarily lead to faults). Combined with information on the challenge impact, these two measures can be used to derive the measure of system resources *exposure to disruptions* from [56], as given in Eq. 2.1.

$$\text{exposure} = (\text{challenge_prob} \cdot \text{fail_prob}) \cdot \text{impact} \quad (2.1)$$

2.3 Errors and Failures

A fault, if not properly dealt with, can lead to an *error*, defined as a deviation of the observed value/state from its specified (correct) value/state [59]. An example of an error could be, e.g., for software—an improper variable value in a given time, e.g., a pointer variable directing to a wrong element in the memory space. Errors, in turn, are considered to be the cause of *service failures* or shortly *failures* [7, 36, 57, 59, 60], i.e., events occurring when the delivered service deviates from correct service.

As justified by Laprie in [36], whether an error will lead to a situation considered a failure state or not depends on three aspects:

- The activity of a system (since an error can be overwritten before causing a damage or being noticed as a failure)
- The meaning of a failure from the perspective of particular users
- The presence of redundant components in the system architecture

Following [36], failures can be further characterized based on three significant aspects: (1) consequences for the environment, (2) domain, and (3) user perception. Concerning the potential consequences for the environment, we can distinguish either *benign failures*, if these consequences are similar in scale to the benefit from the correct functioning of service, or *catastrophic failures* when failure losses are visibly more significant than the mentioned benefit from proper provisioning of service.

The domain of a failure determines whether a given event implies either a *value failure* (when the service value deviates from the specification) or a *timing failure* (denoting services delivered not in time—either too early, too late, or not delivered at all).

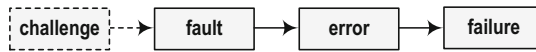


Fig. 2.5 The chain of events leading to failures of services in a networked system [10, 59]

Failure perception divides failures into *consistent failures*—i.e., events considered failures by all users or *inconsistent failures* if such events are not considered failures by all users.

The four events discussed above form the chain of events potentially leading to failures of services, as identified in [10] and illustrated in Fig. 2.5. However, as not all faults are triggered by challenges, the “challenge” box and the related arrow in Fig. 2.5 are marked with dashed lines.

Failures of links/nodes in a networked system often imply severe disruptions to service provisioning. Examples include the aspect of routing of demands in communication systems. The resulting problem of communication path unavailability can be additionally escalated owing to the observed exponential increase in the volume of transmitted information worldwide. Since failures of communication paths are inevitable simply due to the inability to prevent a significant subset of challenges, appropriate modifications, at least to routing schemes, are needed to make end-to-end communications feasible also in failure scenarios.

2.4 Resilience Disciplines

Several definitions for resilience disciplines have been proposed for networked systems in the literature (see, e.g., [36, 37, 39]). The most comprehensive definition of *network resilience* seems to be by Sterbenz et al. from [59]. Following [58, 59], it can be defined as the ability of a network to provide and maintain an acceptable level of service in the face of various faults and challenges to normal operation.

Since faults and challenges are inevitable, network resilience should be viewed as one of the most essential characteristics of the design of any networked system.

Figure 2.6 gives a detailed classification of resilience disciplines. According to [59], resilience disciplines can be classified into two main categories, namely: *challenge tolerance* focusing on network design approaches to provide service continuity in the presence of challenges and *trustworthiness*—describing measurable characteristics of analyzed communication systems. The relation between these two, referred to as *robustness*, is the indicator of the performance of a network under perturbative conditions.

The first of the two considered resilience disciplines can be further decomposed into *survivability* (including *fault tolerance*)—referring to the communications infrastructure of networked systems, *disruption tolerance* for resistance of communication paths to disruptions, and *traffic tolerance* for various challenges related to traffic (e.g., additional volume that is injected into the network).

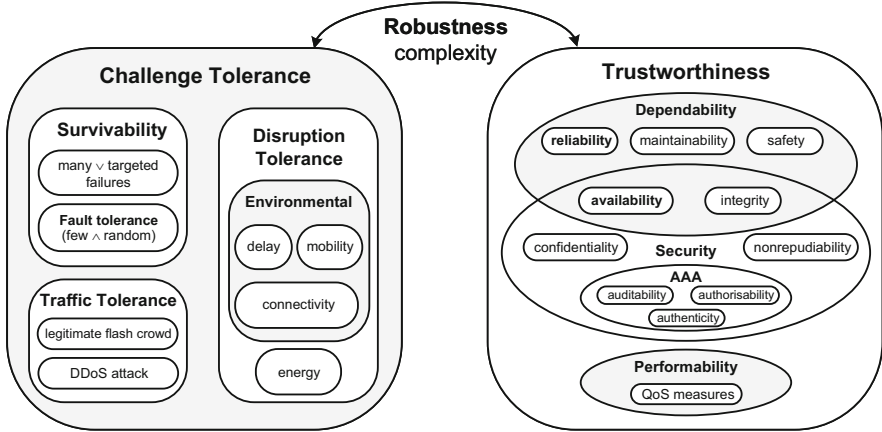


Fig. 2.6 Classification of resilience disciplines from [59]

2.4.1 Survivability and Fault Tolerance

Survivability is typically defined as the capability of a network to fulfill its mission in a timely manner in the presence of threats, including attacks or natural disasters [59]. Another definition from [22] relates survivability with the ability of a network to recover the affected traffic in failure environments and to provide different services continuously. In [32], survivability is, in turn, defined as the ability of a network to continue the service in the presence of failures, while in [11], it is referred to as the ability to automatically react to both physical and software faults by redirecting the traffic from the affected paths to ones which are operating properly.

An important aspect of survivability is *fault tolerance*, referring to the ability of a networked system to cope with faults [60]. Because it is impossible to make a networked system completely fault-free, the fault tolerance strategy focuses on implementing alternative mechanisms for maintaining the system functions in a failure scenario. Therefore, it uses *redundancy* to compensate for random and uncorrelated failures of system elements. As shown in Fig. 2.7, redundancy can take

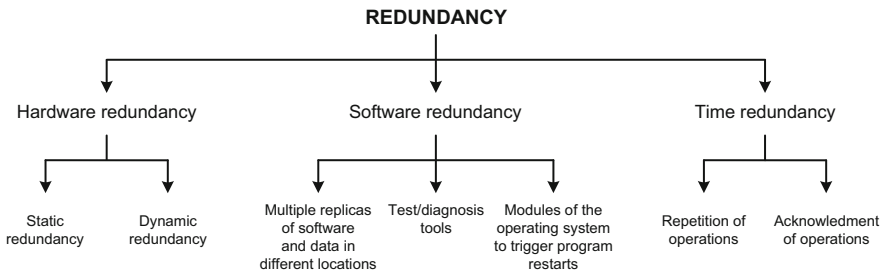


Fig. 2.7 A taxonomy of redundancy strategies based on [4]

three primary forms, namely: hardware redundancy, software redundancy, and time redundancy [4].

Hardware redundancy denotes using additional hardware components to provide fault tolerance. It can be applied, e.g., by replicating (i.e., providing redundant copies of) specific electronic components inside a given hardware module (which are powered and connected) to hide the occurrence of hardware failures within that module (often referred to as *static hardware redundancy*). The effects of faults are indeed not expected to appear in the outputs of such modules as long as the failures will not affect the replicated components at the same time. Otherwise, if a given module manifests its failure at its outputs, the use of *dynamic hardware redundancy* may be needed to bypass the failed component, as, e.g., in the case of a failure of a communication network node triggering a recovery procedure to activate the respective detours for the affected network traffic over a failed node.

Software redundancy, in turn, refers to the use of additional instructions, segments of a program, or even additional programs to take over the role of the main software (i.e., providing recovery in scenarios of software-related failures) or to detect software failures. Following [4], software redundancy can be typically applied in the form of either multiple replicas of software and storage kept in different locations of a system, the related test/diagnosis software tools, or by activities of the operating system triggering the restart of applications.

Time redundancy refers to executing additional operations meant to repeat or acknowledge a correct execution of former operations. Such actions performed at differentiated levels (ranging from single instructions to even entire programs) can indeed be helpful in the fast detection of faults (especially concerning a fine-granularity scale of operations) or recovery (e.g., when restarting entire programs).

Fault tolerance is often insufficient for recovery of a networked system after multiple correlated failures; therefore, it is considered a subset of survivability (see Fig. 2.6). The scope of survivability is thus broader than that of fault tolerance and also comprises issues of correlated failures for unbounded networks [44], including, e.g., failures due to malicious human activities (attacks) [15] or failures of large parts of a communication network infrastructure [2, 46]. Compared to fault tolerance, apart from redundancy needed to provide service recovery, survivability additionally requires *diversity* [42, 58] assuring that the same flaw does not affect multiple elements of a communication system under multiple correlated failures.

It is important to note that, apart from techniques designed for fault tolerance, reduction of the negative effects of faults can be achieved by deploying the strategies of fault prevention in particular referring to *fault avoidance* denoting activities leading to specify, verify, and derive the fault-free software (and hardware) and *fault removal* activities focused on removal of faults from existing software products [40]. In particular, fault avoidance denotes efforts to prevent faults from being incorporated into the system (e.g., by selecting well-tested components for use in the considered system in the phases of its design and deployment). Fault removal, in turn, refers to operations during the testing and maintenance phases aimed at disclosing and eliminating the identified faults.

To summarize, the quantification of network survivability is more complex than that of fault tolerance. One possible way to address simultaneous failures is to utilize multidimensional Markov chains [23]. Also, a network survivability function (i.e., a probability function of the percentage of total flow delivered after a failure) and survivability attributes proposed in [43] can be used to evaluate the survivability of a networked system.

2.4.2 *Disruption Tolerance*

Disruption tolerance is described in [59] as the ability of a network to tolerate disruptions in connectivity among its components. Connectivity is often evaluated in terms of communication path characteristics, and it may be affected due to environmental challenges, e.g., weak and episodic channel connectivity, nodes mobility, unpredictably long delay, and energy/power challenges [31].

Disruptions of end-to-end connectivity may arise particularly due to:

- A dynamic behavior of a network driven by the mobility of its nodes. A good example here is the architecture of vehicular ad hoc networks with a large number of mobile nodes (vehicles) communicating wirelessly with the stationary infrastructure nodes or directly with other vehicles [55]. Due to the movement of vehicles, the related length of wireless links continually changes over time, implying notable changes in link characteristics and frequent disconnections (i.e., a short lifetime of links).
- Long transmission delays not tolerated by network protocols [9], e.g., characteristic to satellite links. Another notable example refers to non-satellite wireless communications in the marine environment where vessels participate in a multi-hop transmission to exchange the environment-related data in the store-and-forward mode implementing the delay-tolerant networking concept—a scheme suitable for sparse network topologies with long-lasting disconnectivity problems [61].
- Energy constraints limiting the operational time of network nodes (see, for example, Wireless Sensor Networks formed by small and low-cost sensors, e.g., movement sensors, proximity sensors, or temperature sensors deployed for monitoring the properties of the environment and often used within an IoT system) powered only by batteries due to their typical outdoor location where even battery replacement itself is hardly possible (infeasible or costly) [41, 65]. Such challenges can thus lead to failures of system nodes and the related incident links.

2.4.3 *Traffic Tolerance*

Traffic tolerance is the last fundamental discipline of challenge tolerance, and, following [59], it refers to the ability of a network to tolerate the unpredictable

traffic load. Traffic can be considered a challenge if its volume rises unexpectedly far beyond the network design assumptions for the normal operational state. Example scenarios include either legitimate activities such as flash crowd [18] following natural disasters like earthquakes, implying the need to get the relevant information [28], a recent increase of network traffic following the COVID-19 pandemic period (remote work), or, e.g., malicious activities like DDoS attacks [20, 66].

Mitigating the negative effects of the increased network traffic is more straightforward to carry out in the case of legitimate human activities because their scale is easier to estimate. Then, it is commonly sufficient to increase the nominal capacity of network links together with the switching capacity of network nodes. Such updates turned out to be sufficient, e.g., during the COVID-19 period [52]. As the amount of traffic generated by malicious activities is, in turn, much more rapid as well as challenging to estimate, increasing the computing power of nodes and capacity of links can be effective only when the volume of malicious traffic turns out not to exceed the capabilities of the updated system.

2.4.4 *Trustworthiness and Its Attributes*

Trustworthiness is defined in terms of measurable service delivery characteristics as the assurance that the communication system will perform as expected [6]. It comprises three disciplines, namely dependability, security, and performability.

Dependability was defined by Laprie as a property of a system such that reliance can justifiably be placed on the service it delivers [36]. It is characterized by several attributes, including *availability* and *reliability* [7] expected to be perceived by users. It thus should reflect the ability of a networked system to deliver services to users under stated conditions in a specified period.

Table 2.3 presents a set of the essential features of resilience next used to determine the values of resilience attributes considered in this section. As discussed in [12], these features can be divided based on their aspect into two groups: *service continuity* and *service downtime*. The first subset includes parameters such as MTTF/MTFF and MUT referring to the period when a considered service is not interrupted by any failure, while the second one comprises such parameters as MTTR/MDT focused on measuring the time when the service is inaccessible. The relation between the considered resilience features is provided in Fig. 2.8.

For any service to be reliable, the mean downtime (MDT) should be much shorter than the mean uptime (MUT) (i.e., $MDT \ll MUT$). The same expectation refers to the corresponding MTTR and MTBF parameters (i.e., $MTTR \ll MTBF$).

For system elements, MTBF values are determined by the equipment vendors [3]. The value of MTBF claimed for an element by its manufacturer is often 100,000 hours. MTTR is, in turn, commonly defined arbitrarily as a unified time—e.g., 4- or 8-hour time of assistance (referring to repair operations) purchased by the customer from the vendor. Therefore, the manufacturers should strive not only to ensure a sufficiently long time of failure-free operation of their elements (for

Table 2.3 Major features of resilience

Parameter name	Aspect	Description
Mean Time to (First) Failure (MTTF/MTFF)	Service continuity	The length of a period between a point when the service was initiated until its failure (for a system element, the period between time t when the element was put into operation until its (first) failure)
Mean Up Time (MUT)	Service continuity	The mean time between a successful restoration of a service and the time of the occurrence of the next service failure
Mean Time Between Failures (MTBF)	Service continuity/downtime	The mean time between the beginning of two consecutive failures of a service
Mean Time Between Maintenance (MTBM)	Service continuity/downtime	The mean time between the beginning of two consecutive periods of (scheduled) preventive maintenance activities
Mean Time to Recovery (MTTR)	Service downtime	The mean value of the length of a period between the occurrence of a failure and the successful completion of a recovery action. The mean time spent purely on repair operations (excluding the time between the occurrence of a failure and the beginning of a repair period) is often called the <i>mean time to repair</i>
Mean Down Time (MDT)	Service downtime	The mean time of service inaccessibility

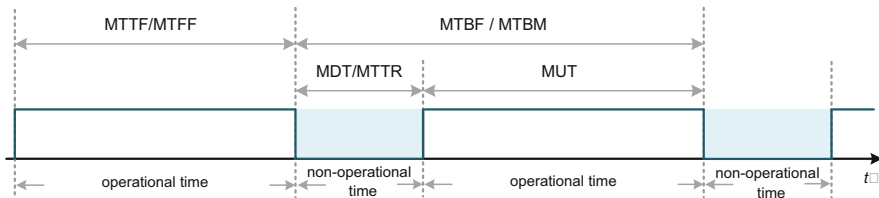


Fig. 2.8 A timeline illustrating the meaning of resilience features

example, by utilizing components characterized by increased reliability) but also seek to provide a short repair time for failed elements (for instance, by simplifying the structure of system elements during their design phase). It is also important to note that the manufacturers determine the average values of MTBF and MTTR parameters based on observations in a dedicated test environment, which may not fully reflect real-world conditions.

Another parameter adequate for resilience evaluation is the *failure rate* λ defined as the quotient of the number of failures in the considered period to the length of this period. Since MTBF has the opposite meaning, the relation between MTBF and λ can be defined as given in Eq. 2.2. This relation can be helpful in the evaluation of

system resilience, in particular when λ can be assumed to remain constant over the analyzed time.

$$\text{MTBF} = \frac{1}{\lambda} \quad (2.2)$$

It is also worth noting that concerning the communication services, dependability parameters can be controlled by the operators, e.g., by a selection of a highly reliable routing path (to assure higher service continuity) or by involving mechanisms of fast restoration of communication paths affected by failures of network elements (investigated in the next chapter of this book).

As discussed earlier in this section, the two major attributes of dependability are availability and reliability. Following [27], **availability** (A) of a system at time t can be defined as the readiness for its usage at time t , as given in Eq. 2.3.

$$A(t) = \sum_{i \in W} P_i(t) \quad (2.3)$$

where:

W is the set of states in which the system is operating correctly;

$P_i(t)$ is the probability that a system is in state i at time t .

Similarly, we can define the availability of a single system element or its part (a subsystem) providing a given service.

Concerning the networked systems expected to provide computational, storage, and communication services, the availability of a networked system defines the probability of a system to deliver its services to users at time t . Therefore, availability can be equivalently represented as the fraction of users receiving full services at time t [54].

Following [26], it can be obtained as the availability indicator from Eq. 2.4.

$$A = \frac{\text{MUT}}{\text{MUT} + \text{MDT}} \quad (2.4)$$

This formula for availability is often used as it reflects the general meaning of the availability coefficient, i.e., to show how frequently the system is operating correctly and how long it takes to bring the system back to a normal operational state after its unavailability.

Another variant of a simplified formula for availability comes from [59] and is given by Eq. 2.5.

$$A = \frac{\text{MTTF}}{\text{MTBF}} = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} \quad (2.5)$$

It is worth noting that MTTR is sometimes considered a synonym for MDT. However, MDT can often be longer than MTTR, as an additional delay may appear before starting the repair procedure. MDT may also refer to periods of scheduled

preventive maintenance activities that are, in fact, not triggered by failures, and it can also include additional delays related to providing proper personnel and spare parts, etc. For similar reasons, MTTF does not have precisely the same meaning as MUT, as MUT is not only impacted by physical failures.

The availability of a networked system naturally depends on the availability of its elements (in particular, determined by their failure rate λ and their repairability represented by the repair rate μ). Other factors affecting the system's total availability include system topology and deployed strategies of survivability and redundancy [24].

The *unavailability* (U) of an element/service is a complement for its availability (A) given by Eq. 2.6.

$$U = 1 - A = 1 - \frac{\text{MUT}}{\text{MUT} + \text{MDT}} = \frac{\text{MDT}}{\text{MUT} + \text{MDT}} \quad (2.6)$$

Availability often appears as a component of the contract between the user and the network operator, e.g., in the context of the availability of services (e.g., Internet access). Since network operators can mainly control the recovery parameters, adjusting service availability characteristics to customers' needs can be done by controlling primarily the MTTR values. Also, it is worth noting that these agreements tend not to comprise scenarios entirely out of network operators' control, such as natural disasters or failures triggered by other external events.

Table 2.4 provides information about the maximum total time of unavailability of a system element/service for different ratios of availability (A), typically denoted by the number of nines. For instance, the "five nines" availability (commonly considered a strict requirement [25]) implies a total unavailability time of only 5.26 minutes per year.

It is important to note that verifying the availability ratios in practice requires a relatively long observation time. Also, the definition of availability does not impose any constraints on how it is calculated. A certain level of availability can refer, e.g., to only one outage during a year, or imply regular short periods of unavailability every day. However, even if denoting the same availability ratio, the latter is much more irritating to the users.

The availability ratio for high-level equipment is generally expected to be well over 90%. Another essential aspect following real-life observations is that the increase of the level of system availability by one (i.e., adding one more "9") typically implies a twofold rise in the overall system cost, and, at the same time, it increases the system availability ten times.

Table 2.4 Maximum allowed total time of unavailability of a system element/service per year according to its availability ratio

Availability (A)	0.9	0.99	0.999	0.9999	0.99999	0.999999
Maximum yearly time of unavailability	36.5 days	3.65 days	8.76 hours	52.56 min	5.26 min	0.53 min

Reliability is a measure of *service continuity* and can be defined as provided in [48] as the probability that a system element/service remains operable (i.e., performs at least at a satisfactory level) in a given time interval $(0, t)$ as expressed by Eq. 2.7. As the exact formula for $R(t)$ cannot be determined in advance, it is necessary to use its approximate form.

$$R(t) = \text{Pr}(\text{no failure in } [0, t]) \tag{2.7}$$

Assuming the constant failure rate over time, i.e., $(\lambda = \lambda_{\text{const}})$ during the entire lifetime of system elements, the reliability of hardware elements and software is typically modeled using the exponential distribution provided by Eq. 2.8.

$$R(t) = e^{-\lambda t} \tag{2.8}$$

In particular, the reliability function $R(t)$ of a system element can be modeled by a negative exponential distribution of failure times [11] given by Eq. 2.9.

$$R(t) = e^{-\frac{t}{\text{MTTF}}} \tag{2.9}$$

However, as the failure rate is, in fact, time-dependent (i.e., $\lambda = \lambda(t)$), the considered exponential reliability function is generalized as follows:

$$R(t) = e^{-\int_0^t \lambda(\tau) d\tau} \tag{2.10}$$

In practice, for hardware elements, during their early-use time, $\lambda(t)$ happens to decrease in time; next, it takes an almost constant low value during the main period of the hardware element operation and tends to increase at the end of the element lifetime due to the “aging” problem of its physical components [25], as depicted in Fig. 2.9. Therefore, to reflect the time-varying nature of the element failure rate, the respective λ coefficient should be defined as given in Eq. 2.11.

$$\lambda(t) = \lambda_e(t) + \lambda_m(t) + \lambda_a(t) \tag{2.11}$$

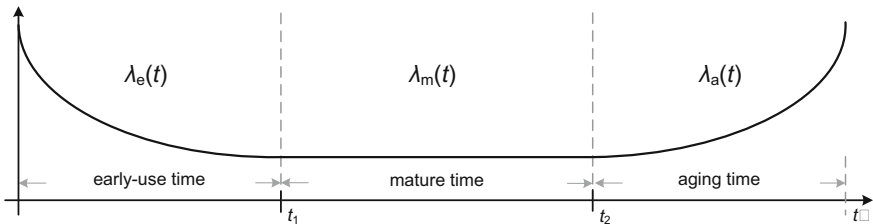


Fig. 2.9 An example function of failure rate for a networked system element in time

where:

$$\lambda_e(t) = \lambda(t) \cdot \chi_{[0, t_1]}(t);$$

$$\lambda_m(t) = \lambda(t) \cdot \chi_{[t_1, t_2]}(t);$$

$$\lambda_a(t) = \lambda(t) \cdot \chi_{[t_2, \infty]}(t),$$

while $\chi_{[a, b]}(t)$ is a common characteristic function defined as

$$\chi_{[a, b]}(t) = \begin{cases} 1, & t \in [a, b] \\ 0, & \text{otherwise} \end{cases} \quad (2.12)$$

The reliability exponential function $R(t)$ should be then specified as in Eq. 2.13.

$$\begin{aligned} R(t) &= e^{-\int_0^t [\lambda_e(\tau) + \lambda_m(\tau) + \lambda_a(\tau)] d\tau} = e^{-\int_0^t \lambda_e(\tau) d\tau} \cdot e^{-\int_0^t \lambda_m(\tau) d\tau} \cdot e^{-\int_0^t \lambda_a(\tau) d\tau} \\ &= R_e(t) \cdot R_m(t) \cdot R_a(t) \end{aligned} \quad (2.13)$$

Similar behavior of λ over time can be observed for software components of a networked system, especially in terms of a decreasing rate of failures at the early time of software use (due to the elimination of bugs at the beginning of the software use period) as well as a relatively constant low failure rate after overcoming the early-life problems. However, unlike hardware elements, software typically does not encounter an increased failure rate during the late period of its utilization.

! Availability vs. Reliability

Reliability is of utmost importance for applications that are session/connection-oriented, requiring a relatively long value of MTTF. Availability is, in turn, an appropriate measure for transactional services (e.g., Hypertext Transfer Protocol—HTTP) performing individual operations in a short time. For such services, as long as MTTR is relatively short, it is less important whether the system fails frequently or not. Availability is also typically used to assess the resilience of communication networks for practical reasons [12].

Since the reliability function for system elements provides information on the probability of their uninterrupted functioning from the beginning of the observation until a given time t , it is sometimes considered (similar to the MTTF parameter) as an attribute associated with non-repairable elements, as opposed to the availability attribute (as well as MTBF and MTTR parameters) often linked with repairable system elements.

A joint feature of high-quality system elements/services is that the more reliable/available they are expected to be, the harder it is to obtain the measurement data necessary to prove their reliability/availability.

Other dependability characteristics include:

- *Maintainability*, i.e., predisposition of a system to updates/evolution.
- *Safety*—a measure of system’s dependability under catastrophic failures, i.e., referring to the effect rather than the cause of a failure, as in the context of cyberattacks [36, 47, 59]. Any system is commonly considered to be safe if it is harmless for normal functioning of the environment.
- *Integrity* being the absence of improper (unauthorized) system alterations [7].

Another essential aspect is *security*, which is the ability of a system to protect itself from various unauthorized activities (e.g., access or updates based on the respective security policies). As discussed in [7], security has joint properties of availability and integrity with dependability, as well as individual characteristics, including:

- *Authenticity* being a property ensuring that communications come from a trusted source.
- *Authorizability* defined as the assurance that the considered elements of a system are accessed according to granted permissions.
- *Auditability* related to the assessment of whether the communication system is safeguarding information, maintaining data integrity, as well as operating in a way to achieve the goals/objectives of the organization.
- *Confidentiality* focusing on assurance of not disclosing information without proper authorization.
- *Nonrepudiability* being the assurance provided by a neutral third party that a given transaction/event did (or did not) occur.

Performability is a discipline that is used to provide measures on the performance of a system compared to the respective quality of service requirements followed from service specifications in terms of delay, jitter, throughput/goodput, and packet delivery ratio [59].

Figures 2.10 and 2.11 present selected resilience characteristics identified by ITU-T and IETF for communication networks. From the client’s perspective, the most crucial resilience characteristics are those related to the perceived service quality, sometimes referred to as the *quality of resilience* (QoR) features, being the QoS characteristics related to the resilience observed by the end users [12].

On the contrary, network operators are mainly interested in characteristics concerning the operational and implementation aspects (known as the operation-related features) influencing the cost of solutions. Since the objectives of these two groups are obviously in contrast to each other, a detailed assessment is necessary to verify whether the offered quality meets the client’s requirements and if, at the same time, it is profitable for the network operator.

There is a remarkable difference between QoS and resilience characteristics concerning the time needed to obtain the results. Unlike QoS features being short term by nature, most resilience measures are long-term [13]. Therefore, the resilience of communication networks can be evaluated in the long term only based

Recommendations of International Telecommunication Union – Telecommunication Standardization Sector (ITU-T)		
ID	Area	Metric
E.800 E.802 E.820 E.850 E.855 E.860 E.862 E.880	General (e.g. Internet access), telephone network	<ul style="list-style-type: none"> instantaneous availability/unavailability – probability defined for a network element of being in an “up”/ “down” state at a given instant of time mean time between failures (MTBF) – mean value of time duration between two consecutive failures of a repaired element mean time between interruptions (MTBI) – mean value of time duration measured between the end of one interruption and the beginning of the next one mean time to failure (MTTF) – mean value of time duration for a network element measured from the instant its state changes from “down” to “up” until the next failure mean time to recovery (MTTR) – mean value of time duration when a network element is in a “down” state due to a failure mean up time (MUT) / mean downtime (MDT) – interval during which an element is in an “up” / “down” state p-fractile repair time probability of fault coverage reliability function $R(t)$ – the probability that a network element can perform as expected under given conditions for a specified time interval retainability: a measure of the probability that a service will continue to be provided failure/repair rate $\lambda(t)/\mu(t)$
G.911	Fiber optic systems	<ul style="list-style-type: none"> failures in time (FIT) – the number of failures occurred per 10^9 device hours median life – a value on a lognormal probability plot of time to failure at which 50% of the devices fail earlier, and 50% of the devices fail later standard deviation – a value of a standard deviation concerning the natural logarithms of the time to failure availability (A), MTBF, MTTR, unavailability (U), $\lambda(t)$
M.60 M.3342	General	<ul style="list-style-type: none"> mean time to restore service (MTRS) – similar to MTTR but here related to the service level A, MTBF, MTTR, $R(t)$, retainability
Y.1540 Y.1541 Y.1542	IP	<ul style="list-style-type: none"> IP packet loss ratio (IPLR) – the total number of lost IP packets to the total number of transmitted IP packets in a given population of interest percent of IP service (un)availability (PIU/PIA) – the percentage of the total time of IP service categorized as (un)available based on the availability function of IP service service availability – a portion of the total scheduled service time for an IP service classified as “available”
Y.1561	MPLS (Multiprotocol Label Switching)	<ul style="list-style-type: none"> packet loss ratio (PLR) – analogous to IPLR recovery time – time needed for recovery actions at the MPLS layer calculated based on the number of successive time intervals of consecutive SLB outcomes at the ingress node severe loss block (SLB) outcome – an event occurring at an ingress node for a block of packets if the ratio of lost packets at an egress node exceeds the upper bound service availability, PIU, PIA – defined similarly as in Y.1540, but here related to SLB
Y.1562	Higher layer protocols	<ul style="list-style-type: none"> service availability – in Y.1562 related to the transfer delay and success ratio of service

Fig. 2.10 Selected resilience metrics defined by ITU-T based on [13]

on end-to-end transmission characteristics. Additionally, unlike QoS measures, resilience characteristics often cannot be derived precisely since, in many cases, they are not adequately perceived by the end users. For instance, increased transmission delay/packet losses may result from either congestion or a network element failure.

Recommendations of Internet Engineering Task Force (IETF) in Requests for Comments (RFCs)		
ID	Area	Metric
2330	IP	<ul style="list-style-type: none"> packet loss rate (PLR) – similar to IPLR
3386	Multi-layer networks	<ul style="list-style-type: none"> protection switching time – the time interval between the network fault occurrence until the completion of protection-switching actions restoration time – time interval from the network fault occurrence until the complete restoration of the affected traffic, exhaustion of spare resources, or existence of no more extra traffic
3469 4378	MPLS	<ul style="list-style-type: none"> availability – the percentage of time that a service is operating full restoration time – the time necessary to switch the traffic onto links/paths meant to handle the traffic in recovery scenarios number of concurrent faults – the number of faults a selected recovery scheme can cover recovery time – time needed for activation of an MPLS backup path (and resumption of affected traffic flows) after a fault setup vulnerability – measure of time when the primary path is left unprotected during recovery paths computations /setup
3945 4427 4428	GMPLS (Generalized Multiprotocol Label Switching)	<ul style="list-style-type: none"> recovery ratio – a fraction of the restored traffic bandwidth divided by the overall traffic bandwidth to be protected recovery time (down time)

Fig. 2.11 Selected resilience metrics defined by IETF based on [13]

2.5 Evaluation and Improvement of System Total Availability

In this part of the chapter, we focus on evaluating the system's total availability. We first discuss the formulas to determine the system's total availability for serial, parallel, and mixed interconnections of system elements. We also provide further insight concerning the availability evaluation for other (arbitrary) system structures. Since the availability of any networked system is built upon the availability of its elements, the respective calculations are presented here for the levels of element availability A_i assumed to be provided by equipment vendors. In the later part of this subsection, we also discuss possible directions for the design and update of architectures of networked systems to improve their overall availability characteristics.

2.5.1 Total Availability for Serial Systems

In the analysis of the availability (as well as reliability) of networked systems, their structure is commonly illustrated by *reliability block diagrams* (RBD) presenting interconnections of system elements [24]. In such diagrams, elements are assumed to be mutually independent (apart from dependencies represented by interconnections of elements in the RBD). In particular, RBD schemes are also valid for evaluating the availability of communication services, allowing for transmission of information between a given r -th pair of source s_r and destination t_r nodes in communication networks. Therefore, the methodology for evaluation of availability

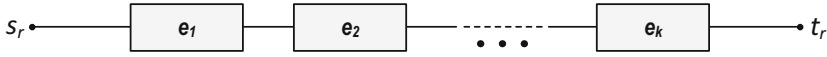


Fig. 2.12 An example of a serial system

Table 2.5 Example values of total availability A_T for serial systems consisting of $k = 1 \dots 8$ elements with values of availability (A) for system elements equal to 0.999, 0.950, 0.900, 0.600, and 0.300

Availability (A)	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$	$k = 8$
0.999	0.999	0.998	0.997	0.996	0.995	0.994	0.993	0.992
0.950	0.950	0.903	0.857	0.815	0.774	0.735	0.698	0.663
0.900	0.900	0.810	0.729	0.656	0.590	0.531	0.478	0.430
0.600	0.600	0.360	0.216	0.130	0.078	0.047	0.028	0.017
0.300	0.300	0.090	0.027	0.008	0.002	0.001	<0.001	<0.001

presented in this subsection (as well as for the assessment of reliability provided later in this chapter) can be associated straightforwardly with the example scenarios of the end-to-end (i.e., two-terminal) transmission between nodes s_r and t_r in communication networks [8].

Figure 2.12 provides an example of a serial system consisting of k elements e_i in series, each element characterized by its level of availability A_i . Any serial system is considered to be available at a given time t if all its elements are available at time t so that the respective service can be provided between its endpoints s_r and t_r . Otherwise, a failure of even one element implies a failure of the entire serial system. Therefore, the total availability A_T of a serial system is determined as a product of availabilities of all its elements, as given in Eq. 2.14.

$$A_T = \prod_{i=1}^k A_i \quad (2.14)$$

In particular, assuming an equal value of availability A for all k components, the total availability of a system is given by Eq. 2.15. The overall availability is thus much affected by less available elements [25].

$$A_T = A^k \quad (2.15)$$

Table 2.5 presents values of total availability calculated for serial systems with a number of elements k ranging from 1 to 8, assuming equal values of availability (A) equivalent to 0.999, 0.950, 0.900, 0.600, and 0.300 for all system elements. Since for any serial system to be available, all its elements need to be available at the same time, as shown in Table 2.5, the total availability of serial systems tends to decrease rapidly with the increase of the number of system elements even for relatively high availability ratios $A = 0.950$ of individual elements. Therefore, a

Fig. 2.13 An example of a parallel system

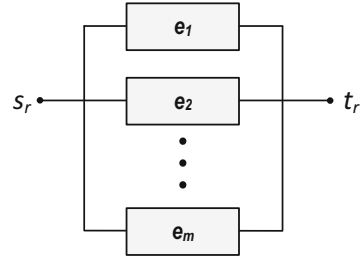


Table 2.6 Example values of total system availability for parallel systems consisting of $m = 1 \dots 8$ elements for the case of equal values of availability (A) for all system elements

Availability (A)	$m = 1$	$m = 2$	$m = 3$	$m = 4$	$m = 5$	$m = 6$	$m = 7$	$m = 8$
0.999	0.9990	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
0.950	0.9500	0.9975	0.9999	1.0000	1.0000	1.0000	1.0000	1.0000
0.900	0.9000	0.9900	0.9990	0.9999	1.0000	1.0000	1.0000	1.0000
0.600	0.6000	0.8400	0.9360	0.9744	0.9898	0.9959	0.9984	0.9993
0.300	0.3000	0.5100	0.6570	0.7599	0.8319	0.8824	0.9176	0.9424

general conclusion from these results is that serial systems are not well-suited for highly resilient system architectures.

2.5.2 Total Availability for Parallel Systems

A parallel system consisting of m elements illustrated in Fig. 2.13 is considered available at time t if at least one of its elements is available at time t . Such a system can be, e.g., a communication network where data transmission between nodes s_r and t_r is possible if it can be assured by at least one of its parallel segments between s_r and t_r . In the example from Fig. 2.13, this is true when at least one of its elements is operational at time t .

The total availability A_T of a parallel system is thus calculated as the complementary value of the probability of system unavailability as given in Eq. 2.16. In particular, assuming equal values of availability A for all m system elements, Eq. 2.16 can be substituted by Eq. 2.17.

$$A_T = 1 - U_T = 1 - \prod_{i=1}^m U_i = 1 - \prod_{i=1}^m (1 - A_i) \tag{2.16}$$

$$A_T = 1 - (1 - A)^m \tag{2.17}$$

Table 2.6 provides information on the total availability for parallel systems consisting of $m=1 \dots 8$ elements characterized by equal values of individual element

availability A . As shown in Table 2.6, unlike for serial systems, increasing the number of elements m of a parallel system increases its total availability. A remarkable improvement of A_T is observed even for very low values of individual availability ratios A of 0.6 or even 0.3. Parallelization of system elements is thus an excellent way to improve the overall system availability since, with the increase of the number m of parallel elements, it becomes less likely that all m elements of a system will become nonoperational at the same time.

2.5.3 Total Availability for Mixed Systems

In practice, networked systems are often neither purely serial nor parallel. However, to a remarkable extent, they possess characteristics of both types, as shown in Fig. 2.14. This observation is also accurate for system total availability characteristics.

In the case of a “mixed” structure of a networked system consisting of m parallel segments, each one including k_i serial elements, the total availability of a system concerning the multi-hop transmission between nodes s_r and t_r can be obtained according to Eq. 2.18.

$$A_T = 1 - \prod_{i=1}^m (1 - \prod_{j=1}^{k_i} A_{i,j}) \tag{2.18}$$

Tables 2.7–2.10 present characteristics of total availability calculated for systems consisting of $m \times n$ elements (i.e., comprising m parallel subsystems, each such subsystem consisting of n serial elements), where each element is characterized by an equal availability level A . As expected, in mixed systems, parallelization (represented by the m factor in these tables) improves the total system availability (the higher the m value, the higher the total system availability). In contrast, serialization (modeled by the n coefficient) plays the opposite role. It is worth noting

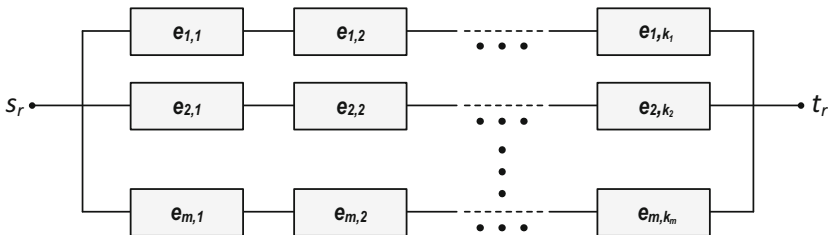


Fig. 2.14 An example of a mixed system

Table 2.7 Total system availability for a mixed system consisting of m parallel subsystems, each subsystem consisting of k serial elements and equal values of availability $A = 0.950$ for all system elements

	$k = 2$	$k = 4$	$k = 6$	$k = 8$
$m = 2$	0.990	0.966	0.930	0.887
$m = 4$	1.000	0.999	0.995	0.987
$m = 6$	1.000	1.000	1.000	0.999
$m = 8$	1.000	1.000	1.000	1.000

Table 2.8 Total system availability for a mixed system consisting of m parallel subsystems, each subsystem consisting of k serial elements and equal values of availability $A = 0.900$ for all system elements

	$k = 2$	$k = 4$	$k = 6$	$k = 8$
$m = 2$	0.964	0.882	0.780	0.676
$m = 4$	0.999	0.986	0.952	0.895
$m = 6$	1.000	0.998	0.989	0.966
$m = 8$	1.000	1.000	0.998	0.989

Table 2.9 Total system availability for a mixed system consisting of m parallel subsystems, each subsystem consisting of k serial elements and equal values of availability $A = 0.850$ for all system elements

	$k = 2$	$k = 4$	$k = 6$	$k = 8$
$m = 2$	0.923	0.772	0.612	0.471
$m = 4$	0.994	0.948	0.850	0.720
$m = 6$	1.000	0.988	0.942	0.852
$m = 8$	1.000	0.997	0.977	0.922

Table 2.10 Total system availability for a mixed system consisting of m parallel subsystems, each subsystem consisting of k serial elements and equal values of availability $A = 0.750$ for all system elements

	$k = 2$	$k = 4$	$k = 6$	$k = 8$
$m = 2$	0.809	0.533	0.324	0.190
$m = 4$	0.963	0.782	0.543	0.344
$m = 6$	0.993	0.898	0.691	0.469
$m = 8$	0.999	0.952	0.792	0.570

that for configurations where $m = n$ (see results located on the main diagonal of these tables), the total system availability tends to improve with the system size for $A > 0.85$. However, the opposite tendency can be observed in Table 2.10 for a lower value of $A = 0.750$.

2.5.4 A General Strategy to Determine the Total Availability of Complex Systems

The complexity of networked systems can significantly complicate the task of determining their total availability level. However, as the system total availability A_T is impacted by the respective availability levels of system elements, as noted in [3], when determining A_T , for each system, it is sufficient to take the following consecutive steps:

1. Decompose the system into subsystems.
2. Determine the formulas defining the availability dependencies (relations) among subsystems.
3. Determine the availability level for each subsystem.
4. Calculate the total availability of a system based on the availability levels of its subsystems.

Calculating A_T is generally simpler for systems characterized by “nested” structures. In such cases, A_T can be obtained stepwise by iteratively replacing certain system parts (representing either basic parallel or serial structures) with single components. Such an approach is reasonable since a given part of a system is a system itself. Example steps of this reduction technique are presented in Fig. 2.15.

In particular, to calculate the total availability of a system from Fig. 2.15a, in the first step, a basic structure of a serial subsystem is identified (consisting of two elements of availability A_2 and A_3), and the availability A' referring to that subsystem is determined. After that, the considered subsystem of availability A' is treated as a single component for further calculations in Fig. 2.15b, where the availability of a parallel subsystem consisting of two elements of availability A_1 and A' can be determined. This subsystem with the level of availability A'' is further considered in Fig. 2.15c as the element forming a serial system with another element of availability A_4 , for which the total availability A''' of a single block shown in Fig. 2.15d can finally be obtained.

The technique illustrated in Fig. 2.15 is generally sufficient for systems composed of nested serial and parallel subsystems. However, there also exist systems for which decomposition into parallel and serial parts is harder, e.g., concerning

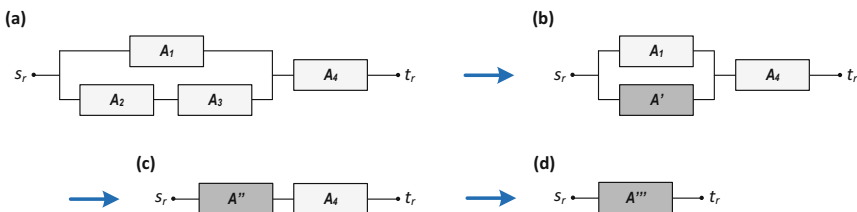


Fig. 2.15 An example of a general iterative strategy to calculate the total availability of a complex system shown in (a) by three consecutive reductions illustrated in (b)–(d)

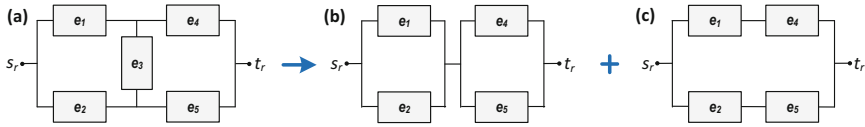


Fig. 2.16 An example of a decomposition strategy applied to determine the availability of a system characterized by cyclic interconnections of elements

structures consisting of elements forming cycles. As discussed in [8], for such interconnections of system elements, there is a need to apply a decomposition strategy following a general Bayes theorem $P(A) = P(A|B) \cdot P(B) + P(A|\bar{B}) \cdot P(\bar{B})$, as presented in the example Fig. 2.16a, assuming that $P(B)$ reflects the probability of a correct state of element e_3 . $P(\bar{B})$, in turn, refers to the probability of e_3 non-correct state.

As presented in Fig. 2.16, each such cyclic structure can be decomposed into two schemes representing two states for a joint element of a system being either available (denoted by availability level A_3) or unavailable (unavailability level of $1-A_3$), illustrated by two cases in Fig. 2.16b and c, respectively. Assuming that the total availability of a system from Fig. 2.16b is represented by A_{T1} , while the one for the case illustrated by Fig. 2.16c is given by A_{T2} , the overall availability of a system from Fig. 2.16a is determined as given in Eq. 2.19.

$$A_T = A_3 A_{T1} + (1 - A_3) A_{T2} \tag{2.19}$$

2.5.5 Improvement of Networked Systems Availability

As noted in [14], the three major aspects crucial in obtaining highly dependable networked systems include:

- Assurance of a highly available environment (including the power supply, security procedures, and physical protection)
- Deployment of networked systems composed of highly available software and hardware units
- Experience and best practices necessary to design and maintain highly available systems

Indeed, environmental features may play a significant role in assuring a high level of availability of networked systems, e.g., by utilizing highly available power supplies equipped with backup power devices, hiring experienced personnel, and implementing stringent security and safety rules. A high level of modularity and redundancy and reduced complexity are those aspects that, if enforced, can contribute to a further increase in the overall system availability. Similar effects can be expected while drawing the best from experience and best practices.

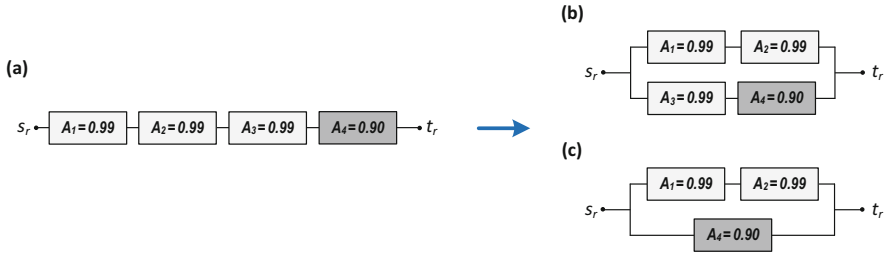


Fig. 2.17 Example redesign options to improve the system total availability by replacing a serial part of a system (a) by parallel variants provided in (b) and (c)

It is worth noting that improving the system's total availability does not always come at an additional cost. Several strategies based on adding redundant elements (such as backup servers, routers, and alternate transmission paths) to improve the total availability of a system indeed imply additional costs. However, it turns out that a higher level of system's total availability can also be achieved without additional costs (or can be obtained even at a lower cost) when deploying specific techniques of redesigning a system architecture, leading to the utilization of the same or a reduced set of system elements.

For instance, in the example configuration of a system shown in Fig. 2.17a consisting of four serial elements, three elements of availability $A_1=A_2=A_3=0.99$ and a fourth one with $A_4 = 0.900$, the overall availability of a system $A_T = A_1A_2A_3A_4$ is equal to $0.99 \cdot 0.99 \cdot 0.99 \cdot 0.9 = 0.873$. However, by introducing parallelization of these four elements in a way shown in Fig. 2.17b, the overall system availability is higher (i.e., $A_T=1-(1-A_1A_2)(1-A_3A_4)=0.997$). It is worth noting that even after removing one element of high availability (e.g., element of availability $A_3 = 0.99$), the overall availability of a parallel system illustrated in Fig. 2.17c $A_T=1-(1-A_1A_2)(1-A_4)=0.998$ is still higher than that of the original serial system from Fig. 2.17a, as well as higher than that in the case of a system from Fig. 2.17b. At the same time, the total cost of a parallel system from Fig. 2.17c consisting of three elements is obviously lower than that of the original serial system from Fig. 2.17a comprising four elements.

2.6 Evaluation of System Reliability

As discussed earlier in this chapter, reliability refers to the ability of a system to remain operational in a given time interval $(0, t)$. Reliability can thus be considered a measure of continuity reflecting the probability for a system to operate continuously without any failure, as formulated earlier in Eq. 2.7.

When determining the reliability of a given system, similar observations may be applied as for the case of the evaluation of a system's overall availability from Sect. 2.5. Therefore, in the case of a serial configuration of k system elements (see Fig. 2.12) with independent failures, the overall system reliability $R_T(t)$ referring to its end-to-end undisturbed operation in time interval $(0, t)$ is defined by a product of reliability values of system elements as given in Eq. 2.20.

$$R_T(t) = \prod_{i=1}^k R_i(t) \quad (2.20)$$

In particular, under a common assumption of the exponential form of $R_i(t)$ (see Eq. 2.8), $R_T(t)$ of a serial system can be specified as given in Eq. 2.21. For serial systems composed of elements characterized by the exponential reliability functions $R_i(t)$, the system failure rate λ_T is, therefore, the sum of the individual failure rates λ_i .

$$R_T(t) = \prod_{i=1}^k e^{-\lambda_i t} = e^{-(\lambda_1 + \lambda_2 + \dots + \lambda_k)t} = e^{-\lambda_T t} \quad (2.21)$$

where $\lambda_T = \sum_{i=1}^k \lambda_i$.

Assuming that the reliability functions $R_i(t)$ are equal for all system elements (i.e., $\forall_i R_i(t) = R(t)$), the formula for the overall reliability of a serial system from Eq. 2.20 can be rewritten as given in Eq. 2.22, and as provided in Eq. 2.23 assuming the exponential form of $R(t) = e^{-\lambda t}$.

$$R_T(t) = R(t)^k \quad (2.22)$$

$$R_T(t) = e^{-k\lambda t} \quad (2.23)$$

Equation 2.23 confirms a common expectation that for serial systems consisting of k elements characterized by the same individual reliability functions $R(t)$, the failure rate λ_T of a serial system is k times higher than the failure rate λ of individual elements (i.e., $\lambda_T = k\lambda$).

For a parallel system illustrated in Fig. 2.13 assumed to be operational in time interval $(0, t)$, if at least one of its m elements is operational in that interval, the overall system reliability can be obtained by formula (2.24).

$$R_T(t) = 1 - \prod_{i=1}^m (1 - R_i(t)) \quad (2.24)$$

Similarly, assuming the exponential form of the reliability formula for each system element ($R_i(t) = e^{-\lambda_i t}$), the overall reliability of a parallel system can be expressed as given in Eq. 2.25.

$$R_T(t) = 1 - \prod_{i=1}^m (1 - e^{-\lambda_i t}) \quad (2.25)$$

The methodology for determining the system total reliability for complex architectures, together with strategies for improving the overall system reliability, follows the general rules explained for the total system availability investigated in Sect. 2.5.

2.7 Summary

Despite efforts to reduce the frequency of failures of elements in networked systems, e.g., through the use of highly reliable components for deploying system elements and methods of physical protection against external challenges, failures in networked systems will continue to occur. Their complete elimination is not possible, mainly due to the multitude of external factors, the number, intensity, and scale of which are increasing, as well as the complexity of system architectures and related software.

In this chapter, we focused on analyzing a broad set of challenges leading to failures of networked system elements and on explaining the related chain of events potentially leading to failures of system services. We also discussed the desired characteristics of resilient systems. We demonstrated how to evaluate the system's total availability and reliability from the perspective of communication services, i.e., by analyzing the system's end-to-end availability and reliability between given end nodes s_r and t_r .

We agree that in practice, for a number of services, system configurations can be even more complex than in the examples provided in this chapter (see, e.g., schemes of resilient transmission utilizing shared backup paths or the concepts of redundant computational/storage units, discussed in Chapter 4 in this book). Regardless of the configuration of services, from the resilience point of view, the most important thing is to ensure that the failure of any physical element of a system will not result in a service failure that the user would notice.

For this purpose, the mechanisms of networked systems resilience (described in the following chapters of this book) should, in fact, become an integral part of the design of any networked system architecture.

? Questions

1. Characterize the major challenges for networked systems and discuss their potential impact on system elements.
 2. Explain the meaning of a fault and present the taxonomy of faults along with their characteristics.
 3. Characterize the chain of events leading to failures in networked systems.
 4. List the disciplines of resilience and provide their definitions and characteristics.
 5. Characterize the attributes of dependability: availability and reliability. Discuss their differences and usability scenarios.
 6. Explain the differences in structures of serial and parallel networked systems. Discuss the main characteristics of architectures of mixed systems.
 7. Explain the methodology for evaluation of the system total availability for serial and parallel systems as well as the respective general strategy for complex systems.
 8. Discuss possible ways of increasing the level of availability of networked systems.
 9. Characterize the methodology of evaluating and improving the system's total reliability.
-

References

1. A10: <https://www.a10networks.com/blog/5-most-famous-ddos-attacks/>. Accessed 14 Apr 2023
2. Agarwal, P.K., Efrat, A., Ganjugunte, S., Hay, D., Sankararaman, S., Zussman, G.: The resilience of WDM networks to probabilistic geographical failures. In: Proceedings of the 30th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM'11), pp. 1521–1529 (2011)
3. Arci, D., Maier, G., Pattavina, A., Petecchi, D., Tornatore, M.: Availability models for protection techniques in WDM networks. In: Proceedings of the 4th International Workshop on Design of Reliable Communication Networks (DRCN'03), pp. 158–166 (2003)
4. Avizienis, A.: Fault-tolerant systems. *IEEE Trans. Comput.* **C-25**(12), 1304–1312 (1976)
5. Avizienis, A., Laprie, J.-C., Randell, B.: Dependability and its threats: A taxonomy. In: Jacquart, R. (eds) Building the Information Society. IFIP International Federation for Information Processing, vol. 156. Springer, Berlin, pp. 91–120 (2004)
6. Avizienis, A., Laprie, J.-C., Randell, B., Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. Technical Research Report TR2004-47, Institute for Systems Research, The University of Maryland (2004)
7. Avizienis, A., Laprie, J.-C., Randell, B., Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. *Trans. Depend. Secure Comput.* **1**(1), 11–33 (2004)
8. Cabarkapa, M., Mijatovic, D., Krajnovic, N.: Network topology availability analysis. *Telfor J.* **3**(1), 23–27 (2011)
9. Caini, C., Cruickshank, H., Farrell, S., Marchese, M.: Delay- and disruption-tolerant networking (DTN): An alternative solution for future satellite networking applications. *Proc. IEEE* **99**(11), 1980–1997 (2011)

10. Cetinkaya, E.K., Sterbenz, J.P.G.: A taxonomy of network challenges. In: Proceedings of the 9th International Conference on Design of Reliable Communication Networks (DRCN'13), pp. 322–330 (2013)
11. Cholda, P., Jajszczyk, A.: Recovery and its quality in multilayer networks. *IEEE/OSA J. Lightwave Technol.* **28**(4), 372–389 (2010)
12. Cholda, P., Mykkeltveit, A., Helvik, B.E., Wittner, O.J., Jajszczyk, A.: A survey of resilience differentiation frameworks in communication networks. *IEEE Commun. Surv. Tutorials* **9**(4), 32–55 (2007)
13. Cholda, P., Tapolcai, J., Cinkler, T., Wajda, K., Jajszczyk, A.: Quality of Resilience as a network reliability characterization tool. *IEEE Netw.* **23**(2), 11–19 (2009)
14. Cisco Systems Whitepaper: Network availability: How much do you need? How do you get it? (2004)
15. Cucurull, J., Asplund, M., Nadjm-Tehrani, S., Santoro, T.: Surviving attacks in challenged networks. *IEEE Trans. Depend. Secure Comput.* **9**(6), 917–929 (2012)
16. Dainotti, A., Squarcella, C., Aben, E., Claffy, K.C., Chiesa, M., Russo, M., Pescapé, A.: Analysis of country-wide Internet outages caused by censorship. In: Proceedings of the ACM Internet Measurement Conference (IMC'11), pp. 1–18 (2011)
17. Dikbiyik, F., Tornatore, M., Mukherjee, B.: Minimizing the risk from disaster failures in optical backbone networks. *IEEE/OSA J. Lightwave Technol.* **32**(18), 3175–3183 (2014)
18. Fangming, L., Bo, L., Lili, Z., Baochun, L., Hai, J., Xiaofei, L.: Flash crowd in P2P live streaming systems: Fundamental characteristics and design implications. *IEEE Trans. Parallel Distrib. Syst.* **23**(7), 1227–1239 (2012)
19. Fry, M., Fischer, M., Karaliopoulos, M., Smith, P., Hutchison, D.: Challenge identification for network resilience. In: Proceedings of the 6th EURO-NF Conference on Next Generation Internet (NGI'10), pp. 1–8 (2010)
20. Geva, M., Herzberg, A., Gev, Y.: Bandwidth distributed denial of service: Attacks and defences. *IEEE Security Privacy* **12**(1), 54–61 (2014)
21. Grottko, M., Trivedi, K.S.: A classification of software faults. In: Proceedings of the 16th International IEEE Symposium on Software Reliability Engineering, pp. 4.19–4.20 (2005)
22. Haider, A., Harris, R.: Recovery techniques in Next Generation Networks. *IEEE Commun. Surv. Tutorials* **9**(3), 2–17 (2004)
23. Heegaard, P.E., Trivedi, K.S.: Network survivability modeling. *Comput. Netw.* **53**(8), 1215–1234 (2009)
24. Held, M., Nellen, P., Wosinska, L.: Availability calculation and simulation of optical network systems. *Proc. SPIE* **3940** (2003)
25. Hilt, A., Jaro, G., Bakos, I.: Availability prediction of telecommunication application servers deployed on cloud. *Periodica Polytechn. Electr. Eng.* **60**, 72–81 (2016)
26. ITU-T Rec. E.800: Terms and definitions related to Quality of Service and network performance including dependability (1994)
27. ITU-T Rec. E.826: Digital networks – Quality and availability targets: End-to-end error performance parameters and objectives for international, constant bit-rate digital paths and connections (2002)
28. Jung, J., Krishnamurthy, B., Rabinovich, M.: Flash crowds and denial-of-service attacks: Characterization and implication for CDNs and web sites. In: Proceedings of the 11th International Conference on World Wide Web (WWW'02), pp. 293–304 (2002)
29. Kalesnikau, I., Pióro, M., Rak, J., Ivanov, H., Fitzgerald, E., Leitgeb, E.: Enhancing resilience of FSO networks to adverse weather conditions. *IEEE Access* **9**, 123541–123565 (2021)
30. Kappenman, J.: A perfect storm of planetary proportions. *IEEE Spectr. Mag.* **49**(2), 26–31 (2012)
31. Khabbaz, M.J., Assi, C.M., Fawaz, W.F.: Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challenges. *IEEE Commun. Surv. Tutorials* **14**(2), 607–640 (2012)
32. Kiaei, M.S., Assi, C., Jaumard, B.: A survey on the p -Cycle protection method. *IEEE Commun. Surv. Tutorials* **11**(3), 53–70 (2009)

33. Kitamura, Y., Lee, Y., Sakiyama, R., Okamura, K.: Experience with restoration of Asia Pacific network failures from Taiwan earthquake. *IEICE Trans. Commun.* **E90-B**(11), 3095–3103 (2007)
34. Krock, R.E.: Effective quality control during disaster recovery. *Bell Labs Tech. J.* **9**(2), 163–171 (2004)
35. Kwasinski, A., Weaver, W.W., Chapman, P.I., Krein, P.T.: Telecommunications power plant damage assessment for Hurricane Katrina – site survey and follow-up results. *IEEE Syst. J.* **3**(3), 277–287 (2011)
36. Laprie, J.-C.: Dependability: basic concepts and terminology. IFIP Working Group 10.4 – Dependable Computing and Fault Tolerance, 1–47 (1994)
37. Laprie, J.-C.: Resilience for the scalability of dependability. In: *Proceedings of the 4th IEEE International Symposium on Network Computing and Applications*, pp. 5–6 (2005)
38. Martins, L., Girao-Silva, R., Jorge, L., Gomes, A., Musumeci, F., Rak, J.: Interdependence between power grids and communication networks: A resilience perspective. In: *Proceedings of the 2017 International Conference on Design of Reliable Communication Networks (DRCN'17)*, pp. 1–9 (2017)
39. Maruyama, H., Legaspi, R., Minami, K., Yamagata, Y.: General resilience: Taxonomy and strategies. In: *Proceedings of the 2014 International Conference and Utility Exhibition on Green Energy for Sustainable Development (ICUE'14)*, pp. 1–8 (2014)
40. Mili, A., Cukic, B., Xia, T., Ben Ayed, R.: Combining fault avoidance, fault removal and fault tolerance: An integrated model. In: *Proceedings of the 14th IEEE International Conference on Automated Software Engineering*, pp. 137–146 (1999)
41. Mingsen, X., Wen-Zhan, S., Deukhyoun, H., Jong-Hoon, K., Byeong-Sam, K.: ECPC: Preserve downtime data persistence in disruptive sensor networks. In: *Proceedings of the IEEE Mobile Ad-Hoc and Sensor Systems (MASS'13)*, pp. 281–289 (2013)
42. Misseri, X., Gojmerac, I., Rougier, J.-L.: IDRd: Enabling inter-domain route diversity. In: *Proceedings of the IEEE International Conference on Communications (IEEE ICC'13)*, pp. 3536–3541 (2013)
43. Molisz, W.: Survivability function: A measure of disaster-based routing performance. *IEEE J. Sel. Areas Commun.* **22**(9), 1876–1883 (2004)
44. Mukherjee, B., Habib, M.F., Dikbiyik, F.: Network adaptability from disaster disruptions and cascading failures. *IEEE Commun. Mag.* **52**(5), 230–238 (2014)
45. NATHAN world map of natural hazards, Munich RE (2011)
46. Neumayer, S., Zussman, G., Cohen, R., Modiano, E.: Assessing the vulnerability of the fiber infrastructure to disasters. *IEEE/ACM Trans. Netw.* **19**(6), 1610–1623 (2011)
47. Nicol, D.M., Sanders, W.H., Trivedi, K.S.: Model-based evaluation: from dependability to security. *IEEE Trans. Depend. Secure Comput.* **1**(1), 48–65 (2004)
48. Pham, H. (ed.): *Handbook of Engineering Statistics*. Springer, Berlin (2023)
49. Public Safety and Homeland Security Bureau Federal Communications Commission (FCC): 2017 Atlantic Hurricane Season Impact on Communications Report and Recommendations Public Safety Docket No. 17-344 (August 2018)
50. Ran, Y.: Considerations and suggestions on improvement of communication network disaster countermeasures after the Wenchuan earthquake. *IEEE Commun. Mag.* **49**(1), 44–47 (2011)
51. Rak, J., Hutchison, D. (eds.): *Guide to Disaster-Resilient Communication Networks*. Springer, Berlin (2020)
52. Said Elsayed, M., Le-Khac, N.-A., Jurcut, A.D.: Dealing with COVID-19 network traffic spikes. *IEEE Security Privacy* **19**(1), 90–94 (2021)
53. Sakano, T., Fadlullah, Z.Md., Ngo, T., Nishiyama, H., Nakazawa, M., Adachi, F., Kato, N., Takahara, A., Kumagai, T., Kasahara, H., Kurihara, S.: Disaster-resilient networking: A new vision based on movable and deployable resource units. *IEEE Netw.* **27**(4), 40–46 (2013)
54. Sauve, J.P., Silva Coelho, F.E.: Availability considerations in network design. In: *Proceedings of Pacific Rim International Symposium on Dependable Computing*, pp. 119–126 (2001)
55. Sichiitiu, M.L., Kihl, M.: Inter-vehicle communication systems: A survey. *IEEE Commun. Surv. Tutorials* **10**(2), 88–105 (2008)

56. Smith, P., Hutchison, D., Sterbenz, J.P.G., Schoeller, M., Fessi, A., Karaliopoulos, M., Lac, M., Plattner, B.: Network resilience: A systematic approach. *IEEE Commun. Mag.* **49**(7), 88–97 (2011)
57. Steinder, M., Sethi, A.: A survey of fault localization techniques in computer networks. *Sci. Comput. Program.* **53**(2), 165–194 (2004)
58. Sterbenz, J.P.G., Çetinkaya, E.K., Hameed, M.A., Jabbar, A., Qian, S., Rohrer, J.P.: Evaluation of network resilience, survivability, and disruption tolerance: analysis, topology generation, simulation, and experimentation. *Telecommun. Syst.* **52**(2), 705–736 (2013)
59. Sterbenz, J.P.G., Hutchison, D., Çetinkaya, E.K., Jabbar, A., Rohrer, J.P., Schöller, M., Smith, P.: Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Comput. Netw.* **54**(8), 1245–1265 (2010)
60. T1A1.2 Working Group: Reliability-related metrics and terminology for network elements in evolving communication networks. American National Standard for Telecommunications T1.R1.524-2004, Alliance for Telecommunications Industry Solutions – ATIS (2004)
61. Tornell, S.M., Calafate, C.T., Cano, J.-C., Manzoni, P.: DTN protocols for vehicular networks: An application oriented overview. *IEEE Commun. Surv. Tutorials* **17**(2), 868–887 (2015)
62. Urushidani, S., Aoki, M., Fukuda, K., Abe, S., Nakamura, M., Koibuchi, M., Ji, Y., Yamada, S.: Highly available network design and resource management of SINET4. *Telecommun. Syst.* **56**(1), 33–47 (2014)
63. Wu, W., Moran, B., Manton, J.H., Zukerman, M.: Topology design of undersea cables considering survivability under major disasters. In: *Proceedings of the 2009 International Conference on Advanced Information Networking and Applications (WAINA'09)*, pp. 1154–1159 (2009)
64. Xanthopoulos, G., Athanasiou, M.: Attica region, Greece July 2018: A tale of two fires and a seaside tragedy. *Wildfire* **28**(2), 18–21 (2019)
65. Xu, C., Xiong, Z., Zhao, G., Yu, S.: An energy-efficient region source routing protocol for lifetime maximization in WSN. *IEEE Access* **7**, 135277–135289 (2019)
66. Zargar, S.T., Joshi, J., Tipper, D.: A survey of defense mechanisms against Distributed Denial of Service (DDoS) flooding attacks. *IEEE Commun. Surv. Tutorials* **15**(4), 2046–2069 (2013)

Chapter 3

System- and Element-Related Metrics Useful in the Evaluation of Resilience



Failures of network elements will undoubtedly continue to occur since it is impossible to eliminate all the factors responsible for them. However, as we discuss in this chapter, the scale of the negative consequences of failure events is determined not only by the characteristics of the related challenges (such as their intensity, duration, and area, as, e.g., in the case of heavy rainfall, fire, hurricane) but also follows from the properties of the system architecture such as system topology, location of servers providing services to end users, transmission schemes, etc.

For instance, if transmission of information is configured via shortest paths (which is a common scenario), then due to the topological properties of networked systems, such as the location of a given node in the topology or the number of links attached to that node, certain network nodes tend to switch a greater amount of network traffic and, therefore, are of greater importance than the other nodes. This also means that their failure significantly impacts the provisioning of services to the end users, as many more transmission paths become affected. Similarly, a malicious attack leading to the failure of a server providing a multitude of services may be a direct consequence of the recognition by an attacker of the properties of that node.

Therefore, to assess the potential impact of a failure of a given network element on the functioning of the entire system, it is important to make use of a set of *metrics*, i.e., functions designed to measure either the individual properties of certain elements or of the entire system and its services. Apart from their essential role in assessing system properties during normal operation and failure scenarios, these metrics can also be helpful in all phases of design, deployment, and update/evolution of the networked system architecture.

Understanding the meaning of certain metrics quite often requires at least some level of knowledge on characteristics of individual elements (nodes/links) of the networked system, as well as the architectural properties of the entire system impacting its performance, being the ability of a unit to provide the function it has been designed for [30].

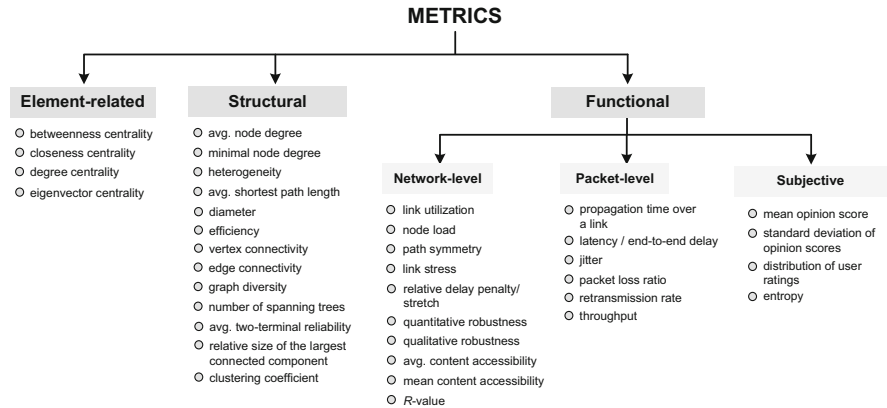


Fig. 3.1 A comprehensive classification for metrics of networked systems and their elements relevant in resilience evaluation

As the set of metrics for networked systems and their elements is relatively large, a particular focus in this chapter is on metrics useful from the perspective of the resilient functioning of a networked system in failure scenarios. In this context, as presented in Fig. 3.1, metrics relevant to evaluating the resilience of networked systems can be broadly divided into three categories: element-related, structural, and functional.

The first group of *element-related metrics* focuses on the properties of individual network elements (nodes/links) following from their existence in the system topology. The *structural metrics*, in turn, refer to the topological properties of the entire system. In contrast, *functional metrics* are used to analyze the system quality of service either at the network level (i.e., *network-level functional metrics*), at the packet level (referred to as *packet-level functional metrics*), or to assess user satisfaction with the service (often called quality of experience—QoE) referred to as the *subjective metrics*.

In the remaining part of this chapter, we first highlight in Sect. 3.1 the standard means of representing the topological properties of a system derived from the graph theory, which are useful in definitions of metrics analyzed later in this chapter. Next, in Sect. 3.2, we discuss the most important metrics dedicated to single elements of the system. Section 3.3 provides information about the most essential structural metrics. In Sect. 3.4, we explain the reasons for the diverse characteristics of system elements, the related irregular character of the system topology, and the resulting potential challenges. In Sect. 3.5, we analyze the major functional metrics, i.e., the ones for the evaluation of system performance at the network level and the packet level, as well as the subjective metrics referring to the satisfaction of users. In Sect. 3.6, we comment on examples of practical applications of the analyzed metrics in common use (e.g., in the configuration of routing protocols) as well as discuss proposals following from research papers for the use of these metrics at virtually every stage of the network system life cycle. Sect. 3.7 concludes the chapter.

3.1 The Formal Representation of Networked Systems Architecture

The architecture of networked systems consisting of a set N of nodes such as switches, routers, servers, etc., interconnected by communication links is commonly defined by graph $G(V, E)$, where V is a set of vertices representing the system nodes, $|V|$ is the number of vertices in G , while E stands for the set of edges of G representing the communication links. A given edge $e_{i,j}$ from E is assumed to interconnect the respective vertices v_i and v_j from V .

Set E of edges often represents *bidirectional network links* enabling transmission in both directions and often characterized by the same capacity $c_{i,j}$ in both directions, as illustrated by graph G_1 in Fig. 3.2a. However, as communication links are *directional* in certain configurations, they are then typically represented by directed arcs $a_h = (i, j)$ from set A (instead of set E). Therefore, in such cases, graph G takes the form of $G(V, A)$. An example representation of a networked system with directional communication links by graph G_2 with directed arcs is provided in Fig. 3.2b.

In general, the structure of any networked system can be defined by graph G at its various abstraction layers, such as the link layer (representing the system topology formed by physical links) or the Internet layer topology (formed by Internet links) [23].

Interconnection of network nodes (represented by a set V of vertices) by communication links represented by set E (or set A) for networks with bidirectional (or directional) links is often defined by the respective *adjacency matrix* \mathcal{A} with elements $\hat{a}_{i,j}$ equal to 1 denoting the existence of communication link from network node i to network node j . Otherwise, $\hat{a}_{i,j}$ values are set to 0. Network nodes i and j are called neighbors if the respective vertices v_i and v_j are adjacent in G , i.e., connected by edge $e_{i,j}$ (or arc $a_h = (i, j)$, respectively).

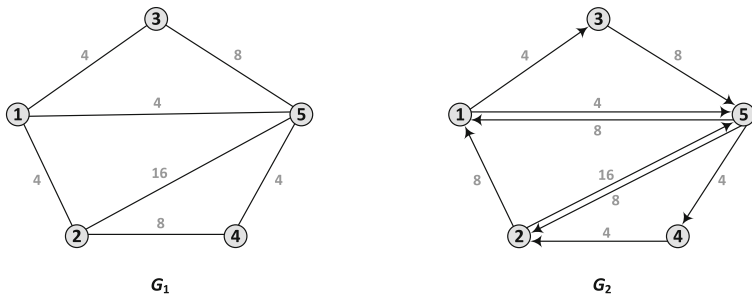


Fig. 3.2 Example graphs G_1 and G_2 representing networked systems with bidirectional and directional communication links, respectively (the numerical values located close to the respective edges/arcs denote the nominal capacity of network links)

Example adjacency matrices for graphs G_1 and G_2 from Fig. 3.2, denoted as \mathcal{A}_1 and \mathcal{A}_2 , are then defined as follows:

$$\mathcal{A}_1 = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \quad \mathcal{A}_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

It is clear that for networked systems with bidirectional (duplex) links, the respective matrices \mathcal{A} are symmetrical, i.e., such that for every pair of nodes i and j connected by a duplex link, $\hat{a}_{i,j} = \hat{a}_{j,i} = 1$ (and 0, otherwise). This is also the case for matrix \mathcal{A}_1 provided for graph G_1 from Fig. 3.2a.

However, for networks with directional (simplex) links, for a given pair of network nodes i and j , transmission is often possible in one way only (e.g., from a given node i to a particular node j , but not vice versa). Therefore, the adjacency matrix for networks with simplex links need not necessarily be symmetrical, as in the case of matrix \mathcal{A}_2 above representing the interconnections of network nodes defined by graph G_2 in Fig. 3.2b.

Adjacency matrices can also provide additional information related to network links, such as link nominal capacity. For this purpose, values of $\hat{a}_{i,j}$ are replaced by the respective weights $c_{i,j}$, which leads to the concept of *weighted adjacency matrix* C . For graphs G_1 and G_2 from Fig. 3.2, the respective weighted adjacency matrices C_1 (symmetrical) and C_2 (nonsymmetrical), with weights $c_{i,j}$ denoting the nominal capacities of network links, are defined as follows:

$$C_1 = \begin{pmatrix} 0 & 4 & 4 & 0 & 4 \\ 4 & 0 & 0 & 8 & 16 \\ 4 & 0 & 0 & 0 & 8 \\ 0 & 8 & 0 & 0 & 4 \\ 4 & 16 & 8 & 4 & 0 \end{pmatrix} \quad C_2 = \begin{pmatrix} 0 & 0 & 4 & 0 & 4 \\ 8 & 0 & 0 & 0 & 16 \\ 0 & 0 & 0 & 0 & 8 \\ 0 & 4 & 0 & 0 & 0 \\ 8 & 8 & 0 & 4 & 0 \end{pmatrix}$$

It is worth noting that the nonsymmetrical character of matrix C for directed graphs may follow not only from the directional nature of arcs representing unidirectional network links but can also refer to different nominal capacities of links in each direction for a given pair of neighboring network nodes. For example, as given in graph G_2 from Fig. 3.2b, the nominal capacity of a link between network nodes 2 and 5 depends on the source/destination of that link and is defined as $c_{2,5} = 16$ and $c_{5,2} = 8$, respectively.

Another way to represent the interconnection of network nodes is via the node–link *incidence matrix* \mathcal{I} providing information on the neighborhood relation of network nodes and links. In this matrix, a given i -th row refers to network node i , while column m is associated with m -th network link. If a link with index m

incident to network node i exists, this is represented by the value of 1 assigned to an element in i -th row and m -th column of \mathcal{I} , 0 otherwise.

Example form of matrix \mathcal{I}_1 for graph G_1 from Fig. 3.2a based on the following assignment of indices m to graph G_1 edges:

$$\begin{array}{llll} e_{1,2} \rightarrow m = 1 & e_{1,3} \rightarrow m = 2 & e_{1,5} \rightarrow m = 3 & e_{2,4} \rightarrow m = 4 \\ e_{2,5} \rightarrow m = 5 & e_{3,5} \rightarrow m = 6 & e_{4,5} \rightarrow m = 7 & \end{array}$$

as well as the respective *weighted incidence matrix* $\hat{\mathcal{I}}$ is defined as follows:

$$\mathcal{I}_1 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{pmatrix} \quad \hat{\mathcal{I}}_1 = \begin{pmatrix} 4 & 4 & 4 & 0 & 0 & 0 & 0 \\ 4 & 0 & 0 & 8 & 16 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 0 & 8 & 0 & 0 & 4 \\ 0 & 0 & 4 & 0 & 16 & 8 & 4 \end{pmatrix}$$

For networks with directional links, values of elements in matrices \mathcal{I} and $\hat{\mathcal{I}}$ are positive when representing indices m of links directed from given network nodes i and negative for links directed to given nodes i .

Another important structure useful in evaluating the topological properties of networked systems is the *Laplacian matrix* \mathcal{L} . Its elements $\mathcal{L}[i, j]$ are defined as given in formula (3.1).

$$\mathcal{L}[i, j] = \begin{cases} d_i, & \text{if } i = j \\ -1, & \text{if } i \neq j \wedge v_i \text{ is adjacent to } v_j \\ 0, & \text{otherwise} \end{cases} \quad (3.1)$$

where d_i is the degree of vertex v_i being the number of its incident edges (arcs).

The elements $\mathcal{L}[i, i]$ located along the main diagonal of \mathcal{L} thus provide information on degrees of vertices v_i , while the other elements of \mathcal{L} store information about the adjacency property of vertices v_i and v_j . For example, for graph G_1 from Fig. 3.2a, the related Laplacian matrix \mathcal{L} is defined as follows:

$$\mathcal{L}_{G_1} = \begin{pmatrix} 3 & -1 & -1 & 0 & -1 \\ -1 & 3 & 0 & -1 & -1 \\ -1 & 0 & 2 & 0 & -1 \\ 0 & -1 & 0 & 2 & -1 \\ -1 & -1 & -1 & -1 & 4 \end{pmatrix}$$

An equivalent definition of \mathcal{L} to the one from formula (3.1) is provided in Chapter 5 of [37] as given in formula (3.2).

$$\mathcal{L} = \Delta - \mathcal{A} \quad (3.2)$$

where Δ is a diagonal matrix with elements $\delta_{i,j}$ defined as given in formula (3.3).

$$\delta_{i,j} = \begin{cases} d_i, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \quad (3.3)$$

The Laplacian matrix is used to derive specific metrics for graphs representing the architecture of networked systems (see, e.g., [47]).

3.2 Centrality Metrics for Evaluation of Resilience of Single System Elements

In this subsection, we focus on centrality metrics aimed at quantifying the topological importance of single elements in networked systems. In the related literature, a particular interest is often in analyzing the centrality aspect of system nodes. This is indeed justifiable since communication paths, commonly established as the shortest ones between any pair of end nodes (to reduce the end-to-end transmission delay), typically traverse such ‘‘central’’ nodes. However, this feature also magnifies the negative consequences of failures of such elements. Also, since central nodes switch large amounts of data, they often become targets of malicious human activities. Therefore, correctly identifying the level of centrality of system nodes is crucial in implementing adequate resilience mechanisms.

The most common metrics for the centrality of networked system nodes are based on the degree, betweenness, closeness, and eigenvector topological properties of these elements. Their current form follows from results of the analysis done in the area of social networks (and the related mutual impact of people in graphs of social connections) starting from 50s of the 20th century (see, e.g., the related works of Bavelas [3], Freeman [11], or Albert and Barabási [1]).

Betweenness Centrality

The primary purpose of the *betweenness centrality* (BC) metric defined for a given network node i by formula (3.4) [6, 42] is to reflect the frequency of its involvement in switching the data transmitted along the shortest paths between all possible pairs of end nodes in the system (i.e., acting as a transit node along the shortest paths).

$$bc_i = \sum_{p \neq q} \frac{sp_i(p, q)}{sp(p, q)} \quad (3.4)$$

where:

$sp_i(p, q)$ is the number of the shortest paths between nodes p and q (of the same minimal cost) traversing node i ;

$sp(p, q)$ is the number of the shortest paths between nodes p and q (of the same minimal cost).

Also, there exists a normalized version of betweenness centrality with the value of bc_i divided by the total number of pairs of vertices in G (except for vertex v_i), i.e., by $(|V|-1)(|V|-2)$. As discussed in [41], a formula similar to (3.4) can be provided for a given network link (i.e., *link betweenness centrality*) to reflect the importance of that link in making multi-hop connections possible.

Closeness Centrality

Closeness centrality (CC) has been formulated to reflect the distance of a given node i to all the other nodes in the system [6, 41]. Therefore, its evaluation is based on the analysis of the length of the shortest paths between a considered node i and all the other system nodes [41]. Its simplified definition provided, e.g., in [42] based on the analysis of the hop count (i.e., the number of path links of the shortest paths) is given by formula (3.5).

$$cc_i = \frac{1}{\sum_{j \in N \setminus \{i\}} h_{i,j}} \quad (3.5)$$

where $h_{i,j}$ is the number of hops for the shortest path between nodes i and j .

Based on formula (3.5), the higher the cc_i value for a given node i , the closer it is to all other nodes. This property can be useful, e.g., when choosing a location for system services, because services located in nodes characterized by high closeness centrality values are closer to end users and, therefore, easily accessible (due to low transmission delay values). An important observation is that nodes characterized by high closeness centrality values are also typically located close to other nodes of high closeness centrality [41].

A normalized version of formula (3.5) assumes multiplication of cc_i by $|V|-1$.

Degree Centrality

Degree centrality (DC) is considered as one of the simplest metrics for the importance of a network node. It is defined based on the degree of node i as the number of system nodes being direct neighbors of that node (i.e., connected by a direct link) [46]. Following [6], degree d_i of node i can be determined using the adjacency matrix \mathcal{A} as given in formula (3.6).

$$d_i = \sum_{j=1}^{|V|} \hat{a}_{i,j} \quad (3.6)$$

Therefore, the importance of node i measured by its degree centrality d_i grows linearly with the increase of its degree [41]. This property remains well in line

with the former observation that higher-degree system nodes (such as switches) commonly process larger data volumes. Also, in the case of failures of nodes characterized by high values of degree centrality, services provided to a large group of users are likely to become affected as well.

Concerning real architectures of networked systems, degree centrality values are often different for different nodes. Also, it is common that only a small subset of system nodes is characterized by high-degree centrality values.

It is worth mentioning that a normalized variant of node degree centrality also exists, where d_i is divided by the maximum possible degree of a node, i.e., by $|V|-1$.

Eigenvector Centrality

The purpose of the *eigenvector centrality* (EC) metric is to evaluate the influence of a given node in the network. Following [6], eigenvector centrality ec_i of node i is defined as the value of the i th element of the eigenvector referring to the largest eigenvalue λ_1 calculated for the adjacency matrix \mathcal{A} .

$$ec_i = \frac{1}{\lambda_1} \sum_{k=1}^{|V|} \hat{a}_{i,k} ec_k \quad (3.7)$$

Therefore, eigenvector centrality is another metric of the centrality of nodes, according to which a node should be considered an important one if it is a direct neighbor of another important node [41, 45]. Indeed, the value of ec_i reflects the number of direct, 2-hop, 3-hop (and so on) neighbors of node i [6].

For two example network topologies shown in Fig. 3.3, the respective normalized values of node centrality parameters are provided in Tables 3.1 and 3.2.

As can be seen in Tables 3.1 and 3.2, the values of node centrality metrics are generally consistent with each other, i.e., the highest value of one of them, say degree centrality (e.g., for node 7 for the NSF-14 network topology): $d_7 = 0.31$ in

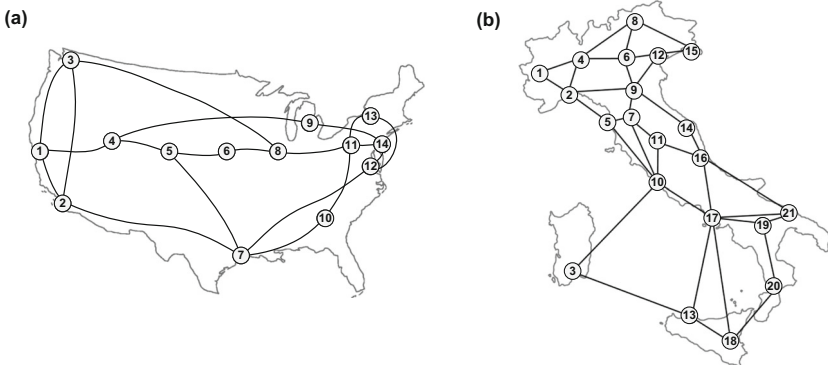


Fig. 3.3 Example topologies of (a) NSF-14 and (b) Italian-21 networks

Table 3.1 Values of normalized centrality parameters for the NSF-14 network nodes

Node index (i)	1	2	3	4	5	6	7	8	9	10	11	12	13	14
bc_i	0.06	0.08	0.07	0.11	0.12	0.03	0.24	0.15	0.06	0.03	0.19	0.10	0.01	0.11
cc_i	0.43	0.46	0.45	0.45	0.48	0.42	0.54	0.48	0.42	0.45	0.5	0.46	0.39	0.46
d_i	0.23	0.23	0.23	0.23	0.23	0.15	0.31	0.23	0.15	0.15	0.31	0.23	0.15	0.23
ec_i	0.29	0.32	0.29	0.24	0.26	0.18	0.37	0.27	0.17	0.23	0.32	0.28	0.20	0.26

Table 3.2 Values of normalized centrality parameters for the Italian-21 network nodes

Node index (i)	1	2	3	4	5	6	7	8	9	10	11	12	13	14
bc_i	0.00	0.17	0.03	0.05	0.15	0.08	0.16	0.02	0.33	0.31	0.02	0.08	0.02	0.14
cc_i	0.30	0.38	0.33	0.32	0.40	0.34	0.43	0.28	0.43	0.44	0.38	0.33	0.31	0.39
d_i	0.10	0.20	0.10	0.20	0.15	0.20	0.20	0.15	0.25	0.25	0.15	0.15	0.15	0.10
ec_i	0.10	0.20	0.15	0.16	0.23	0.18	0.30	0.11	0.26	0.36	0.25	0.14	0.20	0.15
Node index (i)	15	16	17	18	19	20	21							
bc_i	0.01	0.17	0.30	0.05	0.05	0.01	0.02							
cc_i	0.26	0.40	0.40	0.31	0.30	0.25	0.34							
d_i	0.10	0.20	0.30	0.15	0.15	0.10	0.15							
ec_i	0.07	0.28	0.39	0.19	0.20	0.10	0.23							

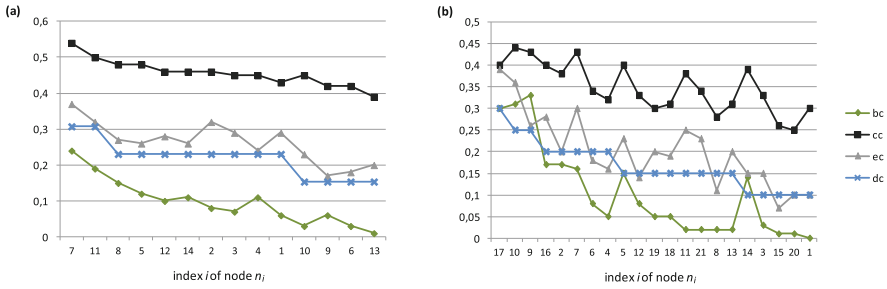


Fig. 3.4 Correlation of node centrality metrics for (a) NSF-14 and (b) Italian-21 network topologies

Table 3.1 also implies high values of the other three centrality metrics: $bc_7 = 0.24$, $cc_7 = 0.54$ and $ec_7 = 0.37$.

A detailed analysis of the correlation of node centrality coefficients is presented for both considered network topologies in Fig. 3.4, where nodes are sorted descending their degrees. A general observation from Fig. 3.4 is that, with the decrease of the network node degree, betweenness centrality values decrease the most rapidly among all considered centrality metrics. Therefore, to identify the central nodes that have the most remarkable contribution to end-to-end transmission, betweenness centrality, among all considered node centrality metrics, turns out to be the most proper one.

3.3 Structural Metrics for Evaluation of Resilience of Networked Systems Architectures

In this section, we highlight definitions and discuss the properties of the selected metrics applicable in the evaluation of the resilience of the entire structure of a networked system. Therefore, they are often referred to as structural metrics. Our analysis begins with metrics related to the degrees of network elements. Next, we consider metrics related to communication paths in the system. The last group of structural metrics analyzed in this section covers selected advanced aspects related to the topology of the networked system.

Average Node Degree

The *average node degree* (k) [41] is a simple measure of the density of the network topology. It provides information on the average number of links incident to a network node. Based on data stored in the adjacency matrix \mathcal{A} , this metric, here denoted by d_{avg} , can be calculated as given in formula (3.8).

$$d_{avg} = \frac{\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} \hat{a}_{i,j}}{|V|} \tag{3.8}$$

Coefficient d_{avg} takes values from 0 (in the case of a system consisting only of isolated nodes) to $|V|-1$ (in the case of a system characterized by a topology of a full graph representing the architecture in which each network node has direct links to all the other nodes).

Since node degree values provide information on the maximum number of disjoint communication paths sourced from/destined to a given node, they are crucial in resilient routing, as they impact the ability of a system to set up multiple disjoint paths. The lower bound on this ability for the entire system is indeed constrained by the minimal node degree in the considered system.

Minimal Node Degree

The *minimal node degree* (d_{min}) is the minimal value of degrees of nodes in the networked system.

$$d_{\text{min}} = \min_{i:v_i \in V} d_i \quad (3.9)$$

Indeed, to deploy a resilient routing scheme in the system involving k disjoint paths (for protection against a simultaneous failure of $k-1$ nodes), a necessary condition is that each network node i should be characterized by its degree of at least k , meaning that d_{min} of the entire networked system should be at least equal to k .

Heterogeneity

Heterogeneity has been introduced as a metric of inhomogeneity of node degrees. Following [41], it is defined as the standard deviation σ_{deg} of degrees of nodes in the system divided by the average node degree (d_{avg}), as given by formula (3.10).

$$h = \frac{\sigma_{\text{deg}}}{d_{\text{avg}}} \quad (3.10)$$

In general, the smaller the values of h (i.e., the closer they are to 0), the greater the homogeneity of the node degrees, and thus, the greater the robustness of the entire networked system architecture to failures of its elements.

Concerning the example topologies of NSF-14 and Italian-21 networks from Fig. 3.3, the related values of the average node degree, minimal node degree, and heterogeneity metrics are provided in Table 3.3.

In particular, the minimal value of node degree for both networks is equal to 2.00, which implies that for both networks, deploying resilient routing schemes for any pair of end nodes based on pairs of node-/link-disjoint paths may be possible.

Table 3.3 Values of structural metrics referring to degrees of network nodes for the example NSF-14 and Italian-21 network topologies from Fig. 3.3

Network	Average node degree	Minimal node degree	Heterogeneity
NSF-14	2.86	2.00	0.23
Italian-21	3.33	2.00	0.33

Topologies of systems with the minimal node degree of 2 are often called “two-connected.” However, neither of the considered NSF-14 and Italian-21 network topologies can utilize schemes based on sets of three (or more) disjoint paths for a pair of nodes, as the degrees of some nodes in these networks are only equal to 2.

Concerning the value of the heterogeneity metric, it is lower for the NSF-14 network topology, implying that the topology of that network is more regular (relative differences of node degree values are lower than for the topology of the Italian-21 network).

Average Shortest Path Length

Average shortest path length (l) coefficient [40] provides information on the average distance (or the number of links) along the shortest paths calculated considering all pairs of source and destination vertices v_s and v_t in G , as given in formula (3.11).

$$l = \sum_{s,t:v_s,v_t \in V} \frac{hc_{s,t}}{|V| \cdot (|V| - 1)} \quad (3.11)$$

where $hc_{s,t}$ is the number of links (hop count) in the shortest path between vertices v_s and v_t .

It is worth noting that the calculation of the number of links in the shortest path instead of their length in the Cartesian sense is often applied due to a common assumption of the unitary length of all links in the system or follows simply from the assumption to focus on the number of hops in the shortest path. Another observation is that formula (3.11) remains valid also for directed graphs, where the number of links of the shortest path from v_s and v_t can be different from that for a reverse path from v_t and v_s .

As the number of nodes and links traversed by the shortest path is correlated with the risk of path failure due to failures of system elements (see the analysis from Chapter 2 of this book), the average shortest path length metric is useful in the resilience context, especially in terms of determining the average resistance of communication paths in the system to failures.

Diameter

Diameter of a network [46] is commonly defined as the minimum hop count between the two most distant nodes in the system. Therefore, to calculate the diameter of a networked system, the numbers of links of the shortest paths between every pair of system nodes s and t (i.e., $hc_{s,t}$) need to be first calculated, and next, the maximum of these values should be returned as provided by formula (3.12).

$$l = \max_{s,t:v_s,v_t \in V} hc_{s,t} \quad (3.12)$$

Similar to the average shortest path length, diameter (being, in fact, the “maximum shortest path length”) can provide useful information about the related maximum risk of affection of a communication path in the system by failures of system elements.

Efficiency

Efficiency of a networked system focuses on the inverse values of the number of links of the shortest paths in the networked system. It can be, therefore, used to evaluate how quickly information can be transmitted between any pair of end nodes s and t in the system. Following [27, 41], it can be defined as the normalized sum of reciprocals of values of hop counts $hc_{s,t}$ for the shortest paths between all pairs of system nodes as given by formula (3.13).

$$\epsilon = \frac{\sum_{s,t:v_s,v_t \in V} \frac{1}{hc_{s,t}}}{|V| \cdot (|V| - 1)} \quad (3.13)$$

Therefore, the higher the value of ϵ , the shorter the communication paths are in the system, and, thus, the more efficient (i.e., faster) the delivery of information to the destination nodes, as well as the smaller the set of network elements traversed by a given path (i.e., the higher is the resilience of paths).

Vertex Connectivity

Following [41], *vertex connectivity*, $\kappa(G)$, is defined as the smallest number of vertices of graph G , the removal of which causes disconnection of system elements (i.e., partitioning of the system architecture into separated zones). As services are often provided by dedicated servers, such system partitioning (e.g., implied by failures due to many reasons discussed earlier in this book) might indeed bring severe consequences for many end users of not having access to these services.

Values of $\kappa(G)$ range from 1—as, e.g., in the case of network graphs being trees (see the example network topology in Fig. 3.5a) to $|V|-1$ for full graphs—see Fig. 3.5b. Therefore, $\kappa(G)$ can help assess the robustness of the system architecture to simultaneous failures of multiple network elements.

Concerning the topologies of two real networks analyzed earlier in this chapter, the related vertex connectivity $\kappa(G)$ is equal to 2 for both networks (see Fig. 3.6). A general observation following from the analysis of properties of different network graphs is that the more irregular the topology of a networked system, the lower the number of nodes, the removal of which partitions the system.

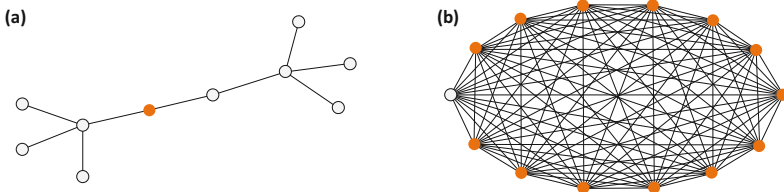


Fig. 3.5 Example network topologies characterized by vertex connectivity $\kappa(G)=1$ (graph (a)), and $\kappa(G)=|V|-1$ - graph (b) (the example subsets of vertices, the removal of which causes graph partitioning, are marked in orange)

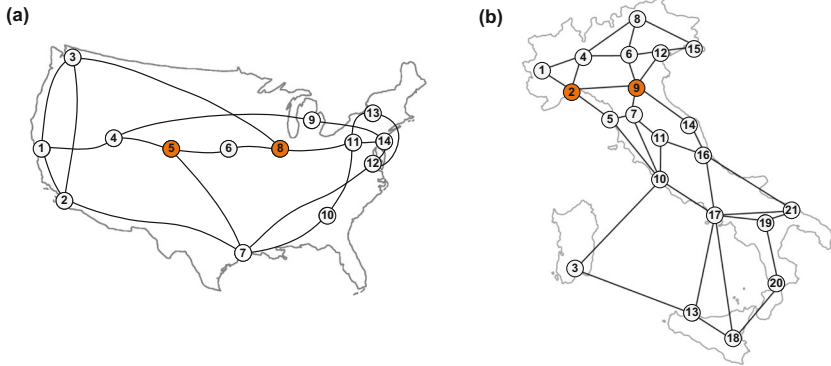


Fig. 3.6 Analysis of vertex connectivity $\kappa(G)$ for NSF-14 and Italian-21 network topologies (the example subsets of vertices, the removal of which causes graph partitioning, are marked in orange)

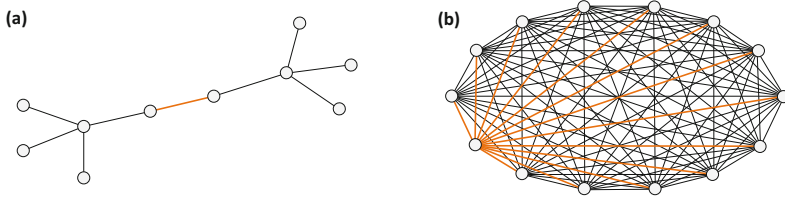


Fig. 3.7 Example network topologies characterized by edge connectivity $\lambda(G)=1$ (graph (a)), and $\lambda(G)=|V|-1$ - graph (b) (the example subsets of edges, removal of which causes graph partitioning, are marked in orange)

Edge Connectivity

Edge connectivity— $\lambda(G)$ —is defined similarly to vertex connectivity as the smallest number of edges from G whose removal leads to system partitioning. Similar to vertex connectivity, edge connectivity values range between 1 (for tree graphs) and $|V|-1$ for full graphs, as illustrated in Fig. 3.7. As simultaneous failures of multiple links of the system can also take place (e.g., due to fires causing the burning of optical wired cables or cuts of links during dig-ups carried jointly in the same duct), $\lambda(G)$ provides valuable information on the resistance of the system architecture in such scenarios.

As illustrated in Fig. 3.8, for the NSF-14 topology and Italian-21 topology, $\lambda(G)$ equals 2. Similar to vertex connectivity, edge connectivity is generally higher for regular topologies and lower for topologies characterized by higher heterogeneity values.

Graph Diversity

According to [39, 41], *graph diversity* is a metric of the frequency of traversing the same communication links and transit nodes by communication paths between given pairs of end nodes s and t . This metric is defined concerning all possible pairs

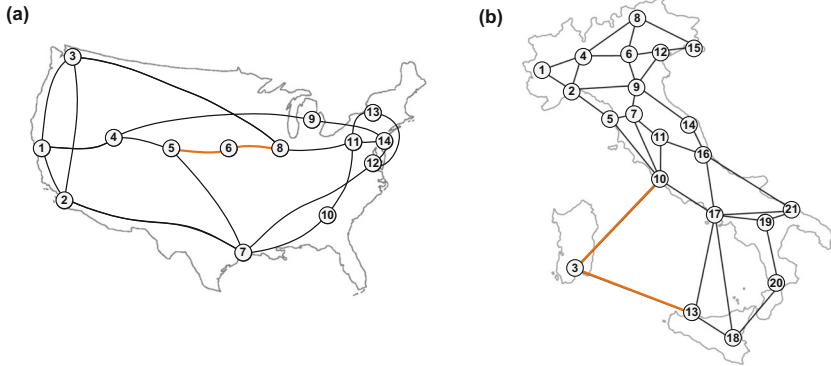


Fig. 3.8 Illustration of edge connectivity $\lambda(G)$ for NSF-14 and Italian-21 network topologies (the example subsets of edges, the removal of which causes graph partitioning, are marked in orange)

of end nodes s and t in the system and is based on values of the effective path diversity, each for a given pair of end nodes. In turn, the values of the effective path diversity follow from the values *path diversity* provided for paths P_i in the context of the respective shortest path P_0 (shortest in terms of the hop count).

For a given pair of end nodes, s and t , the related path diversity metric for a given arbitrary path P_i is defined in relation to the shortest path P_0 between these end nodes as given in formula (3.14).

$$D(P_i) = 1 - \frac{|P_i| \cap |P_0|}{|P_0|} \quad (3.14)$$

where $|P|$ denotes the number of links and transit nodes used by path P .

Therefore, $D(P_i)$ changes from 1 (if paths P_i and P_0 do not share any elements except for the end nodes) to 0 (if paths P_i and P_0 are identical, i.e., traverse the same set of links).

Following [39], the effective path diversity can be determined as an aggregation of path diversities for a selected set of paths between a given pair s and t of end nodes. Finally, the value of the graph diversity metric can be calculated as the average of all effective path diversity values determined for all pairs of end nodes.

Higher values of graph diversity indicate a greater level of system robustness.

Number of Spanning Trees

This metric calculates the total number of distinct spanning trees (i.e., trees that include all nodes of the networked system) that exist for a given network graph [25, 41].

In general, the analysis of the number of spanning trees can provide useful information on the ability of a system to switch to another configuration (i.e., based on another spanning tree) in scenarios of network element failures. This can help restore affected services quickly (see, e.g., the scheme proposed in [24]).

Average Two-Terminal Reliability

The *average two-terminal reliability* ($ATTR$) provides information on the probability that a randomly chosen pair of nodes s and t is connected, meaning a communication path exists between them in the network graph. Following [41], it is defined as the total number of pairs of nodes in all system components of the system divided by the total number of node pairs in the system. Therefore, for fully connected systems (see, e.g., Fig. 3.3), the value of $ATTR$ is equal to 1. Otherwise, in the case of systems partitioned into several separate components, the value of $ATTR$ belongs to the $(0,1)$ range.

For example, for the topology shown in Fig. 3.9a, $ATTR_{G_a} = 111/231 \approx 0.48$. This follows from the fact that topology from Fig. 3.9a consists of two separate components: the upper one with ten nodes and the lower one with 12 nodes. Therefore, the number of connected node pairs is equal to $10 \cdot 9/2$ (the upper part) + $12 \cdot 11/2$ (the lower part) = $45 + 66 = 111$, while the total number of node pairs is $22 \cdot 21/2 = 231$.

The topology from Fig. 3.9b also consists of two separate components. However, one of them is significantly smaller than the second one. They consist of 3 and 12 nodes, respectively. The number of connected node pairs is equal to $3 \cdot 2/2$ (the upper part) + $12 \cdot 11/2$ (the lower part) = $3 + 66 = 69$, while the total number of node pairs is $15 \cdot 14/2 = 105$. The value of $ATTR_{G_b}$ is, therefore, equal to $69/105 \approx 0.66$, which is higher than $ATTR_{G_a}$.

Relative Size of the Largest Connected Component

The *relative size of the largest connected component* ($rLCC$) metric is defined as the ratio of the number of nodes of the largest connected cluster of the system and the total number of system nodes [6].

Values of $rLCC$ metric are generally positively correlated with $ATTR$ values. For the example topologies from Fig. 3.9 with the related $ATTR$ values: $ATTR_{G_a} \approx 0.48$ and $ATTR_{G_b} \approx 0.66$, the related values of $rLCC$ are: $rLCC_{G_a} = 12/22 \approx 0.55$ and $rLCC_{G_b} = 12/15 = 0.80$.

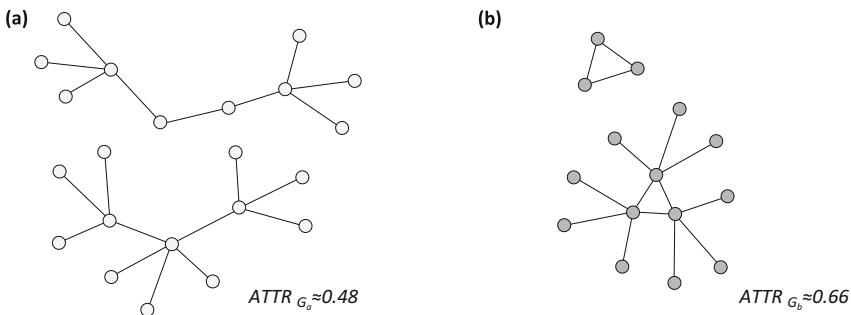


Fig. 3.9 Examples of topologies of two systems to illustrate $ATTR$ properties

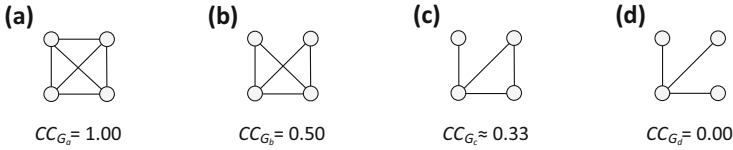


Fig. 3.10 Examples of four topologies for illustration of CC properties

Clustering Coefficient

The *clustering coefficient* (CC) has been proposed to evaluate the scale of cluster formation by nodes in the system topology [44]. This follows from a general observation that in the case of numerous real-world network topologies, system nodes frequently form tightly connected subsets (i.e., with either direct links or very short paths between node pairs in such groups).

The clustering coefficient for the system topology is evaluated based on the identification of triplets of nodes, i.e., groups of three nodes with direct links between them. Triplets can be either “open,” i.e., formed by three vertices connected by two edges, or “closed,” i.e., with three vertices connected by three edges. Three closed triplets, each centered at a different node, form a triangle.

The clustering coefficient is defined for a system topology as the ratio of the number of closed triplets over the total number of open and closed triplets [26]. Therefore, the cc parameter values range from 0 to 1. Fig. 3.10 presents example four topologies with the respective values of the clustering coefficient.

Concerning the topologies of real-world networked systems analyzed in this chapter, the clustering coefficients of the NSF-14 and Italian-21 topologies from Fig. 3.3 are equal to 0.071 and 0.278, respectively.

3.4 Reasons for Diverse Characteristics of System Elements

The structure of networked systems naturally evolves over time. This, in particular, means:

- Replacement of system nodes such as computing and storage nodes, communication links, and network nodes including, e.g., switches, routers, etc., by elements characterized by higher performance. Concerning network nodes, it is essential to mention that the new ones are commonly characterized by more communication ports than the ones being replaced.
- Addition of new elements to the system, increasing the size of the system and, therefore, raising its complexity.

It is also essential to notice that when adding a new element to the system (say, a network node), it is natural to link it to the existing ones characterized by many communication ports and, generally, by higher performance. By linking new elements to high-performance core nodes, one can thus fully benefit from the nominal capabilities of new elements not being bottlenecked by their neighbors' limitations.

When analyzing the related evolution of the system topology graph, we can equivalently say that when adding new vertices to the graph, it is more probable to link a new vertex with an existing one of high rather than low degree. This, in turn, forms the basis of the *preferential attachment rule* provided by Barabási and Albert in [2] defining the dependency between the probability $\Pi(v_i)$ that the existing node represented by vertex v_i in the topology graph will be linked to a new node as given in formula (3.15).

$$\Pi(v_i) = \frac{d_i}{\sum_j d_j} \quad (3.15)$$

As illustrated in Fig. 3.11, such an uncontrolled growth of the structure of a networked system can lead to system topologies being highly irregular, i.e., characterized by a high diversity of node degrees, often taking the power law asymptotic form characteristic to the so-called *scale-free networks* [2].

As can be seen in Fig. 3.11d, for nodes of high degree (the so-called *central nodes*—e.g., nodes 2 and 4), the distance to other high-degree nodes is often small, which follows from another property of scale-free structures, according to which vertices tend to cluster together in groups.

As discussed in Sect. 3.2 in this chapter, high-degree nodes often serve a significant share of network traffic. This is due to their high performance and their “central” location in the system topology. Therefore, their potential failure may lead to severe consequences for many end users, which, in turn, magnifies the risk of possible malicious activities aimed at such elements and raises the need for even more advanced protection mechanisms.

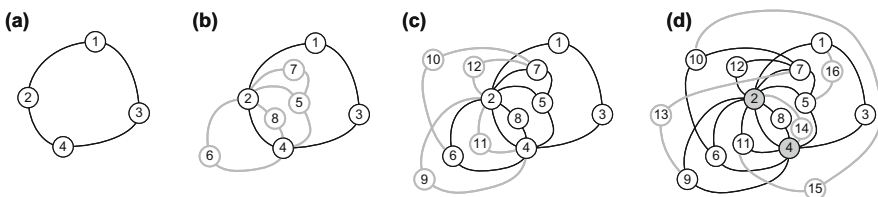


Fig. 3.11 Example illustration of the growth of a system topology following the preferential attachment rule

3.5 Functional Metrics for Networked Systems Resilience Evaluation

In this section, we highlight the selected metrics aimed at evaluating the performance of a networked system both in the case of the correct functioning of all system elements and in scenarios of failures. These metrics can be generally divided into network-level (i.e., focusing on the performance of system elements and multi-hop communication paths), packet-level (i.e., addressing the QoS features related to the transmission of packets), and subjective (i.e., designed to evaluate the performance of system services perceived by end users).

3.5.1 Network-Level Metrics

We start by highlighting the basic metrics for the operation of network elements referring to their involvement in multi-hop transmission. Next, we focus on the performance-related characteristics referring to communication paths, which are related mainly to the overall transmission delay and the probability of a successful setup of communication paths. Finally, we elaborate on network-level performance metrics for more advanced transmission configurations like anycast. We conclude this part by focusing on a selected complex metric aggregating the properties of a set of other metrics.

Link Utilization

Link utilization metric provides information on the percentage of the total (i.e., nominal) capacity used for data transmission [7]. It can refer to either the fraction of link capacity reserved in advance for serving all flows passing through that link (as in the case of allocation of channels of wired links in optical transport networks) or to the instant usage (at time t) of link resources in packet-switched systems.

Node Load

Following [32], *node load* metric has been proposed to measure node importance in overlay networks. It provides information on the number of overlay links passing through a given physical node. The higher the value of node load, the more overlay links get affected due to a failure of that physical node.

Path Symmetry

Following [23, 32], *path symmetry* (PSY) aims to measure the symmetry of paths between source and destination nodes s and t . It focuses on analyzing the end-to-end latency (expressed by the round trip time) and the hop count for the related forwarding and reverse paths, as given in formula (3.16).

$$PSY = \frac{hc}{hc'} \cdot \frac{RTT_{min}}{RTT'_{min}} \quad (3.16)$$

where hc and RTT_{min} denote the hop count and the lowest round trip time for packets concerning the forwarding path, while hc' and RTT'_{min} have the same meaning for the reverse path.

In the ideal case (i.e., when both paths are entirely symmetric), $PSY=1$. Otherwise, $PSY < 1$ denotes a longer reverse path, while $PSY > 1$ implies a longer forwarding path.

For the example configuration of two paths (i.e., forwarding and reverse), as illustrated in Fig. 3.12, we have $hc=3$ and $hc'=5$. Assuming that $RTT_{min}= 75ms$ while $RTT'_{min} = 100ms$, the value of PSY is equal to $(3/5) \cdot (75/100)=0.45$.

Link Stress

The *link stress* metric helps evaluate the efficiency of overlay networks, as it calculates the number of times packets traverse the same physical link [32].

Relative Delay Penalty/Stretch

Relative delay penalty/stretch is another measure for evaluating the efficiency of overlay networks. It is defined as the time needed for a packet to be transmitted end-to-end (from node s to node t) via the overlay path consisting of overlay links divided by the time needed when transmitting this packet between the same pair of end nodes, however, measured directly in the underlying transport network [32].

For example, as illustrated in Fig. 3.13, a path in the overlay network between nodes B and H is provided by three virtual links: (B,C), (C,F) and (F,H). In particular:

- Virtual link (B,C) is established in the physical network via path (B,b,d,c,C) of the total delay equal to 79.
- Virtual link (C,F) is established in the physical network via path (C,c,d,e,f,F) of the total delay equal to 114.
- Virtual link (F,H) is established in the physical network via path (F,f,h,H) of the total delay equal to 49.

Therefore, the overall transmission delay between nodes B and H in such a configuration equals $79 + 114 + 49 = 242$. However, a path established between

Fig. 3.12 Example different forward and reverse paths implying the value of PSY different than 1

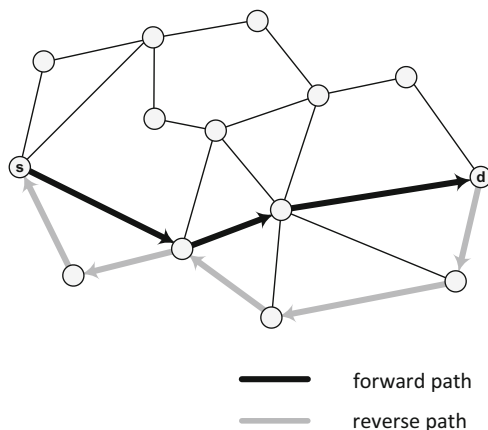
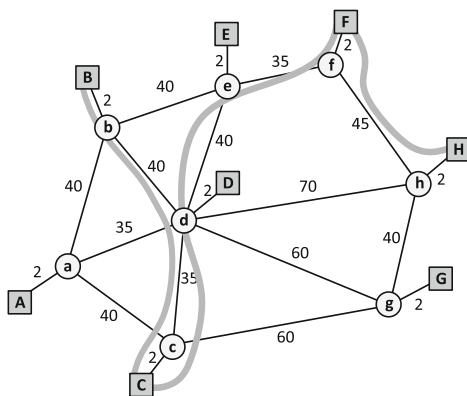


Fig. 3.13 Example of the overlay network (values next to links denote their nominal delay)



nodes B and H directly in the physical network without any intermediate forwarding in the overlay network would be (B,b,d,h,H), and its overall delay would be 114. Therefore, the relative delay penalty (stretch) of such a configuration equals $242/114 \approx 2.12$, meaning that the overlay configuration is at least twice as costly as the original one involving transit processing only at the physical network.

Quantitative Robustness

The *quantitative robustness* metric (*QNRM*) is proposed in [28] to evaluate the efficiency in establishing connections in a given time step t as the fraction of the number of established connections to the total number of connections that should have been established at time step t . For longer intervals of interest, the respective average value of QNRM over all consecutive time steps t_i should be determined.

Qualitative Robustness

The *qualitative robustness* metric (*QLRM*) is introduced in [28] to determine the variation of QoS parameters for a broad range of occurrences of impairments (including random attacks, targeted attacks, dynamic epidemical failures, and dynamic periodical failures). It is focused on the analysis of the average shortest path length (ASPL) and is defined as the quotient of the standard deviation of ASPL and ASPL itself divided by the analogous quotient obtained in the scenario of occurrence of a given impairment.

Average Content Accessibility

The metric of the *average content accessibility* (*ACA*) is proposed to evaluate the possibility of delivering the anycast traffic in scenarios of massive failures implied by disaster events [33]. Generally, this feature is associated with the design problem of locating replica servers in a way that allows the end users to receive information from at least one replica server in post-disaster periods.

Mean Content Accessibility

As discussed in [32, 33], the *mean content accessibility* (μ -ACA) is designed to evaluate the robustness of the networked system concerning the delivery of anycast traffic by taking into consideration a broad range of disasters. Therefore, it can be viewed as an extension of the average content accessibility metric.

R-value

Following [29], the *R-value* metric is defined as the weighted average of values of n other metrics of robustness, as given in formula (3.17).

$$R = \sum_{k=1}^n s_k \cdot t_k \quad (3.17)$$

where s_k and t_k denote the weight and the value of k -th metric, respectively ($\sum_{k=1}^n s_k = 1$; $t_k \in [0, 1]$).

3.5.2 Packet-Level Metrics

Packet-level metrics are useful in measuring the quality of transmission in packet-switched networks. This set of metrics focuses mainly on the aspects of *quality of service (QoS)* defined in [18] as the “totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service.” The set of major QoS characteristics comprises subjective parameters such as packet loss, bit rate, throughput, transmission delay, and jitter analyzed in scenarios of normal network operation and in post-failure periods when ensuring the assumed level of service evaluated by these metrics can be particularly difficult. In particular, concerning the failure scenarios, the following metrics are essential.

Propagation Time over a Link

Propagation time over a link is a metric of time for a packet necessary to travel via the considered link [32].

Latency/End-to-End Delay

The *latency (end-to-end delay)* metric [7] is used to determine the total propagation time for a message to travel via all consecutive links of the transmission path between the source and destination nodes s and t (i.e., the sum of the propagation time values over all consecutive links of a path).

Jitter

Jitter is a metric of the variation of latency likely to occur, e.g., due to changes in queueing/switching time at network nodes due to fluctuations in traffic intensity [32].

Packet Loss Ratio

Packet loss ratio metric is used to measure the fraction of packets that are not received correctly (i.e., received with errors or not received) at the destination node divided by the total number of transmitted packets in a given observation time window [32].

Retransmission Rate

The *retransmission rate* metric is used to evaluate the ratio of retransmitted packets over the total number of transmitted packets in a given observation time window for a certain pair of end nodes of transmission.

Throughput

The *throughput* metric provides information on the nominal message delivery rate via a given link. Depending on the environmental (propagation) properties (e.g., of wireless links) varying over time, throughput value is prone to fluctuations [35].

Generally, the values of almost all the above metrics will likely deteriorate in post-failure periods. This refers particularly to the increase of values of metrics such as latency, jitter, packet loss ratio, and retransmission rate, which, in fact, reflects difficulties of the networked system in failure scenarios (e.g., due to longer transmission paths and worse propagation characteristics). In the case of the last parameter (i.e., throughput), its deterioration denotes a decrease of its value, e.g., under adverse weather conditions such as dense fog in free-space optical networks—FSO [22].

3.5.3 Subjective Metrics

Subjective metrics are used to evaluate the performance of system services perceived by end users. Therefore, they help assess *quality of experience (QoE)*, being largely subjective and integrating user perception, experience, and expectations [10]. In particular, the most comprehensive definition of QoE seems to be the one from the ITU-T P.10/G.100 recommendation, where QoE is described as “the degree of delight or annoyance of the user of an application or service” [19]. A detailed set of QoE-related definitions can be found, e.g., in [5].

While QoE ratings are certainly user-centric (i.e., referring to the needs and expectations of end users), they much depend on QoS characteristics (referring to the ability of a networked system to provide its services at a certain quality level defined by QoS parameters such as packet loss, bit rate, throughput, transmission delay, and jitter). In particular, the authors of [10] show that the dependency of QoE on QoS can be considered exponential.

QoS attributes are often regarded as network-centric and largely represent the interests of network operators/service providers. These two aspects, i.e., the viewpoint of users interested in as good service as possible and of network operators/service providers (often focusing on a minimal level of investments assuring the assumed level of QoS), can be seen as opposing.

A relatively rich set of research results on methods of ensuring and measuring the QoE is available in the literature. Among them, particularly noteworthy seem to be the ones by Hossfeld et al. (see, e.g. [15, 17, 21]). In this section, a selected set of subjective metrics is highlighted, and their usability in failure scenarios is discussed.

Mean Opinion Score

The *mean opinion score (MOS)* metric has been designed to evaluate the perceived quality of experience. It is based on subjective evaluations of users [10]. Following [20], users assign scores based on the following scale: 5-excellent, 4-good, 3-fair, 2-poor, 1-bad. In the final processing of results, the value of MOS is obtained as the arithmetic mean of user opinions. MOS is commonly considered as a standard metric for QoE [43].

Standard Deviation of Opinion Scores

Since assessing the level of QoE by end users might largely be sensitive (e.g., in the case of difficulties in making a clear assessment of QoE by particular users), relying only on the average values user experience of MOS may not be sufficient. Unfortunately, providing only the average values reflected by MOS hides the level of variation in ratings and thus provides, at most, partial information about user experience. It is, therefore, necessary to extend the analysis at least by the evaluation of the diversity of user opinions provided by the standard deviation of MOS focusing on the level of rating diversity, referred to as *standard deviation of opinion scores* (shortly *SOS*), as proposed in [16].

Other Statistical Metrics for the Evaluation of QoE

Since relying only on the mean values of user opinions in the evaluation of QoE is often not adequate, apart from the SOS metric of the distribution of user scores, one can also focus on distributions of user ratings (for comprehensive information about user ratings), entropy (referring to the level of unpredictability of user scores and the uncertainty of the measurement system), or on more detailed ratings coming from fractions of satisfied and dissatisfied users, as proposed in [13], as well as on estimating the confidence intervals for MOS values, as considered in [14].

In post-failure periods, it is naturally more challenging to fulfill obligations concerning the assumed level of QoS. Therefore, there is also a risk of deterioration of QoE perceived by the end users following the related degradation of QoS parameters. Investing in resilience mechanisms supporting communications in failure scenarios is thus an essential aspect of maintaining the QoE level in line with user expectations.

3.6 Selected Examples for Adaptation of Metrics in Networked Systems

Metrics discussed in this chapter are often used in practice. They are commonly applied in evaluating the performance of networked systems and their components. However, it is also worth emphasizing the vital role of some of them, e.g., in the operation of routing protocols (in particular, during the calculation of multi-hop paths characterized by the lowest cost according to a given metric). Another essential utilization of metrics refers to various optimization tasks concerning the design of resilient architectures networked systems, e.g., designing a system

structure or determining the location of crucial components of such systems such as computing nodes or data servers. As the literature provides a large set of examples for the application of metrics, in this section, we will present selected ones that are particularly useful for routing mechanisms and methods of networked systems design.

Concerning the utilization of metrics in routing protocols, metrics relating to the characteristics of communication links are usually used. For instance, a classical *Routing Information Protocol (RIP)* [12] belonging to the distance-vector class of routing algorithms uses the hop count metric to determine the end-to-end paths characterized by the lowest cost expressed by the number of links traversed by these paths. As a result of these calculations, for each determined path, information about the related next-hop node is stored by each network node traversed by that path. Paths are recalculated periodically to respond to changes in network topology (for instance, as a result of a failure of a link or node). RIP is proper for medium-sized systems that are composed of relatively homogeneous equipment (e.g., identical nodes characterized by comparable node processing times and links of the same transmission rate). Then, assuming a comparable length of links in the system, the overall cost of a path can be, thus, well reflected by the number of path hops.

In the case of nonhomogeneous networks (i.e., consisting of nodes from different vendors characterized by differentiated times for packet processing and links of differentiated transmission rates), metrics other than the basic hop count are better suited to reflect the total path cost. For instance, in *Open Shortest Path First (OSPF)* [31], each link is associated with a cost metric which by default is assumed to be inversely proportional to the bandwidth of that link (i.e., network-level functional characteristics). In this protocol, belonging to the class of link-state protocols, each network node is aware of the state (up/down) of each link as well as the associated cost metric and calculates the cheapest communication paths based on the related transmission rates of network links, using Dijkstra's algorithm [8]. In path computations, high-speed links are thus preferred, which, in turn, reduces the overall transmission delay along multi-hop paths. However, as path computations in OSPF are CPU- and memory-intensive, practical utilization of OSPF is limited to medium-sized networks.

Another example of utilization of a network-level functional metric in routing is provided in [36], including a proposal of a routing algorithm using the node load metric to calculate the multi-hop paths. Since, for every demand to establish a multi-hop path, it selects the path associated with the least loaded nodes, an additional feature is that, in the long run, it also leads to balancing the load of nodes.

Metrics referring to node centrality characteristics are used in the literature both in the case of routing algorithms and in network design methods, e.g., to solve various service placement problems. For example, in [34], a routing algorithm is introduced for anycast communications using a metric based on a mix of node degree and hop count. In this algorithm, packets are forwarded by each transit node to the next hop, characterized by a greater number of alternate paths available.

As discussed earlier in this chapter, nodes characterized by high values of centrality metrics often switch large amounts of data (as the shortest paths often

traverse them), as well as are good candidates for placement of certain services (due to low transmission delay from these nodes to other nodes in the system). Therefore, they are also common targets of malicious activities. As presented, e.g., in [38], the availability of communication paths, as well as of certain services, can be improved in scenarios of malicious attacks by using node centrality metrics to determine locations (placement) of services at low-degree nodes (i.e., characterized by a low risk of an attack), as well as by applying a routing scheme with link cost metric determined based on the average values of centrality metrics of its end nodes.

The properties of specific communication environments often call for a metric adjusted to a particular communication scenario. There are many proposals in this context in the related literature. For example, concerning wireless environments, reference [9] focuses on the use of packet-related characteristics: the *round trip time*—*RTT* (i.e., the round trip delay for unicast probes between neighboring nodes) and the *expected transmission count*—*EXT* (referring to the loss rate of packets between neighboring nodes) as a metric for wireless links used for routing purposes. Indeed, collisions of packets transmitted in parallel by different sources over the wireless medium (justifying the need to use certain transmission protocols such as CSMA/CA) [4], as well as other reasons for packet retransmissions (e.g., due to transmission errors) are reasonable justifications for focusing on the instant characteristics of wireless links performance.

3.7 Summary

The analysis of the properties of metrics provided in this chapter confirms their essential role in the correct functioning of networked systems in normal operating conditions and during periods of failures. In a normal operational state, these metrics can deliver valuable information about system functioning, potential disproportions concerning the network load, etc. Also, they can indicate areas of the system where the effects of possible failures would be particularly severe and thus provide valuable information useful for system design, configuration, and update.

? Questions

1. Characterize the ways of formal representation of the architecture of networked systems.
2. Describe and compare the metrics of node centrality.
3. Discuss the features and the purpose of structural metrics in evaluating networked systems resilience.
4. Describe and compare the structural metrics referring to node degrees.
5. Describe and compare the structural metrics referring to communication paths.
6. Describe and compare the structural metrics referring to the system connectivity.
7. Explain the reasons for the irregularities of the system topology.

8. Characterize the functional network-level metrics for evaluation of networked systems resilience.
 9. Characterize the functional packet-level metrics for evaluation of networked systems resilience.
 10. Explain the role and characteristics of subjective metrics of system performance evaluation.
 11. Provide several examples of utilization of node- and link-related metrics in practice.
-

References

1. Albert, R., Barabási, A.-L.: Statistical mechanics of complex networks. *Rev. Mod. Phys.* **74**(1), 47–97 (2002)
2. Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. *Science* **286**, 509–512 (1999)
3. Bavelas, A.: A mathematical model for group structure. *Hum. Organ. Appl. Anthropol.* **7**(3), 16–30 (1948)
4. Bianchi, G.: Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE J. Sel. Areas Commun.* **18**(3), 535–547 (2000)
5. Brunnstrom, K., Beker, S.A., de Moor, K., Dooms, A., Egger, S. et al.: Qualinet white paper on definitions of Quality of Experience (2013). <https://hal.science/hal-04638470v1>. Accessed 11 Sep 2023
6. Cetinay, H., Mas-Machuca, C., Marzo, J.L., Kooij, R., Van Mieghem, P.: Comparing destructive strategies for attacking networks. In: Rak, J., Hutchison, D. (eds.) *Guide to Disaster-Resilient Communication Networks*, pp. 117–140. Springer, Berlin (2020)
7. Chu, Y., Rao, S.G., Seshan, S., Zhang, H.: A case for end system multicast. *IEEE J. Sel. Areas Commun.* **20**(8), 1456–1471 (2002)
8. Dijkstra, E.: A note on two problems in connexion with graphs, *Numerische Mathematik* **1**, 269–271 (1959)
9. Draves, R., Padhye, J., Zill, B.: Comparison of routing metrics for static multi-hop wireless networks, *ACM SIGCOMM Comput. Commun. Rev.* **34**(4), 133–144 (2004)
10. Fiedler, M., Hossfeld, T., Tran-Gia, P.: A generic quantitative relationship between quality of experience and quality of service. *IEEE Netw.* **24**(2), 36–41 (2010)
11. Freeman, Linton C. A set of measures of centrality based on betweenness. *Sociometry* **40**(1), 35–41 (1977)
12. Hedrick, C.: Routing Information Protocol, Request for Comments (RFC) 1058, IET. <https://datatracker.ietf.org/doc/html/rfc1058>. Accessed 30 Sept 2023
13. Hossfeld, T., Heegaard, P.E., Varela, M.: QoE beyond the MOS: Added value using quantiles and distributions. In: *Proceedings of the 2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX'15)*, pp. 1–6 (2015)
14. Hossfeld, T., Heegaard, P.E., Varela, M., Skorin-Kapov, L.: Confidence interval estimators for MOS values ((2018)). arXiv:1806.01126 . <https://arxiv.org/abs/1806.01126>. Accessed 14 Sept 2023
15. Hossfeld, T., Keimel, Ch., Hirth, M., Gardlo, B., Habigt, J., Diepold, K., Tran-Gia, P.: Best practices for QoE crowdtesting: QoE assessment with crowdsourcing. *IEEE Trans. Multimedia* **16**(2), 541–558 (2014)

16. Hossfeld, T., Schatz, R., Egger, S.: SOS: The MOS is not enough! In: Proceedings of the 2011 Third International Workshop on Quality of Multimedia Experience, pp. 131–136 (2011)
17. Hossfeld, T., Schatz, R., Varela, M., Timmerer, C.: Challenges of QoE management for cloud applications. *IEEE Commun. Mag.* **50**(4), 28–36 (2012)
18. ITU-T: Recommendation E.800 – Definitions of terms related to quality of service (2008). <https://www.itu.int/rec/T-REC-E.800>. Accessed 11 Sept 2023
19. ITU-T Recommendation P.10/G.100: vocabulary for performance and quality of service. Amendment 5 (2016). <https://www.itu.int/rec/T-REC-P.10-201607-S!Amd5/en>. Accessed 11 Sept 2023
20. ITU-T: Recommendation P.800 – Methods for Subjective Determination of Transmission Quality. <https://www.itu.int/rec/T-REC-P.800-199608-1>. Accessed 11 Sept 2023
21. Jarschel, M., Schlosser, D., Scheuring, S., Hossfeld, T.: An evaluation of QoE in cloud gaming based on subjective tests. In: Proceedings of the 2011 5th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 330–335 (2011)
22. Kalesnikau, I., Pioro, M., Rak, J., Ivanov, H., Fitzgerald, E., Leitgeb, E.: Enhancing resilience of FSO networks to adverse weather conditions. *IEEE Access* **9**, 123541–123565 (2021)
23. Lareida, A., Meier, D., Bocek, T., Stiller, B.: Towards path quality metrics for overlay networks. In: Proceedings of the 2016 IEEE 41st Conference on Local Computer Networks (LCN'16), pp. 156–159 (2016)
24. Lee, S.S.W., Li, K.-Y., Lin, Ch.-Ch.: Modeling and algorithm for multiple spanning tree provisioning in resilient and load balanced Ethernet networks. *Math. Probl. Eng.* **2015**, 676542 (2015)
25. Li, J., Chee Shiu, W., Chang, A.: The number of spanning trees of a graph. *Appl. Math. Lett.* **23**(3), 286–290 (2010)
26. Luce, R.D., Perry, A.D.: A method of matrix analysis of group structure. *Psychometrika* **14**(1), 95–116 (1949)
27. Maniatakis, D., Balmprakakis, A., Varoutas, D.: On the temporal evolution of backbone topological robustness. In: Proceedings of the 2013 18th European Conference on Network and Optical Communications & 2013 8th Conference on Optical Cabling and Infrastructure (NOC-OC&I'13), pp. 129–136 (2013)
28. Manzano, M., Calle, E., Harle, D.: Quantitative and qualitative network robustness analysis under different multiple failure scenarios. In: Proceedings of the 2011 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT'11), Budapest, Hungary, pp. 1–7 (2011)
29. Manzano, M., Sahneh, F., Scoglio, C., Calle, E., Marzo, J.L.: Robustness surfaces of complex networks. *Nat. Sci. Rep.* **4**, 6133 (2014)
30. Moeller, S.: *Quality of Telephone-Based Spoken Dialogue Systems*. Springer, New York (2005)
31. Moy, J.: OSPF Version 2, Request for Comments (RFC) 2328, IETF. <https://datatracker.ietf.org/doc/html/rfc2328>. Accessed 30 Sept 2023
32. Natalino, C., Ristov, S., Wosinska, L., Furdek, M.: Functional metrics to evaluate network vulnerability to disasters. In: Rak, J., Hutchison, D. (eds.) *Guide to Disaster-Resilient Communication Networks*, pp. 47–62. Springer, Berlin (2020)
33. Natalino, C., Yayimli, A., Wosinska, L., Furdek, M.: Content accessibility in optical cloud networks under targeted link cuts. In: Proceedings of the 2017 International Conference on Optical Network Design and Modeling (ONDM'17), pp. 1–6 (2017)
34. Ohta, S., Makita, H.: Anycast routing based on the node degree for ad hoc and sensor networks. In: Proceedings of the 2013 IEEE 16th International Conference on Computational Science and Engineering, pp. 439–446 (2013)
35. Pyo, C.W., Harada, H.: Throughput analysis and improvement of hybrid multiple access in IEEE 802.15.3c mm-wave WPAN. *IEEE J. Sel. Areas Commun.* **27**(8), 1414–1424 (2009)
36. Qi, Z., Sun, J., Li, W.: A routing algorithm based loading ratio in nodes. In: Proceedings of the Canadian Conference on Electrical and Computer Engineering 2004, pp. 595–598 (2004)

37. Rak, J., Hutchison, D. (eds.): *Guide to Disaster-Resilient Communication Networks*. Springer, Berlin (2020)
38. Rak, J., Walkowiak, K. Reliable anycast and unicast routing: Protection against attacks. *Telecommun. Syst.* **52**, 889–906 (2013)
39. Rohrer, J.P., Jabbar, A., Sterbenz, J.P.G.: Path diversification: A multipath resilience mechanism. In: *Proceedings of the 2009 7th International Workshop on Design of Reliable Communication Networks (DRCN'09)*, pp. 343–351 (2009)
40. Routray, S.K., Sahin, G., da Rocha, J.R.F., Pinto, A.N.: Statistical analysis and modeling of shortest path lengths in optical transport networks. *J. Lightwave Technol.* **33**(13), 2791–2801 (2015)
41. Rueda, D.F., Calle, E. Marzo, J.L. Robustness comparison of 15 real telecommunication networks: Structural and centrality measurements. *J. Netw. Syst. Manag.* **25**, 269–289 (2017)
42. Santos, D., De Sousa, A., Mas-Machuca, C., Rak, J.: Assessment of connectivity-based resilience to attacks against multiple nodes in SDNs. *IEEE Access* **9**, 58266–58286 (2021)
43. Schatz, R., Hossfeld, T., Janowski, L., Egger, S.: From packets to people: Quality of experience as a new measurement challenge. In: Biersack, E., Callegari, C., Matijasevic, M. (eds.) *Data Traffic Monitoring and Analysis*. Lecture Notes in Computer Science, vol. 7754. Springer, Berlin, Heidelberg (2013)
44. Strogatz, S.H., Watts, D.J.: Collective dynamics of ‘small-world’ networks. *Nature* **393**, 440–442 (1998)
45. Tang, L., Liu, H.: *Community Detection and Mining in Social Media*. Morgan and Claypool Publishers (2010)
46. Van Mieghem, P.: *Performance Analysis of Communications Networks and Systems*. Cambridge University Press, Cambridge (2010)
47. Van Mieghem, P.: Pseudoinverse of the Laplacian and best spreader node in a network. *Phys. Rev. E* **96**(3), 032311 (2017)

Part II

Schemes of Resilient Routing

Chapter 4

Strategies and Concepts for Resilient Routing in Circuit-Switched Networked Systems



Providing communication possibilities to users, apart from offering storage and computation services, is one of the major aspects of any networked system. Since the locations of end users and servers in a network are characterized by a significant degree of geographical diversity, the transmission of information in these systems, both between end users themselves and between users and servers, is most often carried out via multi-hop paths, i.e., paths traversing many transit elements such as network links and nodes (routers, optical switches, etc.). Transmission paths are inevitably affected by failures of network elements traversed by these paths. Therefore, to maintain the continuity of transmission in failure scenarios, networked systems need to utilize the reconfiguration mechanisms of communication paths affected in a given failure scenario to bypass the failed network elements.

The objective of this chapter is thus to focus on mechanisms for resilient multi-hop communications able to remain operational in differentiated failure scenarios. A particular focus of this chapter is on approaches dedicated to circuit-switched wired networks providing transmission services on a per-flow rather than on a per-packet basis. Resilience mechanisms for packet-switched networks are the main topic of the next chapter of this book.

In this book, we define *resilient routing* as a routing scheme that can provide continuity of service in the presence of disruptions.

To maintain service continuity after failures, spare capacity (mostly related to link bandwidth) is commonly reserved in the network to provide the possibility to reroute the traffic along the *backup path* (also called *alternate path* or *protection path*) when the *primary (working) path* fails [19].

In general, a given multi-hop path in a circuit-switched network is established as a response to a given demand d_r defined as a triple (s_r, t_r, c_r) to provide a connection between nodes s_r and t_r of a guaranteed capacity c_r . Since this capacity is meant to be guaranteed for demand d_r also after a failure affecting its primary path, in this chapter, both working and backup paths of each demand d_r are assumed to be assigned capacity c_r at all consecutive links of these paths. Therefore, the greater the capacity to be protected, the more significant the task to protect the network from failures.

Following [7, 44], and as previously mentioned in Chapter 1, after the occurrence of a failure, the recovery process is initiated with the detection of a failure. It can be recognized, e.g., by IP-MPLS mechanisms like MPLS LSP ping or MPLS LSP traceroute [25] (sent along Label Switched Paths—LSPs), which are, however, time-consuming. Another option is detecting the failure based on the Loss of Light or Loss of Clock events.

Fault detection should be followed by fault localization and isolation (i.e., determination of the faulty node/link), which is necessary to stop further transmission of information via the affected element [7]. Fault notification messages are next sent to network nodes responsible for further triggering the recovery switching to redirect the affected flows onto the related backup paths.

After the physical repair of a faulty element, the final stage is normalization, i.e., recognition of the repaired element and return to the normal operational state. Concerning routing, this would generally mean a return to transmission paths that were in use before the failure (since recovery paths are typically nonoptimal, e.g., concerning resource usage or end-to-end delay).

The ideal *recovery time* (i.e., the time from the occurrence of a failure until redirection of the affected traffic onto backup paths) should not be greater than 50 ms since the higher layers often see a disruption lasting up to 50 ms as a transmission error only. Any disruption longer than 50 ms may result at least in packet losses or unavailability of service [41]. A detailed classification of the duration of outages from [15] is given in Table 4.1.

Although utilizing protection paths to provide automatic switchover seems relatively intuitive, implementing efficient recovery schemes, being both capacity-efficient and scalable, and including multiple criteria of QoS, especially in heterogeneous mesh network environments, is difficult.

In general, characteristics of any recovery method strongly influence the values of service recovery time [7]. In the later part of this section, we will highlight the most crucial recovery techniques, focusing on restoration time characteristics and their relation with the resource efficiency objective.

In this chapter, we first outline in Sect. 4.1 the architectural properties of ring networks and describe the related resilience mechanisms in detail. The latter part of this chapter, in turn, highlights the major schemes of resilient routing in mesh networks—the most common configuration of today’s communication systems. In particular, in Sect. 4.2, we explain the need to ensure the differentiated levels of resilience to match the differentiated requirements of services. The objective of

Table 4.1 Impacts of outage time from [15]

Target range	Duration	Main effects
Protection switching	≤50 ms	No outage logged; recovery of transmission control protocol (TCP) after one errored frame; no TCP fallback; no impact at all for most TCP sessions
1st type outage	>50 ms ≤200 ms	<5% voiceband disconnects; signaling system switchovers
2nd type outage	>200 ms ≤2 s	Common upper bound on distributed mesh restoration time; TCP/IP protocol back-off
3rd type outage	>2 s ≤10 s	Disconnections of all switched circuit services; disconnections of private lines; TCP sessions time-outs; Hello protocol affection; web page “not available” errors
4th type outage	>10 s ≤5 min	All calls and data sessions terminated; time-outs of TCP/IP application layer programs; users making attempts of mass redial; link-state advertisements (LSAs) sent by routers referring to failed links; updates of topology and resynchronization network-wide
Undesirable outage	>5 min ≤30 min	Massive reattempts causing heavy load of switches; noticeable Internet “brownout”; minor societal/business effects
Unacceptable outage	>30 min	Major societal impacts (societal risks: travel booking, impact on all markets); headline news; regulatory reporting often required; lawsuits; SLA clauses triggered

Sect. 4.3 is to provide a taxonomy for schemes of resilient routing in mesh networks according to the following main criteria: backup path setup method, failure model, scope of recovery procedure, usage of recovery resources, as well as the application of recovery schemes to multidomain and multilayer architectures of networked systems. The following two sections (Sects. 4.4 and 4.5) elaborate on the efficiency of recovery schemes in the two common architectures of communication networks, namely optical transport networks (OTNs) and IP networks. The summary of the chapter is provided in Sect. 4.6.

4.1 Resilient Routing in Ring Networks

A fundamental classification of resilience mechanisms based on the structure of communication networks divides the existing approaches into ring- and mesh-based. The former refers to architectures common about three decades ago, such as Synchronous Optical Networks/Synchronous Digital Hierarchy (SONET/SDH) [46] and early architectures of ring Dense Wavelength Division Multiplexing (DWDM) networks [31].

Based on flow direction, *ring networks* may be classified as unidirectional (referred to as *unidirectional path switched rings*—UPSR) or bidirectional (i.e., *bidirectional line switched rings*—BLSR), respectively. These networks consist

of *add/drop multiplexers* (ADMs) interconnected by fiber links, each fiber link providing transmission in parallel via its multiple nonoverlapping channels, each channel represented by a given wavelength λ_i [42]. The role of each ADM is to add (and drop) a certain subset of wavelengths from a given optical signal (by performing the multiplexing/demultiplexing operations) while allowing the other wavelengths to pass through the ADM.

Common variants of SONET ring networks include 2-fiber UPSR, 2-fiber BLSR (BLSR/2), and 4-fiber BLSR (BLSR/4). As shown in Fig. 4.1, both working and backup paths in ring networks are organized in rings.

In particular, as shown in Fig. 4.1a, each UPSR includes one ring for working paths and another for protection paths, configured to operate in opposite directions. In normal operational conditions, transmission in UPSRs is duplicated on both working and protection rings. The destination transmission node receives data by choosing between two signals, the one of better quality. In a failure scenario, a backup ring is used for detour purposes. Since protection rings are used in UPSRs simultaneously with working rings in a normal scenario, UPSRs are commonly considered an example of 1+1 Automatic Protection Switching (1+1 APS) [23].

A BLSR/2 structure shown in Fig. 4.1b involves two rings used simultaneously in a non-failure scenario for working paths operating bidirectionally (in opposite directions). It is important to note that in BLSR/2 only half of the capacity of each ring is used for working paths, while the other half is reserved for backup paths. The BLSR/4 ring structure illustrated in Fig. 4.1c involving four fibers between each pair of neighboring nodes consists of four rings: two rings for working paths operating in opposite directions and the other two rings configured similarly for backup paths.

Since link resources of a given ring reserved for protection paths in failure scenarios often serve low-priority traffic under normal conditions (and are preempted in failure scenarios), structures such as BLSRs can be considered as a particular form of 1:1 Automatic Protection Switching (1:1 APS) [23].

Due to differences in capacity efficiency of the considered ring systems, UPSRs gained popularity in local access networks, while BLSR configurations became important in metropolitan networks; however, they are both characterized by a relatively low level of available capacity, compared, e.g., to DWDM systems.

Ring networks are often called *self-healing rings*—SHRs. Thus, the considered variants are frequently referred to as USHR, BSHR/2, and BSHR/4, respectively [18].

In a scenario of a failure of a network element, e.g., a network link, as illustrated in Fig. 4.2, the respective detour over the failed link is formed, meaning that the traffic is switched at the node adjacent to the failed link onto the respective detour in the reverse direction. For the UPSR architecture, the non-affected parts of working and protection rings are merged to form a single ring, as shown in Fig. 4.2a. The same effect refers to the BLSR/2 architecture; however, its operational capacity is reduced by a factor of 2, as only one working ring (instead of the former two working rings) remains operational, as given in Fig. 4.2b. For the BLSR/4

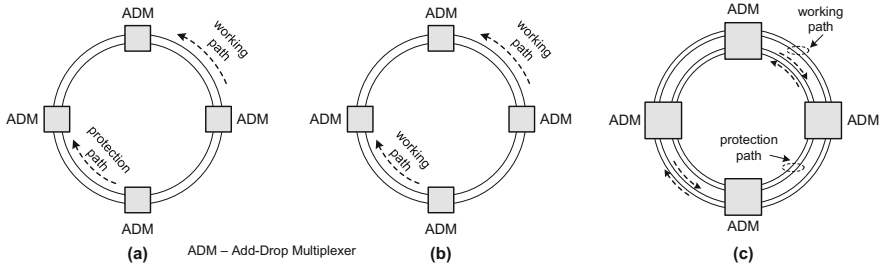


Fig. 4.1 Example of unidirectional path-switched ring (UPSR), 2-fiber bidirectional line switched ring (BLSR/2) and 4-fiber bidirectional line switched ring (BLSR/4) with add-drop multiplexers (ADMs)

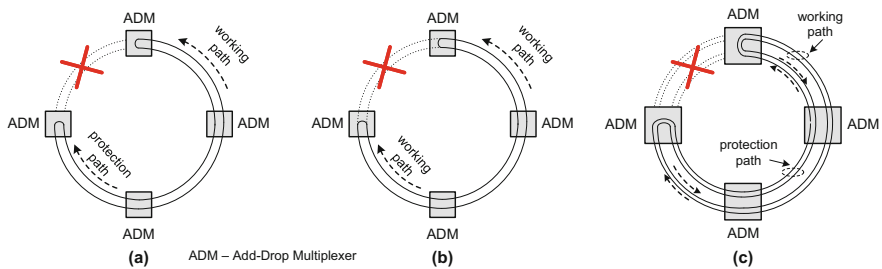


Fig. 4.2 Example operation of (a) UPSR, (b) BLSR/2, and (c) BLSR/4 in a scenario of a link failure

architecture, the respective working and protection rings are merged and form two operating rings after completing the recovery procedure, as shown in Fig. 4.2c.

Backup rings can thus be viewed as a preplanned protection scheme providing a short recovery switching time. However, their disadvantage is the high ratio of *network redundancy* (being the ratio of protection capacity to working capacity) of exactly 100% [19] in scenarios where every working ring is accompanied by the respective duplicate protection ring.

Architectures of ring networks commonly consist of a set of rings. The respective multi-hop transmission paths then often traverse a sequence of rings. In particular, in the case of a normal (i.e., non-failure scenario), transmission is provided using working rings. For instance, in a network consisting of two rings shown in Fig. 4.3a, the transmission path between ADM 2 and ADM 6 takes place via three transit ADMs: ADM 4, ADM 5, and ADM 7. However, in the case of a failure, e.g., a failure of a link between ADM 6 and ADM 7, as given in Fig. 4.3b, the respective backup rings are activated, and transmission is redirected at ADM 7. Therefore, after a failure, the transmission path becomes much longer, as traffic is now forwarded five times at ADM 4, ADM 5, ADM 7, ADM 5, and ADM 4.

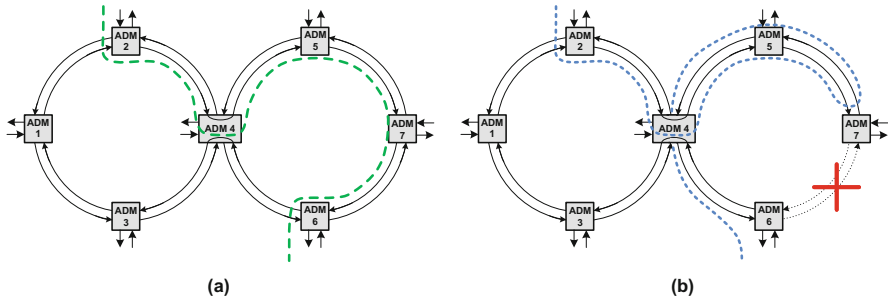


Fig. 4.3 Example multi-hop transmission in a system consisting of two bidirectional line switched rings (BLSR/2) in a normal scenario (a), and in the case of a single link failure (b)

4.2 The Need for Resilience Differentiation in Mesh Networks

The continuous traffic volume increase has triggered the evolution of ring-based topologies of wide-area optical networks toward mesh structures. Indeed, in wide-area networks where the cost of multi-hop transmission is determined by both the capacity and distance, a mesh topology of a networked system can serve a more significant number of demands compared to the capacity-equivalent ring structures [23].

In contemporary networks often characterized by a mesh topology [17], transmission paths are of end-to-end type, i.e., they do not form ring structures. As opposed to networks from the past engineered to offer a single type of service only (either voice or data), current communication networks are expected to provide a variety of services (e.g., real-time services as well as bulk data transfer) to support a wide range of applications (for example, online healthcare services based on data received from embedded sensor systems, massive content streaming, smart transportation, or emergency services) having differentiated requirements concerning resilience (sometimes referred to as the *quality of resilience* (QoR) [6]), as well as to the quality of transmission, as shown in Fig. 4.4.

This differentiation can also follow from different usage of the same application [6]. In other words, a given application can have differentiated requirements depending on how the users utilize it. For instance, even in the case of a classic telephone service, requirements on service availability for a company would be much higher than those sufficient for a home user.

Designing a communication network that consistently meets the highest requirements over the entire range of services (i.e., prepared to provide the highest level of service) by applying over-provisioning (i.e., adding an excessive amount of capacity, as in the case of optical DWDM backbone networks) would be highly costly and unreasonable. Such over-provisioning is also particularly expensive in wireless and

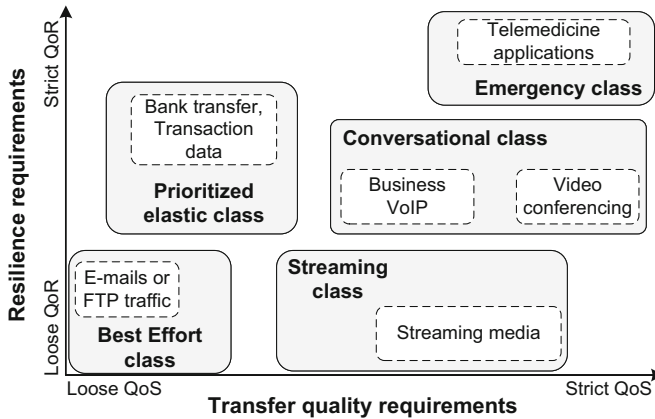


Fig. 4.4 Transfer quality vs. resilience requirements from [48]

access networks where bandwidth is limited (compared, e.g., to optical DWDM long-haul networks) [6].

Therefore, proper *resilience differentiation* (for example, as discussed in [30, 33]) is crucial in client-operator relations as an essential element of Service Level Agreements (SLAs). The operator, interested in maximizing the profit, is looking for cost-efficient resilience mechanisms tailored to specific SLA requirements. The willingness of clients to pay for the service is also differentiated. In particular, clients expect the lowest possible price for the service able to support characteristics of applications, but with only marginal regard to network mechanisms, the operator would deploy to support these applications. Utilizing multiple resilience mechanisms in the network may thus enable clients and operators to increase their profit.

Since applications are indeed characterized by a set of differentiated service requirements, including those related to service resilience, it seems reasonable to group applications into service classes and apply different models of service provisioning (as well as different resilience mechanisms) to these service classes. Indeed, the expectations of applications concerning the resilience requirements, including the level of service availability, continuity, or the maximum length of a service downtime period, vary from application to application, from almost no tolerance for service unavailability (e.g., for real-time telemedicine or financial services), via moderate tolerance of unexpected breaks in service provisioning (see, e.g., video streaming applications accepting slight changes in transmission delays due to the use of buffering) to best-effort service provisioning for the other applications with only marginal requirements on service continuity.

Several research papers also reflect this observation. For instance, in [2], four service classes are proposed based on their tolerance of the time for service downtime after a failure in a network, as summarized in Table 4.2.

Table 4.2 Requirements on service recovery time for resilience classes from [2]

Service class	RC 1	RC 2	RC 3	RC 4
Resilience requirements	High	Medium	Low	None
Recovery time	10–100 ms	100 ms–1 s	1 s–10 s	n.a.

In particular, resilience class 1 (RC1) from Table 4.2 represents high requirements on the maximum recovery time of up to 100 ms. RC2 denotes a class of medium requirements for resilience with a recovery time between 100 ms and 1 s. Low resilience requirements are characteristics of class RC3, tolerating the downtime between 1 and 10 s, while the last class (RC4) refers to the unspecified resilience-related requirements (meaning that any time for service recovery is acceptable for class RC4).

As discussed later in this chapter, different levels of service unavailability tolerance can be translated into the need to deploy different service recovery mechanisms. A general observation is that the time needed for the recovery of services and the resource cost of network resilience solutions are mutually opposing factors, i.e., the lower the acceptable time of service unavailability, the higher amount of extra resources needed (and thus, the more expensive the respective resilience scheme).

4.3 Schemes for Backup Path Resources Reservation in Mesh Networks

This section briefly overviews the most crucial resilience mechanisms proposed in the literature to provide fault-tolerant routing. Resilience differentiation can be obtained by combining several of them in a single network. Figure 4.5 outlines the most important classifications of resilience mechanisms for mesh networks, characterized in detail later in this section.

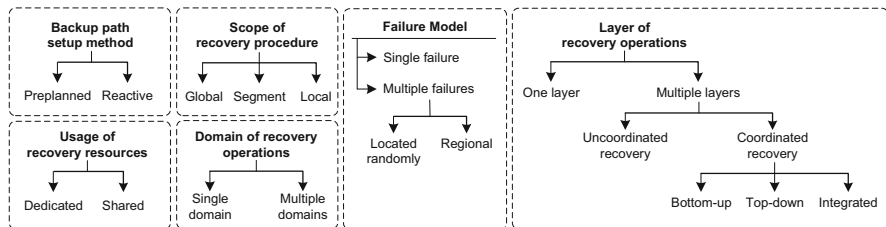


Fig. 4.5 Major classifications of resilience mechanisms in networked systems

4.3.1 Backup Path Setup Method

Concerning methodologies for setting up backup paths, these paths can be:

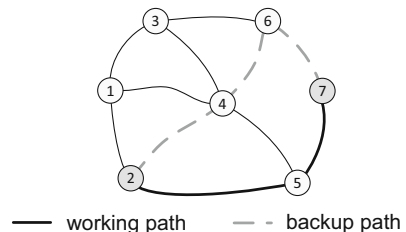
- Installed in a preplanned way (i.e., in advance when finding the working paths) often referred to as the *preplanned protection* in the literature [18]
- Determined dynamically (reactively) after the occurrence of a failure (known as *reactive restoration*)

The former case, historically derived from *Automatic Protection Switching (APS)* schemes [6], enables fast recovery of each failed transmission path (since backup paths are established in advance) [39]. Reactive restoration with its origins in IP networking [7] is, in turn, better in terms of efficiency of network resource utilization (since backup paths are installed here only when necessary, i.e., after a failure, and can reuse link capacities of failed transmission paths) [39]. However, it inherits all the disadvantages of dynamic IP routing, particularly the time-consuming recovery switching, path instabilities, and risk of loop creation. It also does not guarantee recovery due to the unpredictable amount of spare resources available after a failure [8].

In general, to provide 100% of restorability for working data flows, any backup path should not only be characterized by the same capacity as the corresponding working path, but it should also be link-/node-disjoint (i.e., have no common links/transit nodes) with the related working path—Fig. 4.6. The latter requirement is to guarantee that any failure of a link/node affecting the working path will also not disrupt the functioning of the respective backup path [19].

This disjointness is thus to assure that the two considered paths (i.e., working and backup path) of demand do not use resources of network elements belonging to the same *Shared Risk Link Group (SRLG)* defined in [18, 19] as the set of network elements, being either links, nodes, physical devices, or a mix of these, subject to a common risk of failure. Following [19], a given working path is said to be *SRLG-disjoint* with the respective backup path if both paths are not involved in any common SRLG.

Fig. 4.6 Example of end-to-end node-disjoint pair of paths between nodes 2 and 7



4.3.2 Failure Model

As summarized in [37], failures of network elements may occur due to many reasons, including, e.g., hardware faults, non-malicious human activities, malicious attacks, or natural disasters and disruptions. These events may result in *single failures*, i.e., failures of single network elements (links/nodes) at a time or simultaneous failures of many such elements (referred to as *multiple failures*). The risk for the occurrence of certain types of failures depends on many factors, such as the type of a network (local area network vs. wide-area network), dependability characteristics of system elements, as well as the environmental properties (i.e., location, size, intensity, and frequency) of natural disasters and other weather disruptions determining their impact on networked systems.

In scenarios of failures of single elements of a system (such as failures of single communication links or single nodes), it is sufficient to configure one backup path for a given working path. For instance, as illustrated in Fig. 4.6, a single end-to-end backup path being node-disjoint with the related working path can protect that path against any single node failure. The requirement on nodal disjointness naturally refers to all the nodes of a working path except for its end nodes, as both paths are expected to operate between the same pair of end nodes. If a faulty element is one of the transit nodes of a working path, then redirection of the affected onto the related backup path occurs.

Also, it is worth noting that nodal disjointness is stronger than link disjointness of working and backup paths, as in the latter case (involving a backup path being link-disjoint with the related working path), only protection against failures of single links can be assured. A failure of a node is, in turn, equivalent to a failure of all its incident links.

In failure scenarios not affecting working paths directly (e.g., scenarios of a failure of a transit element of a backup path or failures of any other element not traversed by either of these two paths), no recovery switching operation is needed. It is worth noting here that a single backup path can protect more than one network element. However, these elements are then not any possible ones but are associated with subsets of failure scenarios affecting at most one of the two considered paths (e.g., a simultaneous failure of node 5, link (2, 5), and link (1, 3) in Fig. 4.6, where the failure of the first two elements affects the working path only, while the third one does not have any impact on either of the two paths).

Failures of single network elements are indeed among the most common failure scenarios. Single link failures happen most frequently in wide-area networks [37], where it is difficult to ensure adequate physical protection for long-haul links (e.g., undersea optical cables, which can be cut by, e.g., movements of tectonic plates or damaged by shark bites). In turn, the frequency of single node failures is higher for local area networks, where the links are shorter and can, therefore, be better protected.

Scenarios of multiple failures include failures of several network elements that occur at the same time (e.g., a simultaneous cut of several optical links placed

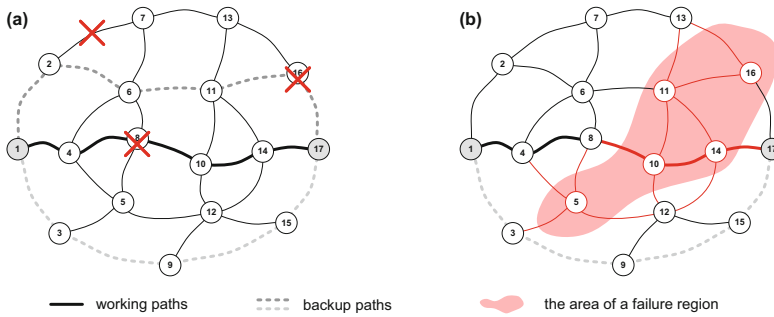


Fig. 4.7 Illustrations of scenarios of multiple random failures (a) and multiple failures confined to a given region (b)

together in a duct) or refer to failures happening sequentially before previously failed elements have been physically repaired. Among scenarios of multiple failures, we can distinguish either *multiple random failures*, i.e., failures occurring simultaneously at random locations of a system, such as failures of node 8, node 16 and link (2, 7) in Fig. 4.7a.

Another scenario of multiple failures occurring simultaneously at different locations might follow from human attacks targeted at several major nodes/links spread across the networked system. In all such cases, protection of the working path against simultaneous failures of k system elements can be achieved by installing a set of k mutually disjoint backup paths. For example, in Fig. 4.7a, a scheme involving one working path and two end-to-end backup paths being mutually node-disjoint is proper for protection against simultaneous failures of two nodes (e.g., nodes 8 and 16 as illustrated in Fig. 4.7a). It is important to note that parameter k cannot be any value, as the possibility to identify k -disjoint paths follows from the degree of system nodes. For instance, in Fig. 4.7a, since the minimum value of node degrees is 3, only three node-disjoint paths can be determined between these nodes, and as a result, protection against a simultaneous failure of at most two randomly selected (or attacked) elements of a system can be provided.

An important share of failure scenarios is linked to weather-related disruptions and natural disasters such as earthquakes, hurricanes, tornadoes, heavy wind, heavy rain causing flooding, or volcano eruptions occurring in certain geographical regions [32, 38]. As a result, they often lead to massive failures of multiple elements of a network located in a given region (referred to as *regional failures*). It is difficult to predict the occurrence of a disaster itself (e.g., earthquakes are known to be unpredictable as opposed to other disasters which are generally predictable [9]), and, in particular, to forecast the consequences of an incoming disaster such as the shape of a failure region and disaster intensity. Therefore, reactive recovery frequently turns out to be the legitimate procedure under natural disasters, where the configuration of the related backup paths is dynamically determined subject to the consequences of a disaster. In such cases, it is crucial to shape the related

backup paths in a way to make a detour over the actual failure region as presented in Fig. 4.7b, where backup path (1, 3, 9, 15, 17) provides a proper detour over the failure region.

To assure the adequate separation of working and backup paths for a given region of failures, these paths are calculated in a way to ensure their *D-geodiversity*, i.e., the geographical distance of at least D from any transit element of one of these paths to any other transit element of the second path [4, 13].

4.3.3 Scope of Recovery Procedure

Considering the scope of recovery, apart from *global protection* (often called *path protection*), assuming utilization of a single end-to-end backup path protecting the entire working path of a demand—Fig. 4.8a, *local protection* may be applied employing backup paths used to redirect the affected traffic over the failed link/node, as given in Fig. 4.8b [7]. The intermediate solution called *segment protection* [29] provides the existence of backup paths, each one protecting a given segment of a working path (consisting of several consecutive elements of a working path), e.g., as in Fig. 4.8c. Concerning the segment protection scheme, in scenarios of node failures, the respective neighboring segments additionally need to overlap each other by one link.

The path protection scheme is the most capacity-efficient concerning all variants of protection scope, while local protection against failures of single links is characterized by the highest amount of spare capacity needed to install the respective backup paths. As illustrated in Fig. 4.9 this is reflected by the total number of links

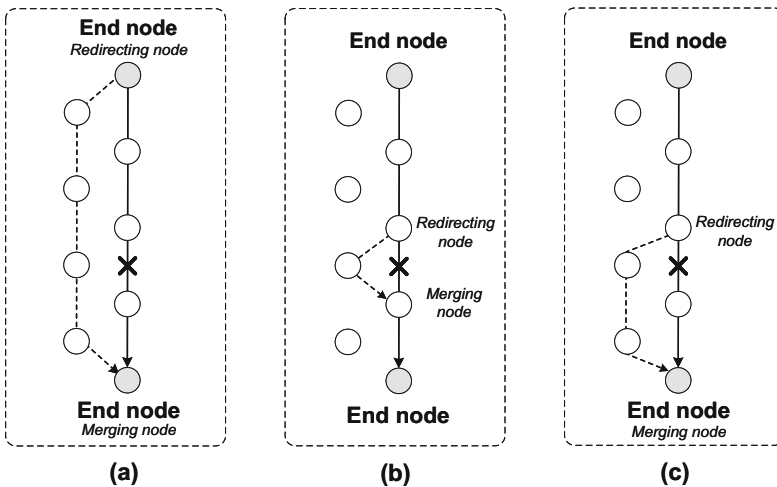


Fig. 4.8 Examples of recovery schemes: global (a), local (b), and segment (c)

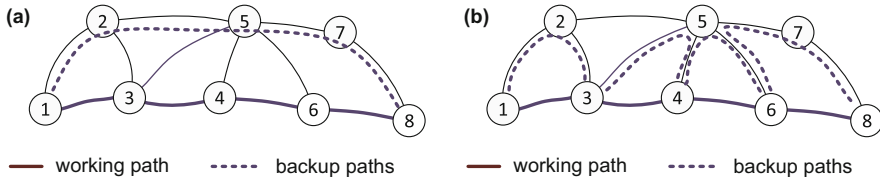


Fig. 4.9 Example illustration of a global protection scheme (a) and link protection scheme (b) for a working path between nodes 1 and 8

traversed by dedicated backup paths for the example scenario of protection of a working path (1, 3, 4, 6, 8), which is equal to four links in the case of global protection in Fig. 4.9a and nine links for the link protection scheme in Fig. 4.9b.

It is worth noting that the variants of the backup path scope analyzed in this section can coexist with the two modes of backup path setup methods. Therefore, among resilience schemes, we can identify *path protection*, *segment protection*, or *link/node protection* schemes (referring to backup paths installed in advance), as well as *path restoration*, *segment restoration*, or *link/node restoration* techniques based on installing the related backup paths reactively (after a failure).

4.3.4 Usage of Recovery Resources

Two solutions should be outlined when analyzing the schemes of assigning network resources to backup paths: dedicated and shared protection. In a *dedicated protection* scheme, resources (link capacities) of any given backup path are reserved to protect a single working path only. This technique is very costly but enables fast recovery of the affected traffic. Additionally, if preplanned protection is applied, backup paths may be either used in parallel with working paths in the normal operational state (i.e., the *1+1 protection* scheme of transmitting the signal simultaneously along both paths) or activated only for short periods to redirect the traffic affected by the failure (known as the *1:1 protection* scheme). In the latter case, capacity reserved for backup paths can be used to serve best-effort traffic under normal operation [18].

The disadvantage of a dedicated protection scheme is that, even though it provides the fastest recovery, it implies high additional cost of over 100% of the related working path cost due to the ratio of network redundancy exceeding 100% (since backup paths typically traverse more links than the corresponding working paths). Therefore, to limit the cost of a solution, the concept of *shared protection* was proposed in which several backup paths can mutually share link capacities. According to [27], the shared protection approach can limit the redundancy ratio to 35–70%.

If flows are required to be 100% restorable, sharing the link capacities by several backup paths is feasible only if the respective parts of working paths (i.e., being

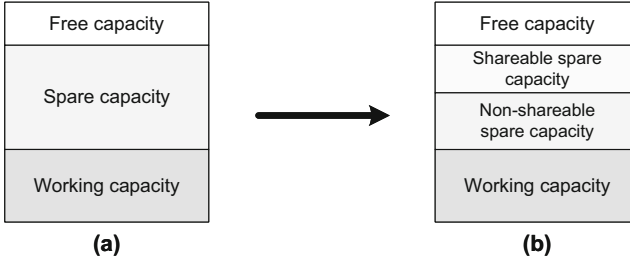


Fig. 4.10 Example link capacity classification under (a) dedicated, (b) shared protection

protected by these backup paths) are mutually disjoint, meaning that they do not share the same risk of failure (i.e., if they do not belong to any common SRLG) [19].

In resilient routing schemes, the capacity of any link can be generally classified into: (1) *working capacity* (i.e., used by existing working paths), (2) *spare capacity* (denoting capacity already reserved for backup paths), and (3) *free capacity* not used by any path (i.e., that can be allocated for either working or backup paths of new demands) [19].

As shown in Fig. 4.10, under backup capacity sharing, the spare capacity of any link is further divided into two classes: *shareable* and *non-shareable*. The former comprises backup capacity reserved for other backup paths that may be shared by the backup path to be established (i.e., when the respective part of a working path of an incoming demand is SRLG-disjoint with parts of all other working paths being protected by backup paths using this shareable capacity). The latter refers to the capacity already reserved for backup paths that cannot be shared.

Following [19, 36, 49], when finding a backup path in a backup capacity sharing scenario, the cost ζ_h of arc a_h is commonly defined as given in Eq. 4.1. According to this metric, the cost of a backup path link is thus determined only by the extra capacity that has to be reserved for a given backup path. Otherwise, if there is no need to reserve the extra capacity at a_h for this backup path (i.e., if the requested capacity is not greater than the shareable backup capacity at a_h), then ζ_h is set to a very small positive value of ε . Links with sharable capacity are thus preferred in backup path computations.

$$\zeta_h = \begin{cases} \varepsilon & \text{if } c_r \leq sh_h^{(r)} \\ (c_r - sh_h^{(r)}) \cdot \xi_h & \text{if } c_r > sh_h^{(r)} \text{ and } \bar{c}_h \geq c_r - sh_h^{(r)} \\ \infty & \text{otherwise} \end{cases} \quad (4.1)$$

where:

- c_r is the capacity requested for r -th demand;
- \bar{c}_h is the unused capacity of arc $a_h = (i, j)$;
- ξ_h is a unitary cost of arc a_h in working path computations;

$sh_h^{(r)}$ is the capacity reserved so far at a_h that may be shared with respect to the backup path of r -th demand.

Considering heuristic approaches to determine the resilient routing with shared protection, the *active path first (APF)* technique described in [20, 21] is typically used. In this two-step scheme, a working path of demand is found first and is followed by calculating a backup path for the topology of a residual network (i.e., with arcs traversed by the working path excluded). Numerous variants of this method have been proposed in the literature aimed at, e.g., determining the working path links in a way to get the most benefits from backup capacity sharing in the second phase [50].

However, if a backup path sharing scheme incorporates the shareability factor into the cost of a backup path link (e.g., as shown in formula (4.1), such backup paths occur to be nonoptimal concerning their length. As we showed in [36], in this case, backup paths may be even 40–50% longer compared to the results for a dedicated protection approach. For instance, for the example scenario from [36] given in Fig. 4.11, the path (2, 1, 3, 4, 7) of the total cost of $10+3\epsilon$ is chosen to be the backup path for the working path, even though there is a much shorter candidate backup path (2, 4, 7) but of the total cost of 27.

Due to the three-way handshake procedure of backup path activation [40] including sending the LINK/NODE FAIL message along the working path links followed by the exchange of SETUP and CONFIRM messages along the backup path, the total time of service restoration is mainly determined by message propagation delay along the backup path. Therefore, for the classical backup path sharing scheme, improved capacity efficiency comes at the price of increased service restoration time.

Concerning the overall time needed for the recovery of the affected working paths, the respective relations among variants of recovery methods referring to the backup path setup method, the scope of the recovery procedure, and the use of backup path resources are summarized in Fig. 4.12 based on [5].

In general, there is a trade-off between capacity efficiency and recovery time, i.e., the larger the segment of the working path being protected by a given backup path, the better capacity efficiency can be obtained, but for the price of longer recovery times. A detailed analysis of service recovery time for various recovery schemes is presented in [7].

Fig. 4.11 Example candidate backup paths (backup path sharing scenario)

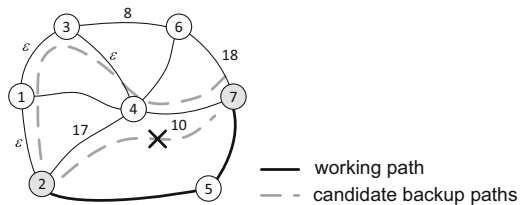


Fig. 4.12 Summary of relations among major variants of service recovery in the context of the overall recovery time

	faster ←-----→ slower		
Backup path setup method	Preplanned (resources pre-reserved)	Reactive (restoration/rerouting)	
	faster ←-----→ slower		
Scope of recovery procedure	Local	Segment	Global
	faster ←-----→ slower		
Use of recovery resources	Dedicated		Shared

To limit the problem of increased service recovery time under shared protection, our approach introduced in [36] assumes that both working and backup paths are first determined based on the same metric of link costs (i.e., reflecting the lengths of links only). In order not to increase the length of backup paths, backup path sharing is then performed “a posteriori” by finding the solution to the problem of vertex-coloring of the respective graph of conflicts for each network link individually (i.e., to perform capacity sharing for the established backup paths to comply with SRLG constraints concerning the respective working paths). After applying our capacity sharing solution, backup paths traverse the same links as under dedicated protection. Therefore, the time needed for the recovery of the affected flows is here as short as in the case of a dedicated protection scheme.

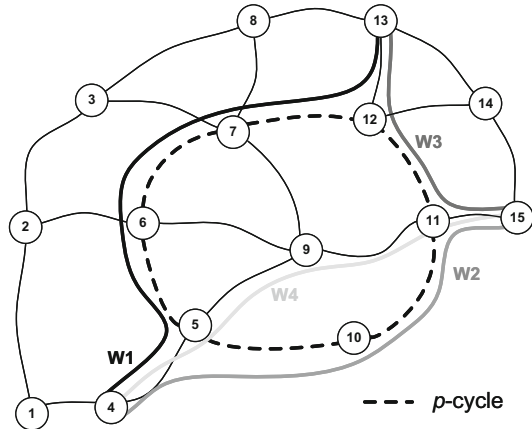
4.3.5 Protection Cycles

Protection cycles (or shortly *p*-cycles) originally introduced in [16] are ring-like protection structures designed for mesh networks to provide backup detours for a set of working paths. They are assumed to be preconfigured, i.e., calculated and installed in the system before the occurrence of any failure (at the time of establishing the respective working paths for demands). Unlike the configuration of ring networks, where protection rings are physically associated with the respective working rings, *p*-cycles are formed using the free capacity of network links. Therefore, contrary to ring networks, *p*-cycles do not impose any limitations on establishing working paths. Also, there is no strict relation between *p*-cycles and the physical structure of a network.

Similar to common backup paths, the role of *p*-cycles is to restore services in scenarios of failures of any single network element by redirecting the affected traffic onto a backup route provided by a given *p*-cycle. For the example working path W1 defined by the sequence of nodes (4, 5, 6, 7, 12, 13) in Fig. 4.13, the related *p*-cycle can provide a detour in the case of a failure of nodes 6 or 7 as well as links (5, 6), (6, 7), or (7, 12).

Similar to ring networks, *p*-cycles can protect segments of working paths traversing the respective *p*-cycle (referred to as the on-cycle spans), as in the case of working paths W1, W2, W3 that share a common *p*-cycle in Fig. 4.13. However, unlike backup rings in ring networks, *p*-cycles can also be used to protect working paths

Fig. 4.13 Example configuration of a p -cycle for four working paths W1–W4



straddling the protection cycle (i.e., not having any common link with the p -cycle), as the example working path W4 in Fig. 4.13. This additional feature improves the capacity efficiency of p -cycles, making it comparable to the one for shared backup path protection [1].

In general, a single p -cycle can protect multiple on-cycle and straddling spans if all these spans are SRLG-disjoint. For instance, the p -cycle from Fig. 4.13 is configured in a way to provide detours for the respective parts of four working paths W1–W4, since all these segments of the considered working paths are mutually disjoint (meaning that they will never fail simultaneously in a scenario of a single network element failure).

In the event of a failure, only two switching actions (like in ring networks) are necessary to redirect the traffic onto the protection path provided by the p -cycle (i.e., at the end nodes of the failed span). Therefore, p -cycles combine the best characteristics of mesh-based and ring-based protection methods, i.e., ring-like service restoration speed with mesh-like capacity efficiency.

Following [23], p -cycles are often selected either from the set of all distinct cycles for a given network graph or from a reasonably large set of candidate cycles. Regarding the combinatorial optimization issues, three major approaches have been used [1]: optimization of only spare capacity, joint optimization of working and spare capacity, and the concept of the protected working capacity envelope (PWCE) from [16] assuming routing of demands based on the information on already established p -cycles.

In research papers, protection cycles have been adapted to many networking scenarios. Apart from their original form focused on protecting single links of working paths (often referred to as link-protecting p -cycles [23]), other major variants include:

- *Path-protecting p -cycles* [22, 24] involving a single p -cycle to protect the entire working path, as illustrated in Fig. 4.14a. A given path-protecting p -cycle can protect a set of working paths, provided these paths are mutually disjoint. It is

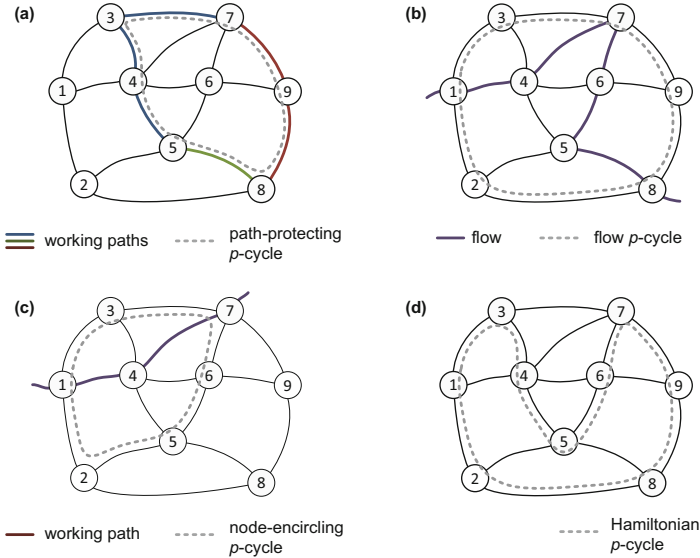


Fig. 4.14 Example configurations for major variants of p -cycles: path-protecting p -cycle (a); flow p -cycle (b); node-encircling p -cycle (c); Hamiltonian p -cycle (d)

worth noting that the constraint of mutual disjointness of working paths that share a given p -cycle enables the cycle to be fully pre-connected. That, in turn, means that there is no need for cross-connection operations after a failure (other than at the end nodes of a failed working path), which significantly reduces the time needed to activate the detours for the affected traffic [23].

- *Flow p -cycles* [14] protecting any given segment (a sequence of consecutive links) of a working path, as illustrated in Fig. 4.14b. The size of a segment protected by a given flow p -cycle can thus vary from a single link to the entire working path. Similar to path-protecting p -cycles, flow p -cycles can also protect against failures of transit nodes (if the related protected segments consist of at least two consecutive links).
- *Node-encircling p -cycles* [11] aimed at protecting working paths in scenarios of failures of their transit nodes. It is necessary that for any node of a given working path to be protected by a node-encircling p -cycle, the related adjacent nodes of that node on a working path must also belong to the p -cycle. Also, the protected node itself cannot be part of that p -cycle so that the cycle itself is not affected after a failure of a given node (see the example illustration of protection against a failure of node 4 provided for a given working path by a node-encircling p -cycle in Fig. 4.14c).
- *Hamiltonian p -cycles* [43]. Since there may be many p -cycles installed in the network to protect all the operating working paths, Hamiltonian p -cycles, being cycles that traverse all network nodes exactly once (see Fig. 4.14d), help reduce this number and, as a result, are characterized by even greater capacity efficiency,

compared to the scenario of using p -cycles traversing a fewer number of network nodes. Indeed, as explained in [43], for Hamiltonian p -cycles, the level of resource redundancy needed to provide protection can be as low as $1/(d_{avg}-1)$, where d_{avg} is the average node degree in the network topology.

4.3.6 Domain of Recovery Operation

End-to-end routing between distant locations frequently needs to be provided over multiple network domains, each defined based on administrative/geographical scope or network provider ownership and commonly identified with an autonomous system [7]. In the context of end-to-end routing, *multidomain routing* encounters problems related to the availability of precise routing information (i.e., following from topological characteristics of domains), which, due to confidentiality aspects, is generally not shared [44].

Another problem refers to the lack of exchanged information concerning the physical deployment of links in different domains related to SRLG disjointness. For instance, as given in Fig. 4.15, even though it may seem that the end-to-end routing using two separate paths over several domains meets the requirements of nodal disjointness, in practice links from different domains (for instance links (B1, B3) and (C2, C3) from Fig. 4.15) may be deployed in the same duct, e.g., physically routed over the same bridge, which raises the risk of a simultaneous failure of both of them. Therefore, applying *inter-domain recovery* techniques (i.e., joint actions taken in multiple domains to recover from failure) is often unrealistic.

As discussed in [26], recovery of communication paths in multidomain network configurations depends on multiple aspects. One of them refers to the location of the end nodes of a connection since both can be located either in the same domain, in separate neighboring domains, or separate non-neighboring domains. Concerning the location of a failed element, we can distinguish either intradomain failures of elements (i.e., failures of links or nodes located entirely within a given domain) or intradomain failures of border links [26, 28] such as of a link (A3, B1) in Fig. 4.16. Also, concerning the failure scenario, we can distinguish either failures of single

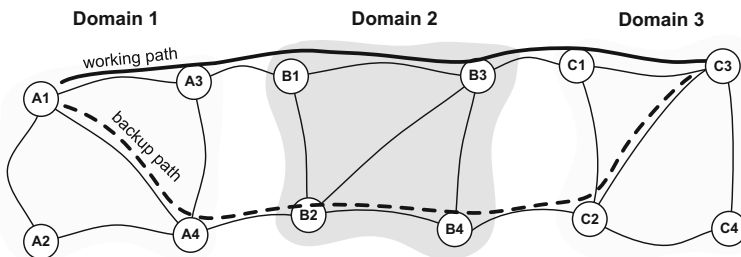


Fig. 4.15 Example scenario of multidomain routing

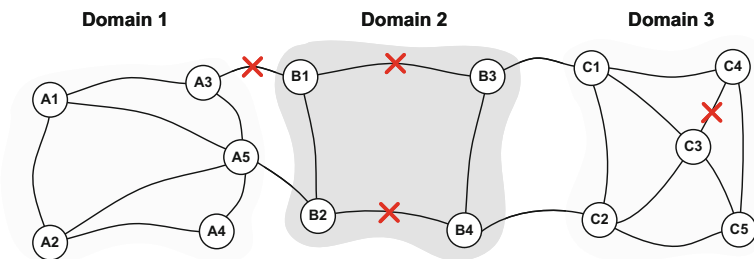


Fig. 4.16 An example illustration of a failure of an inter domain link (A3, B1); two intra domain links (B1, B3) and (B2, B4) implying, in fact, a failure of Domain 2; a failure of an intra domain link (C3, C4)

elements, failures of multiple elements located randomly, or failures of multiple elements located in a given region causing, e.g., a failure of a given domain (see the example scenario for Domain 2 in Fig. 4.16).

The simplest case from the recovery operations point of view is when both end nodes of a given affected connection are in the same domain. Then, it is common for both the working and backup paths to stay within that domain so that the recovery actions are confined to only one domain. In some cases, although both end nodes of a connection belong to the same domain, the related working path connecting them traverses another domain. In such cases, recovery actions should be kept local whenever possible to control the value of the connection restoration time to avoid propagation of recovery operations to other domains.

If both end nodes of a working path are located in different domains, and if failures occur in the domain being a transit one for a given working path, then recovery can be elastic so that the related backup path can even bypass that transit domain.

Cases described above naturally refer to unique characteristics of particular connections. In practice, it is rare to configure recovery schemes per connection. Instead, a single recovery method is deployed in the system, or certain recovery techniques are assigned to certain classes of demands. The following resilience techniques can be distinguished in the context of multidomain environments:

- Dedicated/shared protection, which implies setting up a pair of disjoint end-to-end paths (utilizing the path protection scheme) or configuring backup paths protecting smaller parts of the working path (i.e., implementing segment/link protection). Since end-to-end disjointness of working and backup paths of a connection is hard to achieve in the multidomain configuration (these paths may traverse the same element in a given domain, despite no indication of this issue in the aggregated view [26]), segment or link protection schemes seem more appropriate, especially if protection is arranged within domains.
- Restoration technique with backup paths calculated and set up after the occurrence of a failure. However, contrary to protection schemes, restoration via

multiple domains can be time-consuming and take seconds or more to determine a proper route, bypassing the failed elements.

- Adaptation of the p -cycles concept to serve in a multidomain configuration with the protection cycles determined as the shortest ones at the aggregated level (i.e., the level which considers only border nodes) and the protection mechanisms deployed afterward within domains, as proposed in [47].

4.3.7 Layer of Recovery Operations

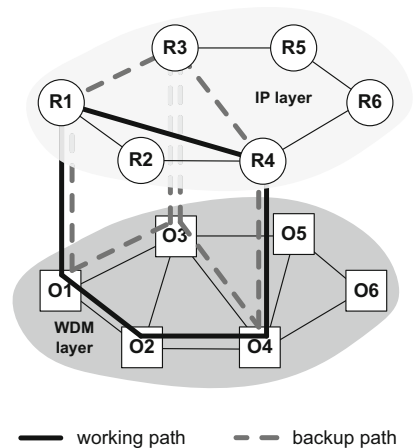
Internet IP traffic is mostly carried over optical networks (e.g., in the backbone). It means that a certain kind of communication network layering is applied there. Indeed, IP links are frequently virtual, meaning they are provided, e.g., by the optical multi-hop paths. Therefore, the resulting IP virtual topology is commonly formed over the underlying optical transport network.

This simple scenario mentions only two layers, i.e., the upper IP layer (frequently enhanced with Multiprotocol Label Switching (MPLS) functionality toward QoS provisioning, often referred to as IP-MPLS) and the lower Dense Wavelength Division Multiplexing (DWDM) [7]. In this case, IP-MPLS routers are connected to lower-layer Optical Cross Connects (OXC)s ports. OXC)s themselves are, in turn, interconnected in a physical mesh topology via multiwavelength optical links.

As shown in the example Fig. 4.17, a working IP layer path for a demand between nodes R1 and R4 consists of a direct virtual link (R1, R4) that is provided in the WDM layer by a lightpath (O1, O2, O4). For this demand, the backup IP layer path consists of two links (R1, R3) and (R3, R4), each one provided by a separate lightpath.

In general, this concept can be extended to the case of networks consisting of more than two layers with a client-server relationship between each neighboring pair of layers (including, e.g., Synchronous Optical Network

Fig. 4.17 Example scenario of a multilayer routing



(SONET)/Synchronous Digital Hierarchy (SDH) between IP-MPLS and WDM layers) [7]. The automated control of multilayer networks has been standardized in the Generalized Multiprotocol Label Switching (GMPLS) framework, including all necessary entities for use by routing and signaling protocols, in particular the User Network Interface (UNI) and the Network-Network Interface (NNI).

Considering the issue of interoperation between layers, following [7, 44], three main schemes may be distinguished, namely:

- The *overlay model* assuming that routing is performed in each layer separately (i.e., no routing information is shared between network layers)
- The *peer model* (also called *integrated model*) allowing for sharing of routing information between network layers
- The *augmented model* (or *hybrid model*) being the extension to the overlay model that makes information about nodes reachability available at the UNIs

In such a multilayer scheme, recovery actions after failures become even more complex. In general, due to the multiplexing (in the time domain) of lower-rate traffic from the upper layers into the higher-rate paths of the lower layers using time division multiplexing (TDM) [34, 35], the granularity of traffic switching becomes coarser from higher to lower layers. Therefore, more recovery actions must be performed in the higher layers (i.e., restoration of many low-rate flows) than in the lower layers (where recovery is efficient due to performing the recovery actions to the aggregate flows). Besides, recovery time in the upper layers may be additionally increased as a result of a significant number of recovery actions to be performed.

Concerning the order of layers in which recovery actions are performed, based on [7, 10], escalation strategies can be distinguished as follows:

- *Bottom-up* where recovery actions are initiated in the lowermost layer and are next propagated toward the upper layers. This technique's clear advantage is performing the recovery actions at an appropriate granularity. In particular, it means that handling the coarsest granularity actions in the lowermost layer is followed by recovery actions in the upper layers only concerning flows that could not be restored at the lower layer (e.g., a failure of the end node of the lower-layer path).
- *Top-down* where recovery is started in the uppermost layer. This approach, although allowing for better differentiation of recovery actions concerning multiple traffic classes, requires more complex signaling (since lower layers have no direct means to detect if the upper layer was unsuccessful in restoring the affected traffic).
- *Integrated* which combines characteristics of both the bottom-up and top-down strategies. In this case, the decision concerning the layer at which the recovery operations should be started depends on multiple factors such as received alarms or gathered survivability statistics.

If recovery actions are available in multiple layers, it is also essential to provide the appropriate interlayer coordination, including, e.g., determination of the sequence of layers according to which recovery actions are performed.

Such coordination between network layers is necessary to prevent multiple reactions of different layers to the same failure. This can be obtained, e.g., by the *hold-off timer* mechanism [44] used to postpone the recovery actions in the higher layer to give the lower-layer time for recovery of the affected traffic. After that, recovery actions are triggered in the higher layer for all the affected traffic that could not be restored in the lower layer.

Another proposal is to use the *recovery tokens* that help shorten the initialization of recovery actions in the higher layer. In this case, as soon as the lower layer finishes the recovery process, it sends a signal to the higher layer to start the recovery actions there.

Due to the client-server relationship, a failure of a higher-layer node (e.g., of an IP-MPLS router) cannot be restored in the lower layer. However, the reverse, i.e., recovery of a failure occurring in the lower layer (of a lower-layer link/node), is possible in the higher layer.

To perform the recovery actions, each layer must estimate the spare capacity necessary to reroute flows after failures. In particular, the IP-MPLS layer is commonly responsible for handling router failures (e.g., a failure of a router R3 from Fig. 4.17 which cannot be dealt with by the lower layer). In comparison, the lower (optical) layer is expected to handle failures of fibers/transit OXCs. Backup resources may be shared between network layers, forming the *common pool* of resources [44] so that the respective protection paths from different layers do not share the risk of being activated simultaneously.

4.4 Analysis of Recovery Time in the Optical Layer

Optical transport networks (OTNs) utilizing wavelength division multiplexing (WDM) are considered the primary communication technology for wide-area networks due to the huge capacity of each bidirectional fiber link of several Tbps. In OTNs, each network link is formed by a pair of unidirectional fiber links with their bandwidth divided into several tens of nonoverlapping transmission channels (wavelengths), each one offering a capacity of several Gbps. This enables parallel transmission of many demands at different channels of a given optical link at different wavelengths [31].

In OTNs, every transmission demand between a given pair of source and destination optical nodes is served by an optical path referred to as a *lightpath*. The nodes of OTN are *optical cross connects* (OXCs) and are used to forward the optical signal from the respective input fiber to the related output fiber of a lightpath (with or without wavelength conversion), all in the optical domain. It is often assumed that each OXC is integrated with the *access station* (ACS) via which the traffic either enters or leaves the lightpath (at the lightpath source node and destination

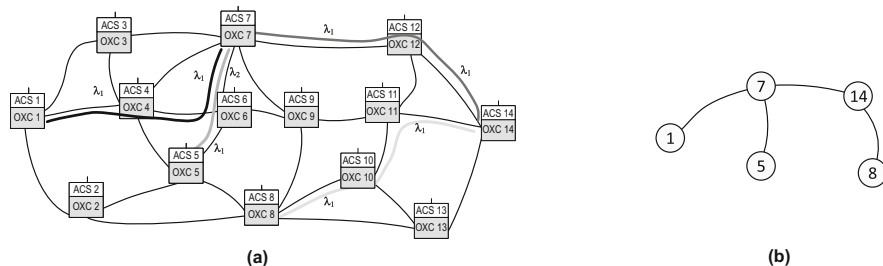


Fig. 4.18 Examples of (a) configuration of four lightpaths in a WDM network with wavelength conversion at node 6 for lightpath (5, 6, 7); (b) the related logical topology

node, respectively) [39]. At the source ACS of a given lightpath, the input signal is typically converted from the electronic to the optical form by the E/O converter and is next routed via OXCs along the consecutive links of the optical transmission path in the optical domain (i.e., without undergoing the optoelectronic conversion).

When switched from the input port to the related output port at each OXC, the signal can either remain on the same wavelength λ_i (e.g., due to lack of wavelength converters at OXCs) or be switched to another wavelength λ_j [39]. For example the two lightpaths (1, 4, 6, 7) and (5, 6, 7) in Fig. 4.18a are multiplexed together at link (6, 7), however, at different wavelengths λ_1 and λ_2 , respectively.

At the destination node of the lightpath, the signal is converted to the electronic form (using the O/E converter). Due to the optical signal attenuation progressing with distance, the signal needs to be periodically amplified (typically once per every 80 km of the optical link), which is done by amplifiers [31].

Data forwarding from a given lightpath to another lightpath is performed in the electronic domain, e.g., by the IP routers from the logical topology (where logical links are provided by lightpaths). For example, for the set of lightpaths from Fig. 4.18a, the related logical topology is provided in Fig. 4.18b. Therefore, transit nodes of lightpaths are not visible in Fig. 4.18b. As a result, nodes 1 and 8 are only three hops away in the logical topology, meaning that any IP datagram to be forwarded between node 1 and node 8 needs to be transmitted along three lightpaths (1, 4, 6, 7), (7, 12, 14) and (8, 10, 11, 14) with the electronic processing at each end node of each lightpath.

A single high-capacity lightpath can carry many low-rate (e.g., IP) streams by assigning timeslots concerning a given transmission channel for particular low-rate streams—the technique referred to as *traffic grooming* [51].

Due to the large distances between wide-area network nodes, optical links are at high risk of failure. Indeed, according to statistics, about 55% of cases refer to failures of single network links [37]. Since optical links undoubtedly serve large amounts of data, any failure of the optical network equipment can lead to severe data loss and thus be harmful to a huge number of end users. Therefore, upon the occurrence of any failure, it is crucial to minimize the *protection switching time*, seen as the downtime of the affected connection, and defined as the time between

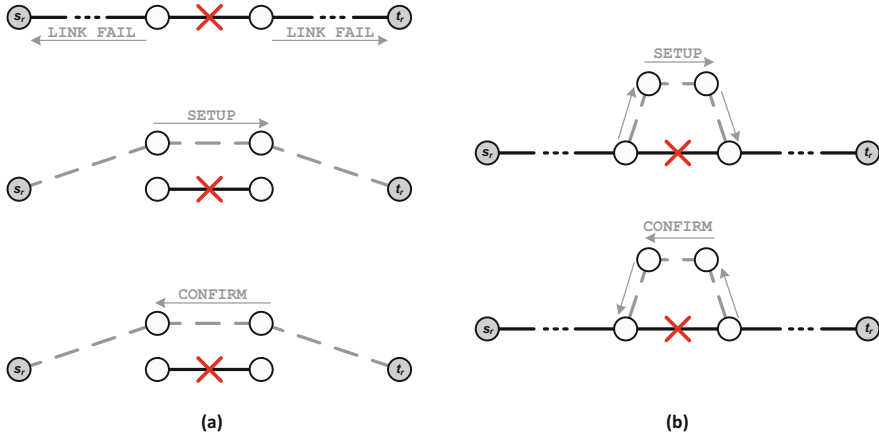


Fig. 4.19 Illustration of the recovery procedure under path protection (a) and link protection (b)

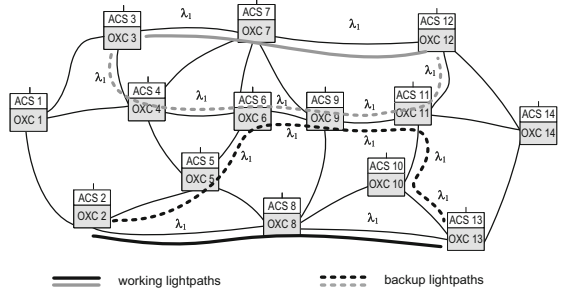
the instant an optical element (e.g., a link) fails, and the instant the backup path is activated as a detour for the affected traffic [40].

Mechanisms of resilient transmission are indeed an important part of the design of OTN architectures. Among all the resilience schemes discussed in this chapter, techniques based on path and link protection/restoration are most commonly used in practice. After detecting a failure of a network element by optical nodes being direct neighbors of the failed element (e.g., of an optical link by monitoring the levels of signal power along that link), the recovery procedure is initiated depending on the applied resilience mechanism.

In particular, in the case of dedicated protection, as illustrated in Fig. 4.19, after detecting and localizing a failure, the respective neighboring nodes of a failed element send the LINK(NODE) FAIL message to the respective source and destination nodes s_r and t_r of a backup path to be activated (note that this operation is not needed in the link protection scheme). After that, to activate the backup path, the respective SETUP message is sent along the backup path by the path source node s_r to its destination node t_r . It is followed by sending the CONFIRM message along the backup path from node t_r back to the source node s_r .

It is worth noting that the pair of SETUP/CONFIRM messages are used not only to activate the backup path but, in some cases (e.g., in the case of backup path sharing), also to apply proper configurations of OXCs along the backup path. Indeed, as illustrated in Fig. 4.20, under backup path sharing, before a failure occurs, it is impossible to configure switching at each intermediate OXC of backup paths. In particular, the configuration of optical signal switching at OXC 6 in Fig. 4.20 (i.e., at one end of the segment shared by backup paths), as well as at OXC 11 is not possible until the occurrence of a failure, since these OXCs will perform switching of the signal depending on the actual backup path activated.

Fig. 4.20 Example configuration of two backup paths sharing the same wavelength λ_1 at links (6, 9) and (9, 11)



The total time needed to activate the backup path thus depends on the related delays occurring when sending the LINK(NODE) FAIL, SETUP, and CONFIRM messages. As discussed in [40], the respective delay components include failure detection time F of about $500 \mu\text{s}$, message processing time D at a node (about $10 \mu\text{s}$ per node), link propagation delay P of $400 \mu\text{s}$ per each 80 km of the link, as well as time C to set up the OXC (up to $500 \mu\text{s}$). The overall recovery time is thus mainly implied by the total length of links and the number of nodes along the backup path.

Indeed, as discussed in [40], assuming the number of hops along the working path for sending the LINK(NODE) FAIL signals and the number of hops of the related backup path for sending the SETUP and CONFIRM signals equal to n and m , respectively, the total time T_{dp} for activation of the related backup path can be calculated for dedicated path protection scheme as given in Eq. 4.2, for shared path protection as provided by Eq. 4.3, and for shared link protection as provided in Eq. 4.4. Therefore, path protection schemes are characterized by higher protection switching time values than link protection mechanisms. However, the use of backup path sharing increases the total recovery time for shared path protection (T_{sp}) and shared link protection (T_{sl}) due to two factors: (a) the need to configure the OXCs along backup paths during the backup path activation procedure, which takes additional time of $(m + 1)C$ as given in Eqs. 4.3–4.4; (b) the increased length of backup paths (see discussion in Sect. 4.3.4 of this chapter). The overall value of the protection switching time for a given failure scenario is calculated as the average time to activate the backup paths for all affected working paths.

$$T_{dp} = F + nP + (n + 1)D + 2mP + 2(m + 1)D \quad (4.2)$$

$$T_{sp} = F + nP + (n + 1)D + (m + 1)C + 2mP + 2(m + 1)D \quad (4.3)$$

$$T_{sl} = F + (m + 1)C + 2mP + 2(m + 1)D \quad (4.4)$$

As discussed in [40], under reactive (dynamic) restoration, upon the occurrence of a failure of a given element of a working path, the arrival of a LINK(NODE) FAIL message at the respective source node of a detour triggers the procedure of searching for a backup path for each failed working path by broadcasting the respective SETUP message on all its outgoing links, which also reserves resources

on links used for broadcasting. The intermediate nodes act respectively. When the SETUP message arrives at the destination node of a detour, that node sends back the CONFIRM message along the path of the original SETUP message and configures the OXCs along that path. Resources reserved at links not confirmed by the CONFIRM message are soon released by the respective canceling messages. This completes the procedure of a dynamic setup of the backup path.

Since the effects of dynamic restoration depend on link resources available after the occurrence of a failure, the *restoration efficiency* coefficient is often used to determine the success ratio of recovery defined as the number of connections that were restored divided by the total number of affected connections [40].

4.5 Recovery Time in the IP-MPLS Layer

In multilayer networks, recovery of a large subset of affected flows can be provided at the IP layer. However, there are several reasons why such a design is not efficient. Firstly, applying the IP layer recovery mechanisms at the routing level may not be fast enough and, therefore, hard to meet stringent QoS requirements. Also, recovery at the optical layer often helps reduce the number of recovery actions that would otherwise need to be performed by the IP layer. This is particularly the case for lightpaths carrying many low-rate IP flows, which, if not restored jointly at the lightpath layer, would have to be restored individually by the IP layer.

However, as already mentioned in this chapter, not all recovery actions are feasible for the execution at the optical layer. An example scenario refers to a failure of one of the end nodes of a lightpath. Since the IP layer sees every lightpath as the IP logical link, a failure of the lightpath end node can be recovered only at the IP layer in the same way as the failure of the IP router (also, it is essential to note that IP routers and OXCs are also often integrated into a single unit).

Another reason for recovery at the IP layer is the need to provide different levels of protection to different IP streams by using different protection mechanisms for several classes of high-priority and low-priority streams [12]. Under optical layer recovery, such streams merged in a given lightpath would have to be recovered jointly using the same protection mechanism, which would not be adequate for a large subset of them.

Also, when proposing a mechanism for the IP layer recovery, it is important to consider the following issues:

- Failures of some IP links may already be handled by the respective backup lightpath set up in the optical layer.
- Only a certain set of high-priority IP traffic needs to be protected, while it is often enough to serve low-priority traffic on a best-effort basis without the recovery guarantees.
- To avoid duplicate recovery operations at different layers of a multilayer network, a proper coordination mechanism (such as the one based on the hold-off timer explained earlier in this chapter) is needed.

As this chapter focuses on mechanisms of resilience for connection-oriented systems, in the context of IP transmission, we draw our attention here primarily to the IP-MPLS proactive resilience mechanisms since multiprotocol label switching (MPLS) used to forward the traffic based on labels instead of addresses can be indeed considered as a close equivalent for the wavelength-based circuit switching characteristic to optical communications.

In the IP-MPLS layer, packet forwarding decisions are made solely based on labels assigned to packets (based on criteria such as the destination node or QoS requirements). These labels assigned to packets as soon as they enter the MPLS domain can be later replaced at transit nodes of a transmission path. Labels thus enable the creation of end-to-end circuits in the form of label switched paths (LSPs), making it relatively straightforward to apply the already discussed circuit-related recovery mechanisms.

The recovery of the affected MPLS traffic is performed similarly to classical protection/restoration mechanisms. It is important to note that the recovery techniques commonly operate in MPLS unidirectionally due to the unidirectional characteristics of MPLS LSPs. The IEFT RFC 3469 document [45] provides a detailed description of MPLS recovery mechanisms according to four aspects of configuration: (1) recovery model (rerouting vs. recovery switching); (2) resource allocation (pre-reserved vs. reserved on demand); (3) scope of recovery (local repair, global repair, or, e.g., multilayer repair); (4) path setup (preestablished or established on demand). This document also defines a sequence of operations in consecutive recovery phases, including fault detection and localization, fault notification, switchover, and post-recovery operation.

The variants of MPLS recovery are also described in detail in [3]. The major ones include:

- Global protection (i.e., path protection) assuming protection of each working LSP by a single backup LSP established in advance (with backup path resources pre-reserved) and being link-/node-disjoint with the related working LSP.

As discussed in [3], under global protection, the total time for recovery is composed of four components: time to detect the failure T_D assumed to be equal to 20 ms, the notification time T_N , the recovery switching time T_{RS} , and the restoration time T_R , as provided in Eq. 4.5.

$$T_r = T_D + T_N + T_{RS} + T_R = T_D + (fD + nD + \sum_{i=1}^n L_i P) + C + \sum_{i=1}^b L_i P \quad (4.5)$$

where:

- f is the flow (LSP) index;
- D is the message processing time at node assumed to be equal to $10 \mu\text{s}$;
- n is the number of nodes between the node upstream of the failure and the source node of a working path;
- b is the number of links along the backup LSP;

- C is the time to configure, test, and set up the forwarding table assumed to be between 1 ms and 10 ms;
- L_i is the length of i th link in km;
- P stands for a propagation delay of $5 \mu\text{s}$ per km.

In particular, since [3] assumes the sequential recovery of individual flows following their flow indices, the notification time T_N is extended by the processing time of a message at the node closest to the failure completed after the fD period.

- Local protection (often referred to as *fast reroute*), where each link of a given LSP is protected by its backup LSP. In the case of protection against a single node failure, a given backup LSP is assumed to protect two neighboring links of the working LSP. Since, under local protection, the number of backup LSPs can be large, a single backup LSP set up for a given MPLS link can be configured to protect all working LSPs traversing that link [3].

In this case, the overall recovery time provided by Eq. 4.6 is shorter than for global protection, as it does not include the related time to send the failure notification message from the node upstream of the failed element to the source node of a working path.

$$T_r = T_D + (fD) + C + \sum_{i=1}^b L_i P \quad (4.6)$$

The local protection scheme discussed above is also called the *one-to-one* backup scheme, as opposed to the *facility backup* approach allowing a single backup LSP to protect a set of working LSPs traversing the same sequence of MPLS links.

- *Rerouting/restoration* denotes a scheme of setting up the backup LSPs (and reserving the related resources for these paths) after detecting failures affecting the working LSPs. Depending on the assumed scope of the recovery scheme, we can distinguish between global or local rerouting/restoration. Due to the determination of backup LSPs after a failure, the time needed for MPLS rerouting schemes to redirect the affected traffic onto the backup LSPs is remarkably higher than for the related protection approaches. It can be measured even in seconds, compared to the millisecond values of recovery time characteristic of protection schemes.

4.6 Summary

In this chapter, we provided a detailed discussion of methods for communications resilience in circuit-switched networks. Starting with the analysis of solutions for classical ring networks, the main focus of the discussion was on the mechanisms of resilient transmission in mesh networks. Following the general classification of

transmission recovery schemes based on six criteria, including the backup path setup method, failure model, the scope of the recovery procedure, usage of recovery resources, operation in multidomain environments, and multilayer resilience, the related schemes for resilient transmission were explained. The analysis focused on the efficiency of recovery schemes assessed mainly in terms of the time needed to recover the affected paths, recovery success ratio, and resource efficiency.

A general conclusion following this analysis is that the two considered objectives, i.e., fast recovery and resource efficiency, are generally two contradicting factors. In particular, the shorter the detours (such as those in the link protection scheme), the shorter the time needed to activate the backup paths, but the higher the costs (regarding network resources). Backup path sharing schemes, although able to reduce the amount of resources needed for backup paths, generally tend to increase the time of recovery operations due to (commonly) lengthening of backup paths as well as forcing the configuration procedures of backup path transit nodes to take place no sooner than after the occurrence of a failure. Additionally, the efficiency of recovery operations can be further challenged by the configuration issues related to multidomain or multilayer routing.

Finally, it is essential to note that despite the availability of a multitude of schemes of resilient routing for circuit-switched networks in the related literature, deployment of a large subset of them has faced various problems, e.g., related to the coupling of the data and control planes common for many network system architectures. As a result, deployments of resilient routing mechanisms in commercial systems have been confined mainly to the path and link protection/restoration schemes in the last three decades. However, this situation is now changing with the growing popularity of software-defined networks—SDNs (e.g., utilizing the OpenFlow switches), where the control plane is decoupled from the data plane and localized in a logically centralized controller. Such a controller is flexible enough to implement practically any scheme of resilient routing since its operation is not confined by the constraints (as well as the life cycle) of the related data plane.

? Questions

1. Explain the principles of resilient routing in ring networks.
2. Describe the reasons for differentiation of the levels of service resilience.
3. Provide the classifications of resilience mechanisms in networked systems based on major criteria.
4. Characterize the main strategies of resilience based on the moment of calculating the backup paths.
5. Describe the major failure models considered in the design of resilient routing strategies.
6. Explain the methods of setting up backup paths based on the scope of a recovery procedure.
7. How do backup path setup methods impact the overall time for recovery of the affected flows? Provide the respective summarized view on this issue.
8. Explain the concept of p -cycles and describe its main features.

9. Discuss the challenges behind multidomain recovery schemes.
 10. Explain the strategies and the related challenges concerning recovery in multilayer networks.
 11. Discuss the main determinants of service recovery time in optical transport networks.
 12. Explain the service recovery process in IP-MPLS networks and characterize the related main components of service recovery time.
-

References

1. Asthana, R., Singh, Y.N., Grover, W.: *p*-cycles: an overview. *IEEE Commun. Surv. Tutorials* **12**(1), 97–111 (2010)
2. Autenrieth, A., Kirstadter, A.: Engineering end-to-end IP resilience using resilience-differentiated QoS. *IEEE Commun. Mag.* **40**(1), 50–57 (2002)
3. Autenrieth, A.: Recovery time analysis of differentiated resilience in MPLS. In: *Proceedings of the 4th International Workshop on Design of Reliable Communication Networks (DRCN'03)*, pp. 333–340 (2003)
4. Cheng, Y., Sterbenz, J.P.G.: Critical region identification and geodiverse routing protocol under massive challenges. In: *Proceedings of the 2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM'15)*, pp. 14–20 (2015)
5. Chiesa, M., Kamiński, A., Rak, J., Rétvári, G., Schmid, S.: A survey of fast-recovery mechanisms in packet-switched networks. *IEEE Commun. Surv. Tutorials* **23**(2), 1253–1301 (2021)
6. Chotda, P., Mykkeltveit, A., Helvik, B.E., Wittner, O.J., Jajszczyk, A.: A survey of resilience differentiation frameworks in communication networks. *IEEE Commun. Surv. Tutorials* **9**(4), 32–55 (2007)
7. Chotda, P., Jajszczyk, A.: Recovery and its quality in multilayer networks. *IEEE/OSA J. Lightwave Technol.* **28**(4), 372–389 (2010)
8. Colle, D., De Maesschalck, S., Davelder, C., Van Heuven, P., Groebbens, A., Cheyns, J., Lievens, U., Pickavet, M., Lagasse, P., Demeester, P.: Data-centric optical networks and their survivability. *IEEE Sel. Areas Commun.* **20**(1), 6–20 (2002)
9. de Sousa, A., Rak, J., Barbosa, F., Santos, D., Mehta, D.: Improving the survivability of carrier networks to large-scale disasters. In: *Guide to Disaster-Resilient Communication Networks*, pp. 175–192. Springer, Berlin (2020)
10. Demeester, P., Gryseels, M., Autenrieth, A., Brianza, C., Castagna, L., Signorelli, G., Clemente, R., Ravera, M., Jajszczyk, A., Janukowicz, D., Van Doorselaere, K., Harada, Y.: Resilience in multilayer networks. *IEEE Commun. Mag.* **37**(8), 70–76 (1999)
11. Doucette, J., Giese, P., Grover, W.D.: Combined node and span protection strategies with node-circling *p*-cycles. In: *Proceedings of the 5th International Workshop on Design of Reliable Communication Networks (DRCN'05)*, pp. 213–221 (2005)
12. Gerstel, O., Ramaswami, R.: Optical layer survivability: a services perspective. *IEEE Commun. Mag.* **38**(3), 104–113 (2000).
13. Gomes, T., Santos, D., Giraio-Silva, R., Martins, L., Nedic, B., Gunkel, M., Vass, M., Tapolcai, J., Rak, J.: Disaster-resilient routing schemes for regional failures. In: *Guide to Disaster-Resilient Communication Networks*, pp. 483–506. Springer, Berlin (2020)
14. Grover, W.D., Shen, G.: Extending the *p*-cycle concept to path-segment protection. In: *Proceedings of the IEEE International Conference on Communications (IEEE ICC'03)*, vol. 2, pp. 1314–1319 (2003)

15. Grover, W.D.: Mesh-Based Survivable Networks. Options and Strategies for Optical, MPLS, SONET, and ATM Networks. Prentice Hall PTR (2004)
16. Grover, W.D.: The protected working capacity envelope concept: an alternate paradigm for automated service provisioning. *IEEE Commun. Mag.* **42**(1), 62–69 (2004)
17. Haddadi, H., Rio, M., Iannaccone, G., Moore, A., Mortier, R.: Network topologies: inference, modeling, and generation. *IEEE Commun. Surv. Tutorials* **10**(2), 48–69 (2008)
18. Haider, A., Harris, R.: Recovery techniques in Next Generation Networks. *IEEE Commun. Surv. Tutorials* **9**(3), 2–17 (2004)
19. Ho, P.-H.: State-of-the-art progress in developing survivable routing schemes in mesh WDM networks. *IEEE Commun. Surv. Tutorials* **6**(4), 2–16 (2004)
20. Ho, P.-H., Tapolcai, J., Cinkler, T.: Segment shared protection in mesh communication networks with bandwidth guaranteed tunnels. *IEEE/ACM Trans. Netw.* **12**(6), 1105–1118 (2004)
21. Ho, P.-H., Tapolcai, J., Mouftah, H.: On achieving optimal survivable routing for shared protection in survivable Next-Generation Internet. *IEEE Trans. Reliab.* **53**(2), 216–225 (2004)
22. Jaumard, B., Rocha, C., Baloukov, D., Grover, W.D.: A column generation approach for design of networks using path-protecting p -cycles. In: Proceedings of the 6th International Workshop on Design of Reliable Communication Networks (DRCN'07), pp. 1–8 (2007)
23. Kiaei, M.S., Assi, C., Jaumard, B.: A survey on the p -cycle protection method. *IEEE Commun. Surv. Tutorials* **11**(3), 53–70 (2009)
24. Kodian, A., Grover, W.D.: Failure-independent path-protecting p -cycles: efficient and simple fully preconnected optical-path protection. *IEEE/OSA J. Lightwave Technol.* **23**(10), 3241–3259 (2005)
25. Kompella, K., Swallow, G.: Detecting Multi-Protocol Label Switched (MPLS) Data Plane Failures. IETF RFC 4379 (2006)
26. Larrabeiti, D., Romeral, R., Soto, I., Uruena, M., Cinkler, T., Szigeti, J., Tapolcai, J.: Multi-domain issues of resilience. In: Proceedings of the 2005 7th International Conference Transparent Optical Networks (ICTON'05), vol. 1, pp. 375–380 (2005)
27. Liu, Y., Tipper, D., Siripongwutikorn, P.: Approximating optimal spare capacity allocation by successive survivable routing. *IEEE/ACM Trans. Netw.* **13**(1), 198–211 (2005)
28. Manolova, A., Ruepp, S., Dittmann, L., Calle, E., Marzo, J.: Location-based restoration mechanism for multi-domain GMPLS networks. In: Proceedings of the 2009 International Symposium on Performance Evaluation of Computer & Telecommunication Systems, pp. 304–310 (2009)
29. Molisz, W., Rak, J.: Region protection/restoration scheme in survivable networks. *Lecture Notes in Computer Science*, vol. 3685, pp. 442–447. Springer, Berlin (2005)
30. Molisz, W., Rak, J.: A novel class-based protection algorithm providing fast service recovery in IP/WDM networks. *Lecture Notes in Computer Science*, vol. 4982, pp. 338–345. Springer, Berlin (2008)
31. Mukherjee, B.: *Optical WDM Networks*. Springer, New York (2006)
32. Pioro, M., Fitzgerald, E., Kalesnikau, I., Nace, D., Rak, J.: Optimization of wireless networks for resilience to adverse weather conditions. In: *Guide to Disaster-Resilient Communication Networks*, pp. 523–556. Springer, Berlin (2020)
33. Rak, J.: Priority-enabled optimization of resource utilization in fault-tolerant optical transport networks. *Lecture Notes in Computer Science*, vol. 4208, pp. 863–873. Springer, Berlin (2006)
34. Rak, J.: Fast service recovery under shared protection at connection level in WDM grooming networks. In: Proceedings of the 22nd IEEE International Symposium on Computer and Information Sciences (ISCIS'07), pp. 1–6 (2007)
35. Rak, J., Molisz, W.: Fast service restoration under shared protection at lightpath level in survivable WDM mesh grooming networks. *Communications in Computer and Information Science*, vol. 1, pp. 362–377. Springer, Berlin (2007)
36. Rak, J.: Fast service recovery under shared protection in WDM networks. *IEEE/OSA J. Lightwave Technol.* **30**(1), 84–95 (2012)

37. Rak, J., Hutchison, D. (eds.): Guide to Disaster-Resilient Communication Networks, pp. 1–818. Springer, Berlin (2020)
38. Rak, J., Hutchison, D., Tapolcai, J., Bruzgiene, R., Tornatore, M., Mas-Machuca, C., Furdek, M. Smith, P.: Fundamentals of communication networks resilience to disasters and massive disruptions. In: Guide to Disaster-Resilient Communication Networks, pp. 1–43. Springer, Berlin (2020)
39. Ramamurthy, S., Mukherjee, B.: Survivable WDM mesh networks, part I—protection. In: Proceedings of the IEEE Conference on Computer Communications (INFOCOM'99), vol. 2, pp. 744–751 (1999)
40. Ramamurthy, S., Mukherjee, B.: Survivable WDM mesh networks, part II—restoration. In: Proceedings of the IEEE Integrated Circuits Conference, pp. 2023–2030 (1999)
41. Ramamurthy, B., Sahasrabudde, L., Mukherjee, B.: Survivable WDM mesh networks. *IEEE/OSA J. Lightwave Technol.* **21**(4), 870–883 (2003)
42. Ramaswami, R., Sivarajan, K.N., Sasaki, G.H.: *Optical Networks: A Practical Perspective*. Morgan Kaufmann, Los Altos (2010)
43. Sack, A., Grover, W.D.: Hamiltonian p -cycles for fiber-level protection in semi-homogeneous, homogeneous, and optical networks. *IEEE Netw.* **18**(2), 49–56 (2004)
44. Schupke, D.: Multilayer and multidomain resilience in optical networks. *Proc. IEEE* **100**(5), 1140–1148 (2012)
45. Sharma, V., Hellstrand, F. (eds.): Framework for Multi-Protocol Label Switching (MPLS)-based Recovery, pp. 1–40. IETF RFC 3469 (2003)
46. Siller, C.A., Shafi, M.: *Synchronous Networking*. IEEE Press, IEEE Communications Society (1996)
47. Szigeti, J., Romeral, R., Cinkler, T., Larrabeiti, D.: p -cycle protection in multi-domain optical networks. *Photon. Netw. Commun.* **17**, 35–47 (2009)
48. Tapolcai, J., Cholda, P., Cinkler, T., Wajda, K., Jajszczyk, A., Autenrieth, A., Bodamer, S., Colle, D., Ferraris, G., Lonsethagen, H., Svinnset, I.-E., Verchere, D.: Quality of resilience (QoR): NOBEL approach to the multi-service resilience characterization. In: Proceedings of the 2nd International Conference on Broadband Networks (BROADNETS'05), vol 2, pp. 1328–1337 (2005)
49. Xiong, Y., Xu, D., Qiao, Ch.: Achieving fast and bandwidth-efficient shared-path protection. *IEEE/OSA J. Lightwave Technol.* **21**(2), 365–371 (2003)
50. Xu, D., Qiao, C., Xiong, Y.: An ultra-fast shared path protection scheme – distributed partial information management—part II. In: Proceedings of the 10th IEEE International Conference on Network Protocols (IEEE ICNP'02), pp. 344–353 (2002)
51. Ye, Z., Cao, X., Gao, X., Qiao, C.: A predictive and incremental grooming scheme for time-varying traffic in WDM networks. In: Proceedings of the IEEE INFOCOM'13, pp. 395–399 (2013)

Chapter 5

Resilience Schemes for Fast Recovery in Packet-Switched Communication Systems



Packet-switched networks, invented independently by Paul Baran and Donald Davies during the 1960s, have been playing a key role worldwide in delivering communication services in numerous deployment scenarios, including the Internet, data center networks, or enterprise networks [7]. In *packet switching*, data is organized into packets of a limited length consisting of the *packet header* and the *packet payload*. Packet headers include data utilized by the network nodes to deliver the packets to destination nodes. Packet load, in turn, denotes data used by higher layer protocols and applications. Concerning the TCP/IP protocol family, major forms of packets include Layer-2 Ethernet frames and IP Layer-3 datagrams.

The uninterrupted availability of packet-switched networks has become crucial for the operation of many classes of applications, e.g., related to business or health. However, in failure scenarios, it is often common that the response of the conventional resilience mechanisms deployed in the control plane is not efficient enough to provide a fast recovery of the affected communication paths. Indeed, the time needed for conventional control plane mechanisms to recompute communication paths can be high and even involve tens of seconds [12].

In this chapter, we discuss the properties of mechanisms extending the operation of conventional Layer-2 and Layer-3 route calculation schemes, which are necessary to reduce the noticeably long convergence time, i.e., the time needed for network nodes to obtain a new joint view of the network state and the set of updated transmission paths that are valid after a failure. In the remaining part of this chapter, we first discuss in Sect. 5.1 the properties of Layer-2 message dissemination schemes, namely the spanning tree protocol (STP) characteristic of Ethernet networks (being the most common IP Layer-2 technology), and further explain the major variants of STP aimed at ensuring fast recovery of affected spanning trees. Next, in Sect. 5.2, we discuss the properties of the selected IP Layer-3 fast recovery mechanisms, while

in Sect. 5.3, we highlight the mechanisms of fast recovery in IP-MPLS networks. Section 5.4 concludes the chapter.

5.1 Link-Layer Recovery Mechanisms in Packet-Switched Networks

This section aims to discuss the mechanisms of resilient transmission for Ethernet networks being the most common IP Layer-2 technology [7]. In general, assuring resilience is much more challenging in Ethernet networks since, contrary to IP Layer-3 multi-hop transmission, Layer-2 frames do not include fields similar to the Layer-3 Time-to-Live (TTL) to prevent forwarding loops in failure scenarios.

To avoid forwarding loops while restoring the Layer-2 communication paths affected by failures, solutions based on the concept of the *spanning tree* were proposed. In this context, the first notable scheme is the IEEE 802.1D Spanning Tree Protocol (shortly, *STP*) standardized as IEEE 802.1D [15] using a single spanning tree, i.e., a tree connecting all the nodes in the network. In this way, any pair of network nodes remains connected by a single path being part of that tree. In the event of a failure, the spanning tree is reconfigured in a way that provides transmission opportunities for any pair of nodes surviving the failure.

However, the procedure for reconfiguring a spanning tree in STP is relatively slow and often unacceptable for many applications. Indeed, following [11], the recovery of an affected spanning tree can even take tens of seconds, depending on the network size. Therefore, this section, apart from discussing the properties of STP, will also review the characteristics of two other representative approaches aimed at reducing the time needed for the recovery of the affected spanning tree, namely the Rapid Spanning Tree Protocol (RSTP) referred to as IEEE 802.1w [18] and a scheme using multiple spanning trees (IEEE 802.1s standard [17]), both later on included in the IEEE 802.1Q-2014 standard [16].

5.1.1 *Spanning Tree Protocol*

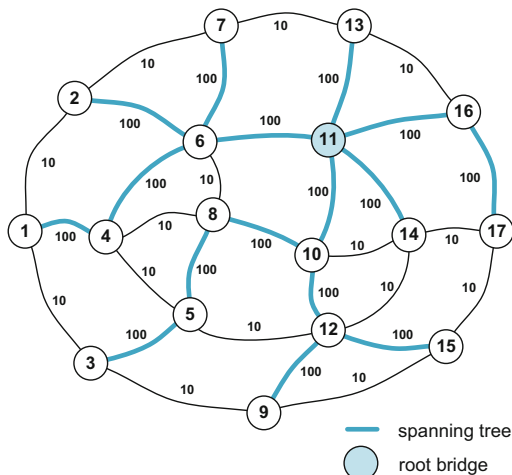
As already mentioned in this section, the purpose of the *Spanning Tree Protocol* (*STP*) proposed by Radia Perlman is to establish and maintain a tree topology connecting all nodes of an Ethernet network. In a tree topology, for every pair of network nodes, there is exactly one path in that tree connecting them (i.e., there are no loops).

Prevention of loops is indeed one of the major objectives of STP since, as already mentioned in this chapter, Ethernet frames do not provide a field similar to the Layer-3 Time-to-live (TTL) field to avoid the endless forwarding of frames likely to occur in mesh topologies. For this purpose, STP disables network links not

Table 5.1 Link cost vs. link bandwidth in STP (IEEE 801.1D-1998)

Link bandwidth	4 Mbps	10 Mbps	16 Mbps	100 Mbps	1 Gbps	2 Gbps	4 Gbps
STP link cost	250	100	62	19	4	3	2

Fig. 5.1 An example spanning tree determined by STP for a 17-node network (the values next to links denote the nominal link capacity in Mbps)



belonging to the spanning tree and, therefore, maintains only a single path between each pair of network nodes.

In STP, one switch in the network is elected as a *root bridge*. This election takes place based on the lowest value of bridge priorities configured for each switch manually. In the case of several equal lowest values of bridge priority configured for several switches in the network, a switch with the lowest MAC address among these switches becomes the root bridge. After that, each non-root switch determines the best communication path (i.e., of the lowest cost) between itself and the root bridge. This path will next become part of the tree. Table 5.1 illustrates the costs of links in STP in relation to link bandwidth based on IEEE 802.1D-1998, while Fig. 5.1 gives an example spanning tree for a 17-node topology with node 11 elected the root bridge. In general, as the costs of links are inversely proportional to their bandwidth, links of higher capacity are preferred in path computations.

During path calculations, STP switches exchange information using *bridge protocol data units (BPDUs)*. After all paths between switches and the root bridge are determined, each switch configures one of its ports as a root port, which connects it with the root bridge. Links not present in any path between switches and the root bridge are thus excluded from the tree (i.e., blocked).

Upon a change of the network topology (as a result of, e.g., adding a new node or following a failure of a given network element), topology change notification (TCN) BPDUs are sent by the respective non-root node (i.e., the switch at which the change was detected on one of its ports) toward the root bridge. Upon receiving the TCN BPDU, the root bridge initiates the topology update procedure by setting the related

“topology change” flag in exchanged BPDUs. Setting this flag triggers the spanning tree update by forcing the non-root nodes to recalculate their best paths to the root bridge.

Since the exchange of BPDUs in STP is periodic (typically once every 2 seconds), the reaction of STP to failures leading to a reconfiguration of the spanning tree, measured even in tens of seconds, is indeed slow. For this reason, as well as owing to the preference for high-capacity links when forming the spanning tree, remarkable data losses may occur in STP in failure scenarios. Therefore, the focus of mechanisms discussed in the remaining part of this section is on the fast recovery of spanning trees.

5.1.2 Rapid Spanning Tree Protocol

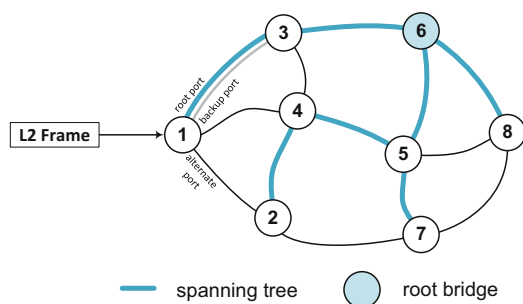
The main reason behind the introduction of the *Rapid Spanning Tree Protocol (RSTP)* was to reduce the long time of the STP algorithm convergence, e.g., in post-failure periods. Compared to STP, which typically requires from 30 to 50 seconds to re-establish the spanning tree, the time needed to finalize a new configuration of a spanning tree in RSTP is significantly improved. As verified in [22], RSTP is able to converge even within milliseconds.

RSTP is similar to STP concerning the rules for electing the root bridge, root ports, designated ports (i.e., ports leading to certain segments of a network), and in terms of blocking certain ports to avoid loops.

Compared to STP, apart from the root ports configured at each switch, RSTP introduces additional roles illustrated in Fig. 5.2 that can be assigned to ports of switches to improve the algorithm convergence time, namely:

- *Alternate port*: a port providing the alternate path from a given switch to the root bridge (i.e., a path that is different from the main one via the root port).
- *Backup port*: a port being a backup port for a given root port providing a backup path from the root bridge to a given network segment.

Fig. 5.2 An example configuration of a spanning tree including information on roles of selected ports of node 1 specified in RSTP



In RSTP, these ports can immediately enter the forwarding state instead of waiting for the final result of the algorithm convergence (as in STP) due to the ability of neighboring switches (i.e., connected by a point-to-point link) to acknowledge messages indicating that a given port asks to enter the forwarding mode.

As RSTP continuously monitors the network to detect any changes in network configuration (as in a link-state algorithm), it can detect changes in the network topology in a fast way. Also, unlike in STP, in RSTP, any switch can respond to the BPDUs received from the direction of a root bridge. This, in turn, enables switches to propose a spanning tree by sending the details of the suggested tree via their designated ports. Such a strategy of a rapid transition to the proposed variant of a spanning tree can visibly accelerate the entire convergence procedure.

Among several modifications of the RSTP protocol available in the literature, it is worth mentioning the following schemes:

- The strategy from [25] of Fast Spanning Tree Reconfiguration (FSTR) by means of executing an offline ILP program to identify for a set of predefined failure scenarios the best sets of links that could be added to the spanning tree (called reconnect links). As the preconfiguration of these reconnect links is done in advance (prior to failures), recovery time can be visibly reduced.
- The scheme for a spanning tree recovery after a simultaneous failure of two links from [26] utilizing a similar idea of adding links to the spanning tree as in the FSTR scheme, however, here aimed at avoiding loops in scenarios of failures of two links.
- An extension of the FSTR scheme provided in [28] that assumes protection of only those flows that require protection by triggering the recovery of a tree only with respect to failures of a certain subset of links. This is indeed a reasonable assumption since not all flows require full protection.
- The update of the spanning tree by reusing parts of the former spanning tree not affected by the failure [19]. In the event of a link failure, if that failed link belongs to the spanning tree, the technique from [19] would replace the failed link with a non-tree link that remains operational. This scheme involves three phases: fault detection, failure propagation (for broadcasting the information about the failure), and reconfiguration. Due to the reactive nature of this mechanism, maintenance of multiple structures of spanning trees (valid in certain failure scenarios) can be avoided.

5.1.3 Multiple Spanning Trees

The *Multiple Spanning Tree Protocol (MSTP)* originally proposed in the IEEE 802.1s standard provides an extension/evolution of STP and RSTP protocols. It is particularly useful for *virtual local area networks (VLANs)*, i.e., isolated broadcast Layer-2 domains. It allows for a parallel operation of multiple instances of spanning trees within the network (also called Multiple Spanning Tree Instances—MSTI) as illustrated in Fig. 5.3.

Fig. 5.3 An example configuration of multiple spanning trees

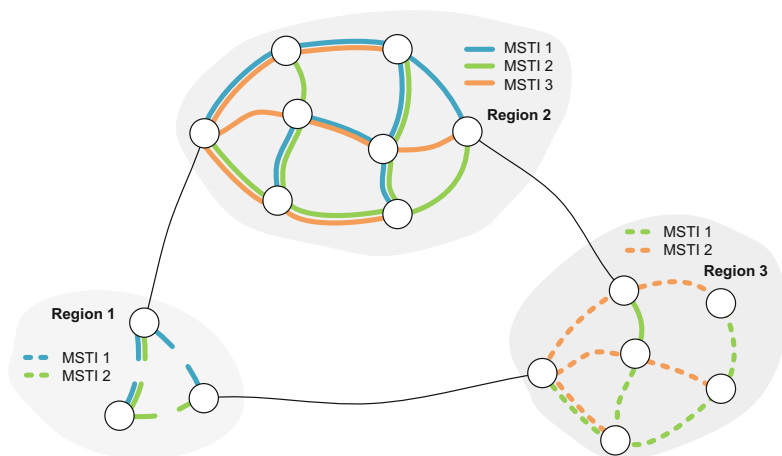
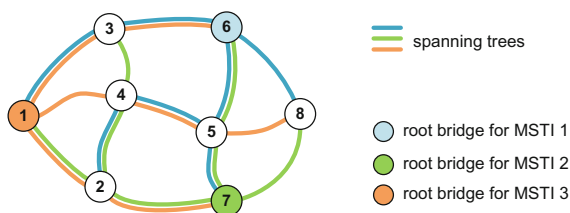


Fig. 5.4 An example configuration of multiple regions of MSTP operation

In MSTP, each spanning tree is assigned a unique VLAN number, which is included in the header of Layer-2 frames. Therefore, in MSTP, frames can be forwarded by switches within a given spanning tree if the VLAN identifier included in the header of the Layer-2 frame matches the VLAN number of a given spanning tree. The existence of multiple spanning trees thus allows Layer-2 frames to follow different paths depending on the value of the VLAN identifier stored in the frame header.

The possibility to set up multiple spanning trees within a network also allows for the configuration of multiple regions, where each region can be served by its own subset of spanning trees, as given in Fig. 5.4. These regions (together with other switches and local area networks) can be, in turn, connected by a single *Common Spanning Tree (CST)* and *Common and Internal Spanning Tree (CIST)* for connectivity among MST regions and other STP and RSTP LANs in a way to avoid forwarding loops on a global scale (i.e., beyond the reach of particular segments).

Similar to RSTP, MSTP also uses the concept of alternate ports and backup ports for fast restoration of end-to-end connectivity of network nodes in the case of failures affecting spanning trees. As discussed in [7], the fastest recovery can be achieved by substituting the root ports with the respective alternate ports. The

explanation for this is that in failure scenarios, switches located farther away from the failed element will not experience a network topology change. Otherwise, if the alternate port is not activated on time, the MSTP would trigger a conventional procedure for re-establishing spanning trees.

Among several alternatives/extensions to MSTP available in the literature, as noticed in [7], the following ones are worth mentioning:

- The Viking scheme from [29], which, contrary to MSTP, allows spanning trees to cover the network topology instead of being confined to particular network segments.
- A scheme involving the deployment of alternate trees configured before a failure occurrence from [24]. In the event of a failure, data transmission is switched onto a backup tree at a local node located upstream of the failed element. Two respective variants of this scheme were proposed in [24], namely the connection-based (where switching the traffic onto a given backup spanning tree depends on the source node, destination node, and the original VLAN ID of frames) and the destination-based (where the backup spanning tree for given frames is determined based on only the destination node, and the original VLAN ID of these frames, i.e., regardless of their source node).

5.2 Mechanisms of Fast Recovery in IP Networks

In this section, we discuss mechanisms of recovery designed for the Layer 3 (the network layer) of the Internet protocol stack. The concepts covered here are thus suitable for operation in IPv4 and IPv6 environments. However, as Layer 3 offers connectionless best-effort data transmission services, the implementation of fast recovery mechanisms based on preplanned protection (with backup paths established before failure) is hardly possible. In fact, ensuring a certain level of service quality in IP networks is already difficult in the normal (operational) scenario and becomes even more challenging in scenarios of failures since the connectionless behavior of the IP network layer does not allow for the association of packets with certain alternate paths before the failure occurrence. Therefore, without additional mechanisms of resilience deployed, it is common for IP datagrams to be served in a best-effort manner by means of backup detours determined reactively [7].

Despite these difficulties, there are several data plane mechanisms available for IP networks, which are designed to make the best use of the properties of link-state routing algorithms to recover the affected traffic as fast as possible in the IP domain by focusing on the adoption of preplanned local detours. These mechanisms of fast recovery in IP networks, often called *IP Fast-Reroute* (shortly, *IPFRR*), described in this section are designed to operate on top of unicast connectionless IP data plane service and typically require at most minimal updates (extensions) of the original IP specification [7, 8]. Notable examples include approaches based on shortest path rerouting, such as Loop-Free Alternates, Remote Loop-Free Alternates, Not-Via addresses, or Failure Insensitive Routing.

The shortest path rerouting schemes discussed here extend the operation of common link-state routing algorithms (e.g., Open Shortest Path First—OSPF [21] or Intermediate System-to-Intermediate System —IS-IS [6]). Link-state routing is, by default, based on a flooding mechanism used to periodically disseminate the actual information on the network topology to all routers in the network and, based on that, to recalculate the related primary paths by all routers. The shortest path rerouting schemes extend these schemes by also calculating one or more backup paths configured before the failure event at routers as the secondary next hops. Therefore, when a failure occurs, backup paths are already available and can be immediately used as bypasses for the affected flows.

Loop-Free Alternates (LFA)

LFA [1] is one of the simplest techniques focused on the deployment of repair paths (i.e., backup paths providing local detours over the failed element). The alternate paths are computed in a way to avoid loops (i.e., scenarios when the secondary hops, being not aware of the failure, are looping back packets to the router that initiated the switchover).

In order to ensure that the computed routes are loop-free, LFA verifies the fulfillment of a set of conditions given by formulas (5.1)-(5.4). In particular, for a given node s and a next hop e of node s on the shortest path toward t , assuming that $\text{dist}(i, j)$ is the shortest path distance between i and j , any node $n \neq e$ is classified as:

- An *ECMP alternate* if

$$\text{dist}(s, n) + \text{dist}(n, d) = \text{dist}(s, d) \quad (5.1)$$

- A *downstream neighbor LFA* if

$$\text{dist}(n, d) < \text{dist}(s, d) \quad (5.2)$$

- A *node-protecting LFA* if

$$\text{dist}(n, d) < \text{dist}(n, e) + \text{dist}(e, d) \quad (5.3)$$

- A *link-protecting LFA* if

$$\text{dist}(n, d) < \text{dist}(n, s) + \text{dist}(s, d) \quad (5.4)$$

These equations are ordered descending their coverage, i.e., any equal cost multipath (ECMP) alternate router is always a downstream neighbor LFA. Every downstream neighbor LFA is always a node-protecting LFA, and every node-protecting LFA is always a link-protecting LFA [7]. As discussed in [7], during the operation of LFA, when deciding about the alternate next hop, a stronger property is always preferred.

The major shortcomings of LFA are as follows:

1. By allowing only local detours, LFA can provide protection in the case of about 80% of single link failures and 40–50% of node failures due to topological constraints.
2. During the time of recovery, loops may be encountered when not all routers have a consistent view of the failure scenario.
3. To verify conditions given by formulas (5.1)–(5.4), additional execution of Dijkstra’s algorithm is needed to determine distances $\text{dist}(i, j)$.

Remote Loop-Free Alternates (rLFA)

The rLFA was proposed in [9] as an extension of the LFA scheme to improve the ratio of failure scenarios (over the result provided by the LFA) successfully covered by backup paths. For this purpose, compared to LFA, the scope of rLFA is extended to multi-hop backup paths [7]. In rLFA, any remote router is also allowed to become an alternate router if the three following conditions are met:

1. The originating router is able to perform packet tunneling from itself to that alternate router.
2. The shortest path between a pair of the originating router and the alternate router does not include the failed element.
3. For the remote router, there is a valid path to the destination node available in a considered failure scenario.

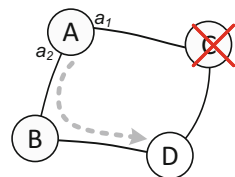
The failure coverage of rLFA backup paths, although higher than for LFA, may still not be able to reach 100%.

Not-Via

The *Not-Via* approach [5] was proposed to overcome the problem of limited coverage of failure scenario characteristic of LFA and rLFA schemes. In particular, in LFA and rLFA schemes, although for a given primary path node, there exists a suitable router, which could serve as a proper alternate next hop, all shortest paths to the candidate next hops may actually converge to a given failed next hop.

To avoid this scenario, Not-Via uses explicit signaling for advertising exclusions of certain failed elements when disseminating the reachability information. For example, as illustrated in Fig 5.5, a given router A having two interfaces a_1, a_2 and being aware of a failure of router C disseminates its reachability information, however, not via router C, following the recognition of a failure of node C. Any router receiving such advertisements will update all backup paths heading toward node A in a way that they omit the failed node C.

Fig. 5.5 Dissemination of explicit notifications on excluded next hops following a failure of router C



The traffic to be sent along a given backup path from a given source router toward a given destination router via router A needs to be tunneled between that source router and router A to make sure that any transit router between the source router and router A (e.g., router B in Fig. 5.5) will not forward the traffic back to the source router.

Failure Insensitive Routing (FIR)

FIR is able to provide full protection in scenarios of single link failures. Similar to Not-Via, FIR is also able to exclude failed elements from communication paths. However, contrary to Not-Via (which excludes certain elements by means of explicit Not-Via addresses explicitly communicated across the network), such exclusions are applied by routers in FIR by deducing them based on the way packets arrive at these routers. For instance, if certain packets from a given source router arrive through a nontypical interface of a given router (i.e., the one that would never be in use for that purpose in a normal scenario), a set of potentially failed links that may cause such behavior of packets can be identified. Such inferred information is next used by routers to redirect packets to other next hops.

A drawback of FIR is that full coverage in scenarios of all single link failures requires updates of the conventional IP data plane. This is because decisions on packet forwarding are based not only on the destination addresses but also on the interface of a given intermediate router they arrived at.

5.3 IP-MPLS Mechanisms for Fast Recovery

The architecture of *Multiprotocol Label Switching (MPLS)* [27] was introduced to assure a certain level of QoS in IP networks by default offering the best-effort services only. In MPLS, packets are forwarded across the network based on 20-bit *labels* contained in the MPLS packet header between the headers of Layer 2 and Layer 3 as given in Fig. 5.6.

IP-MPLS networks are formed by *label switch routers (LSRs)*, a subset of which localized at the border of the system is referred to as *label edge routers (LERs)* [31]. Contrary to conventional IP networks, packet processing at transit nodes is not based on the longest prefix matching but is solely determined by the values of the mentioned labels. These labels are assigned to packets by edge routers (i.e., when entering the IP-MPLS network) based on several parameters related to the IP destination address, QoS requirements, VPN identifiers, etc. and can be updated later on by transit LSRs.

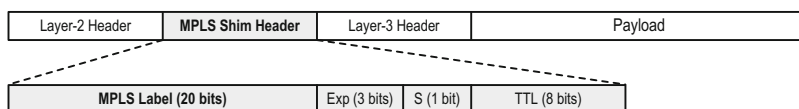
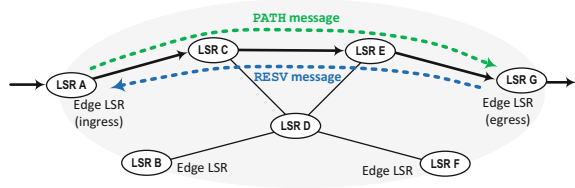


Fig. 5.6 The structure of an MPLS header

Fig. 5.7 Illustration of a procedure of setting up the LSP



MPLS labels, in fact, assign packets to certain *Forwarding Equivalence Classes (FECs)* defined as the groups of packets forwarded by several consecutive LSRs in a consistent manner [7], i.e., following the same path. Indeed, each LSR determines the next hop for a given packet solely based on the packet label following the respective entry from the label forwarding table of that LSR. The utilization of MPLS labels thus makes IP-based systems behave in a way that is closer to connection-oriented systems. Dissemination of information on the association of certain labels with FECs among the LSRs is provided by *Label Distribution Protocol (LDP)*.

As illustrated in Fig. 5.7, before packets are sent along a given *label switched path (LSP)*, the path needs to be established between a given pair of ingress and egress LSRs. For this purpose, the respective *PATH message* is first sent from the ingress LSR toward the egress LSR. It is important to note here that the demanded path can also be explicitly included in that *PATH message*. Otherwise, the installation of a path is determined by the *RESV message* sent back from the egress LSR to the ingress LSR via the sequence of transit LSRs in reverse order to the one for the *PATH message*. The *RESV message* also includes the label assigned to that path by the egress LSR. While forwarding the *RESV message*, the transit LSRs also reserve the necessary resources for the path. The reception of the *RESV message* by the ingress node completes the procedure of setting up the LSP.

Failures of LSRs or MPLS links may undoubtedly affect the label switched paths. Among various resilience schemes, we can distinguish the proactive ones using preestablished dedicated or shared backup LSPs, as well as reactive approaches where backup LSPs are determined only after the occurrence of a failure. Selected techniques belonging to these two classes are discussed in the remaining part of this section. However, as noted in [7], only proactive schemes are able to ensure fast recovery of the affected working LSPs.

5.3.1 Proactive Schemes of Resilient Routing in MPLS Networks

Mechanisms of fast recovery in MPLS networks typically involve local protection schemes, where backup LSPs provide local detours over the failed transit links or nodes of a working LSP. Such local protection techniques are commonly called **Fast**

Reroute schemes. As discussed in [7], they can be classified into *one-to-one backup* and *facility backup* schemes illustrated in Fig. 5.8.

In one-to-one backup schemes, a given backup LSP is designed to protect only a given working LSP. The facility backup approach, in turn, allows a single backup LSP to protect a set of working LSPs traversing the same sequence of MPLS links.

As both classes, in fact, imply local recovery operations, in the event of a failure, one of the end nodes of the backup LSP located closest to the failed element (called *Point of Local Repair—PLR*) redirects the traffic from the affected working LSP onto the backup LSP. Both types of schemes are considered by fast recovery procedures of the RSVP-TE (Resource Reservation Protocol-Traffic Engineering) solution from [23] commonly used in practice in MPLS networks.

Resilience schemes in IP-MPLS networks are undoubtedly resource demanding due to the need for reservation of link capacity also for backup LSPs. However, as these backup LSPs often remain unused in normal (i.e., non-failure) periods, techniques of **backup LSP sharing** can help lower the total cost of backup LSP installation. As discussed in several papers on fast reroute covering this aspect (see, e.g., [4, 30]), a set of several backup LSPs can share resources at a given link as long as the corresponding working LSPs are guaranteed not to fail at the same time. This, in turn, is assured by a mutual disjointness of these working LSPs as illustrated in Fig. 5.9.

Apart from solutions based on local detours, fast redirection onto the backup LSPs can be achieved by some of the global protection schemes with the redirection of the affected flow made by the LSP located close to the failed network element. Such an idea of **local redirection** is utilized, e.g., in the *local-to-egress protection* from [13, 14] involving a backup LSP configured in the reverse direction from the last-hop working LSP node toward the working LSP source node and next back to the destination node of a working LSP via a path being node-disjoint with the related working LSP, as shown in Fig. 5.10.

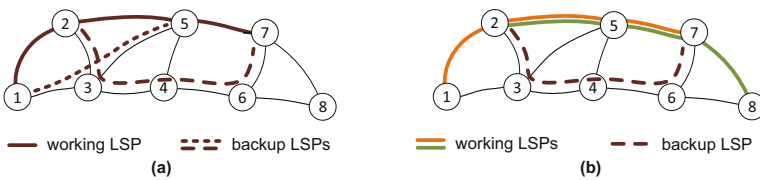


Fig. 5.8 Illustration of (a) one-to-one local protection method where each node of the primary LSP is protected by its own backup LSP and (b) facility backup scheme involving the use of one backup LSP to protect a certain joint segment of several working LSPs

Fig. 5.9 Illustration of a possibility for sharing the resources of backup LSPs at a link between nodes 4 and 5

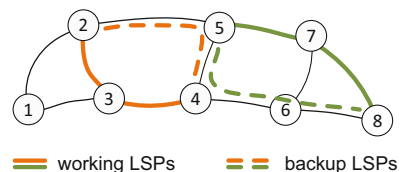


Fig. 5.10 Illustration of the local-to-egress configuration of a backup LSP

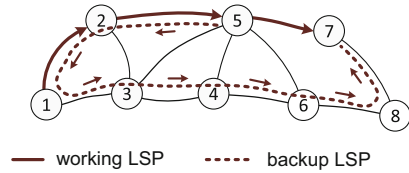
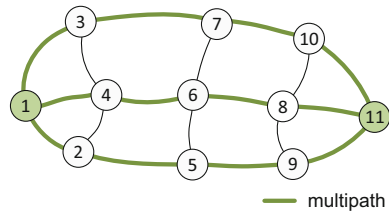


Fig. 5.11 An example configuration of a self-protecting multipath (SPM) between LER 1 and 11



In this scheme, after a network node/link traversed by the working LSP fails, switching the traffic onto the backup LSP is done at the working LSP node adjacent upstream to the failed element. As this operation does not involve any multi-hop recovery signaling, it is, therefore, fast. Also, only a single backup LSP needs to be set up for a given working LSP.

As soon as the upstream (source) LSR recognizes the backward flow, it marks the last packet sent along the affected primary LSP and stores the subsequent packets in its queue to avoid packet reordering. These queued packets are next released by that LSR from that queue and forwarded along the backup LSP (right after receiving the marked packet again from the downstream LSR and forwarding it along the backup LSP).

Apart from mechanisms involving the use of one working LSP for transmission for a certain FEC between a given pair of end nodes in the system, there are also schemes available that involve a set of disjoint paths utilized in parallel in a normal state. For example, the scheme of *self-protecting multipaths (SPMs)* from [20] uses a set of k preestablished mutually node-disjoint multipaths for data transmission between a given pair of end nodes, as illustrated in Fig 5.11.

In the event of a failure affecting, e.g., one of these paths, as the remaining $k-1$ paths continue their operation, the flow from the affected path is redistributed onto all other (operational) paths. Such a switchover can be indeed fast since there is no need for setting up any new path after a failure. An additional advantage of the SPM scheme is its ability to ensure adequate load distribution across the network.

5.3.2 Reactive Approaches to Resilient Routing in MPLS Networks

In the case of using the reactive schemes for resilience in MPLS networks, the determination of backup LSPs for all affected working LSPs is triggered after the

occurrence of a failure. Therefore, compared to proactive schemes, the overall time needed to switch the affected traffic onto the alternate paths is extended by the time to determine the backup LSP [2], which commonly denotes the time needed for the delivery of the PATH message to the egress LSR and the time to send the related RESV message back to the ingress LSR. Compared to protection schemes, the improved capacity efficiency characteristic of reactive schemes comes at a price of increased recovery time.

Therefore, as requirements on service availability are often differentiated for different demands, as discussed in [3], reactive recovery schemes involving rerouting of the affected traffic seem to be proper for services for which the acceptable recovery time is between 100 ms and 10 s. In [3], such services are identified as belonging to classes RC2 and RC3 with medium and low resilience requirements, followed only by the “best-effort” service class RC4, for which the recovery time upper limit is unspecified, and, therefore, no specific resilience mechanism is assumed. Any service requiring the recovery time to be lower than 100 ms (class RC1), in turn, calls for the use of preconfigured backup LSPs discussed in Sect. 5.3.1.

The validity of using reactive recovery methods in class-based resilience approaches is also confirmed in several other works, including, e.g., [10] introducing a proposal of a differentiated resilience scheme for serving anycast flows in MPLS networks in a way to survive failures of single links and failures of single replica servers. The three considered classes include Class 1 with working LSPs protected by the preestablished dedicated backup LSPs, each backup LSP leading to another replica server than the corresponding working LSP, Class 2 with working LSPs protected by the preestablished shared backup LSPs, and Class 3 with backup paths determined reactively after the occurrence of a failure using the free capacity of links available after a failure. The results of performance evaluation presented in [10] confirm that apart from the resource efficiency of the reactive recovery, in such a class-based approach, the existence of Class 3 (with no backup paths installed in advance) allows for reducing the blocking probability for demands from higher service classes.

5.4 Summary

In this chapter, we discussed the properties of mechanisms for the resilient operation of packet-switched systems. Our analysis focused on IP networks, particularly the resilience of IP Layer-2 Ethernet mechanisms, IP Layer-3 routing, and IP-MPLS switching. As Layer-2 frames do not include fields similar to the Layer-3 Time-to-Live (TTL) to prevent forwarding loops in failure scenarios, in this chapter, we highlighted the properties of selected spanning tree algorithms designed for fast recovery of spanning trees affected by failures. In the middle part of this chapter, we discussed major schemes of IP fast reroute, namely, LFA, rLFA, Not-Via, and FIR, to enable fast and loop-free recovery of the affected transmission routes using

local detours. Despite operating in a connectionless manner, these mechanisms can indeed be efficient in restoring the affected traffic, as they focus on adopting preplanned local detours determined proactively by link-state routing algorithms. The IP-MPLS recovery mechanisms described in the final part of this chapter can also operate efficiently in failure scenarios, mainly if their proactive variants are deployed.

? Questions

1. Explain the properties and the operation of the STP protocol.
2. Characterize the differences between the operation of the RSTP and the STP protocols.
3. Discuss the scenarios for the use of the MSTP protocol.
4. Explain the challenges in ensuring fast recovery in IP networks.
5. Describe the main features of the LFA technique.
6. Explain the recovery-related advantages of the rLFA scheme over the LFA approach.
7. Characterize the main features of the Not-Via scheme concerning the failure recovery aspects.
8. Discuss the difference between the FIR and the Not-Via scheme.
9. Discuss the main features of proactive mechanisms supporting the resilient operation of IP-MPLS networks.
10. Explain the operation of selected reactive mechanisms of resilience for IP-MPLS networks.

References

1. Atlas, A., Zinin, A.: Basic specification for IP fast reroute: Loop-free alternates (2008). <https://tools.ietf.org/html/rfc5286>
2. Autenrieth, A.: Recovery time analysis of differentiated resilience in MPLS. In: Proceedings of the 4th International Workshop on Design of Reliable Communication Networks (DRCN'03), pp. 333–340 (2003)
3. Autenrieth, A., Kirstadter, A.: Engineering end-to-end IP resilience using resilience-differentiated QoS. *IEEE Commun. Mag.* **40**(1), 50–57 (2002)
4. Alicherry, M., Bhatia, R.: Simple pre-provisioning scheme to enable fast restoration. *IEEE/ACM Trans. Netw.* **15**(2), 400–412 (2007)
5. Bryant, S., Previdi, S., Shand, M.: A framework for IP and MPLS fast reroute using Not-Via addresses. RFC6981 (2013). <https://tools.ietf.org/html/rfc6981>.
6. Callon, R.: Use of OSI IS-IS for Routing in TCP/IP and Dual Environments. RFC 1195 (1990). <https://www.ietf.org/rfc/rfc1195.txt>
7. Chiesa, M., Kamisinski, A., Rak, J., Retvari, G., Schmid, S.: A survey of fast-recovery mechanisms in packet-switched networks. *IEEE Commun. Surv. Tutorials* **23**(2), 1253–1301 (2021)

8. Cacic, T, Hansen, A.F., Apeland, O.K.: Redundant trees for fast IP recovery. In: Proceedings of the 2007 Fourth International Conference on Broadband Communications, Networks and Systems (BROADNETS'07), pp. 152–159 (2007)
9. Csikor, L., Retvari, G.: IP fast reroute with remote Loop-Free Alternates: the unit link cost case. In: Proceedings of the 2012 International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT'12), pp. 663–669 (2012)
10. El-Gorashi, T.E.H., Elmighani, J.M.H.: Differentiated resilience for anycast flows in MPLS networks. In: Proceedings of the 2009 11th International Conference on Transparent Optical Networks (ICTON'09), pp. 1–5 (2009)
11. Elmeleegy, K., Cox, A.L., Ng, T.S.E. : On count-to-infinity induced forwarding loops in Ethernet networks. In: Proceedings of the IEEE INFOCOM 25th IEEE International Conference on Computer Communications (IEEE ICC'06), pp. 1–13 (2006)
12. Francois, P., Filsfils, C., Evans, J., Bonaventure, O.: Achieving sub-second IGP convergence in large IP networks. SIGCOMM Comput. Commun. Rev. **35**(3) 35–44 (2005)
13. Haskin, D.L., Krishnan, R.: A method for setting an alternative label switched paths to handle fast reroute (2000). <https://datatracker.ietf.org/doc/html/draft-haskin-mpls-fast-reroute-05>
14. Hundessa, L., Domingo-Pascual, J.: Reliable and fast rerouting mechanism for a protected label switched path. In: Proceedings of the 2022 IEEE Global Telecommunications Conference (GLOBECOM'02), pp. 1608–1612 (2002)
15. IEEE: IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges, pp. 1–281 (2004)
16. IEEE: IEEE 802.1Q-2014 IEEE Standard for Local and metropolitan area networks—Bridges and Bridged Networks (2014)
17. IEEE: 802.1s-2002, Amendment to 802.1Q Virtual Bridged Local Area Networks: Multiple Spanning Trees (2002)
18. IEEE: 802.1w-2001, Part 3: Media Access Control (MAC) Bridges: Amendment 2—Rapid Reconfiguration, pp. 1–116 (2001)
19. Jin, D., Chen, W., Xiao, Z., Zeng, L.: Single link switching mechanism for fast recovery in tree-based recovery schemes. In: 2008 International Conference on Telecommunications (ICT'08), pp. 1–5 (2008)
20. Menth, M., Reifert, A., Milbrandt, J.: Self-Protecting Multipaths—A simple and resource-efficient protection switching mechanism for MPLS networks. Lecture Notes in Computer Science book series, vol. 3042, pp. 526–537 (2004)
21. Moy, J.: OSPF Version 2. RFC 2328 (1998). <https://www.ietf.org/rfc/rfc2328.txt>
22. Pallos, R., Farkas, J., Moldovan, I., Lukovszki, C.: Performance of rapid spanning tree protocol in access and metro networks. In: Proceedings of the 2007 Second International Conference on Access Networks & Workshops, pp. 1–8 (2007)
23. Pan, P., Swallow, G., Atlas, A.: RFC4090 - Fast reroute extensions to RSVP-TE for LSP tunnels (2005). <https://tools.ietf.org/html/rfc4090>.
24. Qiu, J., Gurusamy, M., Chua, K.C., Liu, Y.: Local restoration with multiple spanning trees in metro Ethernet networks. IEEE/ACM Trans. Netw. **192**, 602–614 (2011)
25. Qiu, J., Liu, Y., Mohan, G., Chua, K.C.: Fast spanning tree reconnection for resilient Metro Ethernet networks. In: Proceedings of the 2009 IEEE International Conference on Communications, pp. 1–5 (2009)
26. Qiu, J., Mohan, G., Chua, K.C., Liu, Y.: Handling double-link failures in metro Ethernet networks using fast spanning tree reconnection. In: Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'09), pp. 1–6 (2009)
27. Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol label switching architecture (2001). <https://tools.ietf.org/html/rfc3031>
28. Shan, D.M., Chiang, C.K., Mohan, G., Qiu, J.: Partial spatial protection for differentiated reliability in FSTR-based metro Ethernet networks. In: Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM'11), pp. 1–5 (2011)

29. Sharma, S., Gopalan, K., Nanda, S., Chiueh, T.: Viking: A multi-spanning-tree Ethernet architecture for metropolitan area and cluster networks. In: Proceedings of IEEE INFOCOM'04, vol. 4, pp. 2283–2294 (2004)
30. Wang, D., Li, G.: Efficient distributed bandwidth management for MPLS fast reroute. *IEEE/ACM Trans. Netw.* **16**(2), 486–495 (2008)
31. Xin, Ch., Ye, Y., Dixit, S., Qiao, Ch.: An agent-based traffic grooming and management mechanism for IP over optical networks. In: Proceedings of the 11th International Conference on Computer Communications and Networks (IEEE ICCCN'02), pp. 425–430 (2002)

Chapter 6

Optimization Methods for Resilient Routing in Connection-Oriented Communication Networks



Optimization problems related to routing in communication networks belong to the class of combinatorial optimization problems due to constrained network resources (mainly the limited capacity of network links). Indeed, although to establish end-to-end communication paths for a given set D of demands d_r , one may suggest calculating $|D|$ shortest paths (one for each demand d_r), in practice, these paths, in general, compete for limited capacity that is available at network links. Therefore, instead of considering only the shortest paths when selecting a path for a demand d_r , other candidate paths (often not shortest themselves, but possibly necessary in approaching the global optimum) should be examined as well.

However, the number of candidate paths can be huge (even for a single demand d_r) and often intractable when addressing real-world problems. The problem becomes even more significant in the context of resilient routing, where, for each demand d_r , one or more mutually node-/link-disjoint alternate paths need to be installed in parallel to working paths.

In this chapter, we use a notion of demand d_r defined by a triple (s_r, t_r, c_r) to establish the communication paths between its source node s_r and destination node t_r requesting the capacity (also called the *demand volume* [22]) c_r at all its consecutive links. This chapter aims to highlight the major properties of optimization schemes with a typical objective of minimizing the total cost of resilient routing in connection-oriented communication systems.

The class of resilient routing problems belongs to a broad *network flows* domain encompassing applications in various disciplines, including computer science, electrical engineering, management, operations research, or physics [1]. In these disciplines, a common task is to move efficiently (e.g., fastly, cost-effectively) given entities from source to destination nodes along selected paths formed by sequences of transit nodes. In this context, Ford and Fulkerson [7] are often considered precursors of the network flow theory.

Generally, formulations of optimization problems consist of an objective function followed by a set of constraints (equality and/or inequalities) that define the domain of the problem. The optimal solution to a given optimization problem is the one for which the objective function achieves its global optimum over its domain, i.e., global minimum or maximum—depending on the definition of the problem. An optimization problem is called a *linear programming (LP)* problem if its objective function and all constraint functions are linear and all variables are continuous [22]. Additionally, if all variables in the formulation are integer, then the formulation specifies an *integer linear programming (ILP)* problem. If only a subset of variables are integer (and the remaining ones are continuous), the problem is referred to as the *mixed-integer linear programming (MILP)* one. If all variables are required to be binary (that is equal to 0 or 1), the formulation is called the *binary linear programming (BLP)* problem. If at least one of the formulas in the model is nonlinear, the formulation refers to a *nonlinear programming (NLP)* problem.

In the remaining part of this chapter, we first discuss in Sect. 6.1 the network flow problem being an important basis for further considerations of this chapter concerning the schemes of (resilient) routing. After that, in Sect. 6.2, we define a network model used in this chapter, taking into account related technological assumptions. Next, in Sect. 6.3, we explain the basic modeling for the problem of establishing a set of single communication paths using two standard notations: node-link and link-path notation.

The main focus of this chapter is on modeling the optimization problems related to resilient routing in connection-oriented communication systems, considering the most common strategies of preplanned protection investigated in detail in Chap. 4. In this context, in Sect. 6.4, we investigate an optimization model to find the shortest pairs of disjoint working and backup paths, where the link capacity of each backup path is assigned to the respective working path exclusively (i.e., without the backup path sharing). In Sect. 6.5, we highlight the properties of a common optimization model, here referred to as the “a priori” sharing scheme to incorporate sharing of link capacity by backup paths. In Sect. 6.6, we present the optimization model for the original concept of the “a posteriori” sharing of backup paths, which overcomes the major disadvantage of the “a priori” sharing, which is the increased length of backup paths. Section 6.7 discusses details of the optimization model to establish the protection cycles (shortly *p*-cycles) utilized jointly by several working paths. Section 6.8 explains the properties of the most relevant mathematical methods and programming tools that can be used to solve the optimization problems addressed in this chapter. Sect. 6.9 provides the concluding remarks.

6.1 Network Flows

One of the major network flow problems is the *minimum cost flow* problem (MCFP) [1] where the objective is to transport a given entity (commodity) in the network between a given source node of demand and the destination node at a

minimum cost. In the context of data transmission in communication networks, this problem translates into a task to determine the least-cost end-to-end communication path between the related end nodes of transmission. The notion of the commodity is closely linked to communication patterns. In this context, we can distinguish [22]:

- *Multi-commodity flow problem* referring to the problem of multiple commodities (i.e., demands) defined between different source and destination nodes
- *Single-commodity flow problem* where the objective is to solve the network flow problem for a single demand

As discussed in the book by Ahuja et al. [1], for minimum cost flow problems, it is common to consider a directed network represented by a directed graph $G(N, A)$ where N is the set of nodes representing network nodes, while A is a set of arcs $a_h = (i, j)$ representing directed network links between certain pairs of network nodes i and j . Each arc is also assigned its cost ξ_h (referring to the cost per unit flow via that arc) and capacity c_h being the maximum amount able to flow via that arc. Each node n is associated with an integer value $b(n)$ denoting its supply/demand. In general, *supply nodes* are characterized by $b(n) > 0$, while for *demand nodes* values of $b(n)$ are negative. For nodes serving as *transit nodes* (often referred to as the *transshipment nodes*), we have $b(n) = 0$. In minimum cost flow problems, decision variables x_h represent the amount of flow served by arc a_h .

Directed graphs are indeed convenient in representing networks of various types. Concerning communication networks, their directionality follows general features of signal propagation from a given source node toward a destination node of a link. A bidirectional transmission in communication networks between a given pair of neighboring nodes i and j is, in turn, commonly provided by a pair of oppositely directed links (see, e.g., DWDM links), modeled in graph G by a pair of opposite arcs $a_h = (i, j)$ and $a_{h'} = (j, i)$.

Following [1], the minimum cost flow problem can be stated for a single commodity as follows.

$$\text{minimize } \varphi(x) = \sum_{h \in A} \xi_h x_h \quad (6.1)$$

subject to:

$$\sum_{\substack{h: a_h = (n, j) \in A; \\ j = 1, 2, \dots, |N|; j \neq n}} x_h - \sum_{\substack{h: a_h = (i, n) \in A; \\ i = 1, 2, \dots, |N|; i \neq n}} x_h = b(n); \quad (6.2)$$

where:

- $a_h = (i, n)$ refers to an arc incident into node n ;
- $a_h = (n, j)$ denotes an arc incident out of node n ;
- $n = 1, 2, \dots, |N|$

$$x_h \leq c_h; \quad \forall_h a_h \in A \quad (6.3)$$

$$x_h \in \mathcal{Z}_+; \quad \forall_h a_h \in A \quad (6.4)$$

Formula (6.2) represents the flow conservation constraints according to Kirchhoff's law [27]. In particular, the first part of the formula (6.2) stands for the total outflow of node n , while its second part refers to the total inflow of that node. In general, formula (6.2) says that, for each node n , the total outflow minus the total inflow gives the supply/demand value of that node. Formula (6.3) provides the flow bound constraints according to the upper limit c_h on the maximum flow at link a_h , while constraint (6.4) limits values of x_h variables to nonnegative integer numbers. It is worth noting that this assumption of the integrality of x_h also holds in other scenarios, as integer values can be obtained as a result of the transformation of the input values by multiplying them by a reasonably large integer number [1]. Two following aspects are worth mentioning:

1. The part of the optimization problem given by formulas (6.1)–(6.2) can be rewritten in a matrix form as follows:

$$\text{minimize } \varphi(x) = \mathbf{c}\mathbf{x} \quad (6.5)$$

subject to:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad (6.6)$$

where:

- \mathbf{c} is a row vector of unitary costs ξ_h for arcs a_h . The number of elements in this vector equals the number of arcs $|A|$ in graph G . Each h -th element in this vector stores the unitary cost of arc a_h ;
 - \mathbf{x} is a column vector of variables x_h , each variable x_h used to represent the amount of flow served by arc $a_h=(i, j)$. The number of elements in vector \mathbf{x} is, therefore, also equal to $|A|$;
 - \mathbf{A} is a node-arc incidence matrix of $|N| \cdot |A|$ size. In this matrix, a given element located in the n -th row and h -th column is equal to 1 if arc a_h is sourced at node n , and -1 if arc a_h is targeted at node n . All other values of $\mathbf{A}[n, h]$ are equal to 0;
 - \mathbf{b} is a column vector of supply/demand values of nodes n . The number of elements in this vector is thus equal to the number of nodes $|N|$ in graph G . A particular n -th element in this vector stores the supply/demand value for node n .
2. By allowing only values of 1 and -1 to be assigned as supply/demand values $b(n)$, we, in fact, transform the minimum cost flow problem into the shortest path problem. In this problem, the objective is to determine the least-cost (in terms of ξ_h values) path of unitary capacity between a particular source node s , for which

$b(s) = 1$ and the related destination node t with $b(t)$ set to -1 . This variant of the minimum cost flow problem has been adapted to model transmission paths in communication networks.

Also, it is worth noting that if we set $b(i)$ to $|N| - 1$, while all the other values of $b(j)$ are set to -1 , the solution to such a problem will consist of the shortest paths from node i to all other nodes j in the network [1].

6.2 The Network Model Applied in This Chapter

The optimization models discussed in this chapter are dedicated to a common architecture of wide-area optical transport networks (OTNs) with circuit-switched data transmission [22]. By utilizing dense wavelength division multiplexing (DWDM), optical links provide sets of nonoverlapping transmission channels, each channel associated with a given wavelength λ_i . This, in turn, enables multi-hop lightpaths to operate in parallel using different link channels. For a detailed specification of the technology-related properties of OTNs, the reader is referred to important books in this area, such as the one by Mukherjee [18], by Ramaswami et al. [26], or by Chatterjee and Oki [4].

The architecture of circuit-switched optical transport networks is modeled here by a directed graph $G(N, A)$, where N is the set of nodes representing the optical cross connects (OXC), while A is the set of arcs referring to optical communication links.

The following assumptions hold unless stated otherwise. A given bidirectional optical link is represented here by a pair of two oppositely directed arcs $a_h = (i, j)$ and $a_{h'} = (j, i)$. We assume single-mode optical links, each link offering Λ_h (e.g., 80) transmission channels in both directions. These channels are identified by wavelengths λ_i and are assumed to offer equal capacity (often referred to as *modular capacity* [22]), which, in practice, denotes values of, e.g., 10, 40, or 100 Gbps per channel [32]. In our modeling, we assume an equal number Λ of channels available at each network link, i.e., $\forall_h \Lambda_h = \Lambda$.

In our model, each OXC is assumed to offer a full wavelength conversion capability, implying that any optical signal incoming via any input port of a given OXC at any wavelength λ_i can be switched at any output port of that OXC to any output wavelength λ_j . For each multi-hop transmission path, all-optical switching of signal is assumed at each transit node. The electronic processing of a signal requiring the respective optical-to-electrical (O/E) and electrical-to-optical (E/O) conversions needs to take place at the end nodes of each lightpath. Therefore, data traveling a sequence of lightpaths is processed electronically at the end node of a given lightpath (e.g., by an IP router) before being inserted into the next lightpath.

The set of transmission demands D consisting of demands d_r , $r = 1, 2, \dots, |D|$; $1 < r \leq |N| \cdot (|N| - 1)$ defined as triples (s_r, t_r, c_r) is given, where s_r and t_r denote the source and destination nodes of a demand d_r , while c_r is the transmission capacity requested by that demand. In models discussed in this chapter, the assumption of

unitary capacity requested for demand d_r is commonly made, which is represented by $c_r=1$. Also, we assume that working paths need to be 100% restorable, i.e., the capacity required for backup paths is the same as for the related working paths.

Since the capacity of each arc a_h is limited by a certain number Λ_h of available transmission channels, problems of resilient routing are addressed in this chapter for *capacity-constrained* networks, which is, in fact, a major reason for these problems to belong to the class of complex (\mathcal{NP} -hard) optimization problems [22].

6.3 Finding the Set of Working Paths

In this part, we provide the formulation for the basic optimization problem of finding single end-to-end communication paths from a given set of demands D , i.e., aimed at determining only the working paths. For each demand d_r , its working path is to be established between a given pair of source node s_r and destination node t_r in a way that the overall cost of deployment of all working paths expressed by the total amount of resources (link channels) to be reserved for these paths is minimized.

The problem of determining single unicast communication paths between pairs of source s_r and destination d_r nodes for demands from D in capacity-constrained networks is \mathcal{NP} -hard [21], meaning that no polynomial algorithm has been proposed so far to find the optimal solution. Since in the considered problem, the end-to-end working paths compete for constrained capacity of links, finding the optimal solution requires calculation of these paths jointly instead of sequential processing (the latter case could, in turn, be handled by a heuristic scheme).

Model 6.1 for Determination of a Set of Working Paths (Node–Link Notation)

Symbols

$G(N, A)$	Graph representing a directed network
N	Set of nodes representing network nodes (their number is given by $ N $)
A	Set of arcs a_h representing network links (their number is given by $ A $)
D	Set of demands d_r (the number of demands is given by $ D $)
r	Index of a demand d_r ; $1 < r \leq N \cdot (N - 1)$
$s_r(t_r)$	Source (destination) node of demand d_r
c_h	Total capacity of arc a_h expressed by integer units (referring to the number of transmission channels, each channel of equal capacity)

Constants

ξ_h	Cost per unit flow of arc $a_h = (i, j)$
---------	--

Variables

$x_{r,h}$	Takes the value of 1, if arc $a_h = (i, j)$ is traversed by a working lightpath of r -th demand; 0 otherwise.
-----------	---

Objective

For a given set of demands D , it is to find the single communication paths between the respective source and destination nodes of demands to minimize the total cost defined by Eq. 6.7.

$$\text{minimize } \varphi(x) = \sum_{r=1}^{|D|} \sum_{h=1}^{|A|} \xi_h \cdot x_{r,h} \quad (6.7)$$

Constraints

(a) Flow conservation law (Kirchhoff's law) for working lightpaths:

$$\sum_{\substack{h:a_h=(n,j) \in A; \\ j=1,2,\dots,|N|; j \neq n}} x_{r,h} - \sum_{\substack{h:a_h=(i,n) \in A; \\ i=1,2,\dots,|N|; i \neq n}} x_{r,h} = \begin{cases} 1 & \text{if } n = s_r \\ -1 & \text{if } n = t_r \\ 0 & \text{otherwise} \end{cases} \quad (6.8)$$

where:

$a_h = (i, n)$ refers to an arc incident into node n ;
 $a_h = (n, j)$ denotes an arc incident out of node n ;
 $r = 1, 2, \dots, |D|$;
 $n = 1, 2, \dots, |N|$.

(b) On finite arc capacity.

$$\sum_{r=1}^{|D|} x_{r,h} \leq c_h \quad (6.9)$$

where: $h=1, 2, \dots, |A|$.

(c) On the allowed values of variables:

$$x_{r,h} \in \{0; 1\} \quad (6.10)$$

where: $r = 1, 2, \dots, |D|$, $h = 1, 2, \dots, |A|$.

The optimization model defined by formulas (6.7)–(6.10) belongs to binary linear programming (BLP) models, as all its formulas are linear, while all variables are allowed to take binary values.

It is worth noting that variables $x_{r,h}$ do not need to be characterized by an additional symbol referring to the channel index since, due to the assumption of full wavelength conversion possible at each network node, it is not necessary to monitor which channel of a given link was finally assigned to a given path determined for demand d_r . Another observation is that this model is equivalent to the min-cost

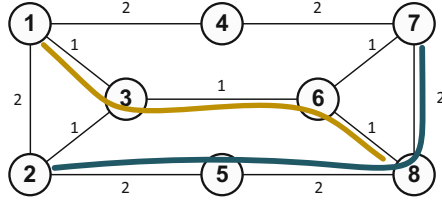


Fig. 6.1 Illustration of two shortest paths obtained as a solution to the shortest path problem specified by Model 6.1. The nominal capacity c_h of all links is assumed to be equal to 1. Link costs ξ_h are provided next to the respective links.

flow model given by formulas (6.1)–(6.4) if allowing only values of 1 and -1 to be assigned to the related supply/demand values of $b(i)$ in that model.

Figure 6.1 illustrates two shortest paths established between a pair of nodes 1 and 8 as well as between a node pair 2 and 7 following the assumptions of Model 6.1. The requested capacity for each path was unitary. Also, to analyze the model's behavior in scenarios of heavily constrained link resources, all network links were assumed to offer only unitary capacity (i.e., c_h was assumed to be equal to 1 for all network links).

For the topology from Fig. 6.1 and the considered demands, the shortest possible paths are as follows: (1, 3, 6, 8) and (2, 3, 6, 7), each of the total cost of 3. However, due to constrained network resources, only one of these paths can traverse link (3, 6), since for each path there was a request to reserve the entire nominal capacity of link (3, 6). Therefore, as a result, only the path between nodes 1 and 8 traverses that link. The other path (path between nodes 2 and 8) has to follow a sequence of network elements, i.e., (2, 5, 8, 7) of the total cost equal to 6. Therefore, the optimal value of the objective function minimizing the total cost of both paths $\varphi(x)$ is $3+6=9$.

In the model given above defined for directed links and demands, values of variables $x_{r,h}$ denote particular flows for demand d_r on a directed link represented by arc a_h . For any given network node n , the relation between the input and output flows follows the so-called flow conservation law represented by formula (6.8). Therefore, due to the association of variables $x_{r,h}$ with directed links represented by arcs a_h , the optimization model given by formulas (6.7)–(6.10) is an example of the *node-link formulation* [22]. Recall that this node-link relation of the network topology is also reflected by elements matrix \mathbf{A} in formula (6.6). This formulation will be used in the remaining part of this chapter.

Another common variant used in modeling routing problems is the *link-path formulation* [22] reflecting the association of end-to-end communication paths with network links. This formulation is valid for both directed and undirected links. The link-path formulation corresponding to Model 6.1 is given by formulas (6.11)–(6.14).

Model 6.2 for Determination of a Set of Working Paths (Link-Path Notation)**Symbols**

A	Set of arcs a_h representing network links (their number is given by $ A $)
D	Set of demands d_r (the number of demands is given by $ D $)
r	Index of a demand d_r ; $1 < r \leq N \cdot (N - 1)$
c_h	Capacity available at arc a_h expressed by integer units (referring to the number of transmission channels, each channel of equal capacity)
p	Index of a path from the set P of precomputed paths

Constants

$v_{r,p}$	Cost of a precomputed path p to serve demand d_r
$\delta_{h,r,p}$	Is equal to 1, if arc a_h is used by path p to serve demand d_r

Variables

$x_{r,p}$	Takes the value of 1, if path p is selected as a transmission path for demand d_r ; 0 otherwise
-----------	---

Objective

$$\text{minimize } \varphi(x) = \sum_{r=1}^{|D|} \sum_{p=1}^{|P|} v_{r,p} x_{r,p} \quad (6.11)$$

Constraints

(a) Demand constraints:

$$\sum_{p=1}^{|P|} x_{r,p} = 1; \quad r = 1, 2, \dots, |D| \quad (6.12)$$

(b) Capacity constraints:

$$\sum_{r=1}^{|D|} \sum_{p=1}^{|P|} \delta_{h,r,p} x_{r,p} \leq c_h; \quad h = 1, 2, \dots, |A| \quad (6.13)$$

(c) On allowed values:

$$x_{r,p} \in \{0; 1\}; \quad r = 1, 2, \dots, |D|; \quad p = 1, 2, \dots, |P| \quad (6.14)$$

A clear advantage of the link-path formulation is a smaller number of variables $x_{r,p}$ than in the case of the corresponding node-link notation because the number of variables $x_{r,p}$ depends only on the number of demands and the number of candidate paths p precomputed for each demand d_r before solving the optimization

problem. Therefore, it leads to better scalability when increasing the problem size, as it does not associate variables $x_{r,p}$ with characteristics of the network structure. However, it is essential to note that by operating on a limited set of pre-calculated candidate paths p for each demand d_r , the optimization model based on the link-path formulation may lead to results relatively far from the optimum assured by the corresponding node-link notation.

6.4 Finding the Set of Pairs of Disjoint Working and Protection Paths

In this section, we describe the formulations of the LP model to solve the optimization problem of establishing pairs of end-to-end disjoint paths under dedicated protection (i.e., path protection scheme) characterized by the total minimal cost of all paths. Therefore, for each demand d_r from D , the model needs to ensure one backup path, providing the end-to-end protection for the related working path. Backup paths are assumed here to be assigned explicitly to specific working paths without sharing link capacity among several backup paths (i.e., a classical dedicated protection scheme). As discussed earlier in this book, a single backup path can take over the role of the related working path in scenarios of single failures of network elements. It can also be able to operate in scenarios of failures of several network elements provided that at least one of the paths of a considered demand d_r (i.e., either a working path or a backup path) remains operational.

Failures of single network elements are indeed the most common [24]. In particular, scenarios of single node failures can be handled by backup paths being node-disjoint with the related working paths (i.e., having no common transit nodes with their working paths), as illustrated in Fig. 6.2a. This is the aim of the optimization model presented in our work [17] and discussed in the following Sect. 6.4.1. The case of establishing pairs of link-disjoint paths for protection against failures of single links (see example in Fig. 6.2b) is, in turn, analyzed in Sect. 6.4.2. As models in both Sects. 6.4.1 and 6.4.2 are provided for the assumption of unitary capacity ($c_r=1$) required for paths of all demands, they are examples of BLP problems.

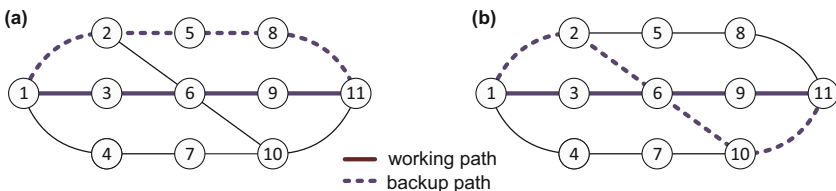


Fig. 6.2 Illustration of a pair of node-disjoint working and backup paths (a), and the related link-disjoint variant (b)

In general, protection schemes relevant for scenarios of node failures are also proper to assure protection against failures of links since, from the graph theory point of view, a failure of a node is equivalent to a failure of all its incident links. Therefore, the model from Sect. 6.4.2 is indeed a simplified version of the model discussed in Sect. 6.4.1.

The problem of providing the resilient routing for a set of demands by backup paths being disjoint with the respective working paths in capacity-constrained networks was shown to be \mathcal{NP} -hard in [25]. Therefore, the use of heuristic schemes, such as those based on Suurballe's algorithm [30, 31], or its modification—Bhandari's approach [2, 3], is often needed for larger problem instances.

6.4.1 Nodal Disjointness of Working and Protection Paths

The optimization model, adequate in determining pairs of end-to-end node-disjoint paths, extends the previous Model 6.1 by the additional formulations related to backup paths as follows.

Model 6.3 for Calculation of Pairs of Node-Disjoint Paths

Symbols

The list of symbols is the same as for Model 6.1.

Constants

The list of constants is the same as for Model 6.1.

Variables

The list of variables is the same as for Model 6.1 and is additionally extended by:

$y_{r,h}$ Takes the value of 1, if arc $a_h = (i, j)$ is traversed by a backup lightpath of r -th demand; 0 otherwise.

Objective

It is to find pairs of node-disjoint working and backup paths between the respective source and destination nodes of demands for the scenario of a single node failure in a way to minimize the total cost given by $\varphi(x)$ in Eq. 6.15.

$$\text{minimize } \varphi(x) = \sum_{r=1}^{|D|} \sum_{h=1}^{|A|} \xi_h(x_{r,h} + y_{r,h}) \quad (6.15)$$

Constraints

(a) Flow conservation constraints (Kirchhoff's law):

(a1) For working lightpaths: the same as in Model 6.1.

(a2) For backup lightpaths:

$$\sum_{\substack{h:a_h=(n,j)\in A; \\ j=1,2,\dots,|N|;j\neq n}} y_{r,h} - \sum_{\substack{h:a_h=(i,n)\in A; \\ i=1,2,\dots,|N|;i\neq n}} y_{r,h} = \begin{cases} 1 & \text{if } n = s_r \\ -1 & \text{if } n = t_r \\ 0 & \text{otherwise} \end{cases} \quad (6.16)$$

where:

$a_h = (i, n)$ refers to an arc incident into node n ;

$a_h = (n, j)$ denotes an arc incident out of node n ;

$r = 1, 2, \dots, |D|$;

$n = 1, 2, \dots, |N|$.

(b) On finite arc capacity:

$$\sum_{r=1}^{|D|} (x_{r,h} + y_{r,h}) \leq c_h \quad (6.17)$$

where: $h = 1, 2, \dots, |A|$.

(c) To guarantee the nodal disjointness of working and backup paths of a demand:

$$\sum_{\substack{h:a_h=(n,j)\in A; \\ j=1,2,\dots,|N|;j\neq n}} (x_{r,h} + y_{r,h}) \leq 1 \quad (6.18)$$

$$\sum_{\substack{h:a_h=(i,n)\in A; \\ i=1,2,\dots,|N|;i\neq n}} (x_{r,h} + y_{r,h}) \leq 1 \quad (6.19)$$

where:

n is used here to represent transit nodes (i.e., $n \neq s_r$ and $n \neq t_r$, which are valid for working and backup paths consisting of at least two arcs);

$r = 1, 2, \dots, |D|$.

(d) On the allowed values of variables:

$$x_{r,h}, y_{r,h} \in \{0; 1\} \quad (6.20)$$

where: $r = 1, 2, \dots, |D|$, $h = 1, 2, \dots, |A|$.

The inclusion of backup paths in the design of communication paths is reflected by variables $y_{r,h}$ in the objective function (see Eq. 6.15)—as now it also comprises the capacity reserved for backup paths, in formula (6.16) referring to flow conservation constraints for backup paths, formula (6.17) referring to the finite capacity of network links, as well in formulas (6.18)–(6.19) added to assure the nodal disjointness of related working and backup paths.

6.4.2 Link Disjointness of Working and Protection Paths

The model to determine the optimal solution to the problem of establishing pairs of link-disjoint paths in a way to minimize the overall network cost is similar to Model 6.3 described earlier in this chapter, with the only exception referring to the constraints to guarantee the link disjointness (instead of the nodal disjointness) of the related working and backup paths. The model is defined as follows.

Model 6.4 for Calculation of Pairs of Link-Disjoint Paths

Symbols

The list of symbols is the same as in Model 6.3.

Constants

The list of constants is the same as in Model 6.3.

Variables

The list of variables is the same as in Model 6.3.

Objective

It is to find pairs of end-to-end link-disjoint working and backup paths for demands from D defined between the respective source and destination nodes to provide protection of working paths in scenarios of single link failures in a way to minimize the total cost represented by formula (6.15).

Constraints

The set of constraints is the same as in Model 6.3 except for constraints on nodal disjointness (formulas (6.18)–(6.19)) replaced by the following formula (6.21) to assure the link disjointness of the related pairs of working and backup paths:

$$x_{r,h} + y_{r,h} \leq 1 \quad (6.21)$$

where: $r = 1, 2, \dots, |D|$; $h = 1, 2, \dots, |A|$.

6.5 Optimization Model for the “A Priori” Sharing of Backup Path Resources

In this section, we discuss the optimization approach enabling several backup paths to mutually share capacity at certain links traversed by these paths. The respective formulations presented here can be used as an extension to various optimization models for establishing working and protection paths originally proposed without backup path sharing. The problem of resilient routing with shared backup paths extends to the conventional problem of resilient routing, e.g., as defined by Model 6.3 in this chapter. Therefore, it is also \mathcal{NP} -hard.

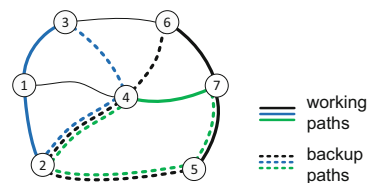
Since this section describes the approach to sharing the backup link capacity to be applied at the time of determination of working and backup paths, it is referred to as the “a priori” sharing here, as opposed to another variant of “a posteriori” backup path sharing, introduced by us in [23] and described later in Sect. 6.6. The concept of “a priori” sharing of backup paths has been analyzed in detail, e.g., in [11, 12, 25].

In this section, the use of the “a priori” sharing concept is explained for the path protection scheme (where for each working path, there is one end-to-end backup path) considering a scenario of a single node failure (thus, for the scheme of pairs of end-to-end node-disjoint working and backup paths).

As discussed in Chap. 4, sharing of link capacity among several backup paths at a given network link is possible when these backup paths protect mutually node-/link-disjoint segments of working paths (when protecting against failures of single nodes/links, respectively). In the case of a path protection scheme, a segment of a working path naturally means the entire working path, as illustrated in Fig. 6.3. This sharing condition has been formulated to avoid the need to activate more than one backup path from the set of backup paths sharing capacity at a given link. If sharing is appropriately applied, then every working path is guaranteed to be 100% restorable in the context of their capacity c_r in any failure scenario of a single network element.

Contrary to dedicated protection schemes, where the total capacity requested for each demand d_r (i.e., equal to c_r) has to be reserved exclusively for the related backup path at all links traversed by that path, under shared protection, the cost ζ_h of installing a given backup path at a given link reflects only the cost of allocating for this backup path the extra capacity (i.e., which could not be shared at that link), as given in formula (6.22).

Fig. 6.3 Illustration of the fulfillment of the condition for sharing of capacity at the link (2, 4) by backup paths protecting mutually disjoint working paths



$$\zeta_h = \begin{cases} \varepsilon & \text{if } c_r \leq sh_h^{(r)} \\ (c_r - sh_h^{(r)}) \cdot \xi_h & \text{if } c_r > sh_h^{(r)} \text{ and } \bar{c}_h \geq c_r - sh_h^{(r)} \\ \infty & \text{otherwise} \end{cases} \quad (6.22)$$

where:

c_r is the capacity requested for r -th demand;

\bar{c}_h is the unused capacity of arc $a_h = (i, j)$;

ξ_h is a unitary cost of arc a_h in working path computations;

$sh_h^{(r)}$ is the capacity reserved so far at a_h that may be shared with respect to the backup path of r -th demand.

To apply the “a priori” sharing of backup paths to Model 6.3 given by formulas (6.15)–(6.20), and formula (6.8) originally defined for the case of dedicated protection (i.e., with no backup path sharing), certain modifications to that model are necessary. In particular, the objective function given by Eq. 6.15 has to be replaced by Eq. 6.23, while the set of constraints has to be extended by formulas (6.24)–(6.28). Therefore, the extended model with “a priori” backup path sharing is defined as follows.

Model 6.5 to Find Pairs of Node-Disjoint Working and Protection Paths with Backup Path Sharing

Symbols

The list of symbols is the same as in Model 6.3.

Constants

The list of constants is the same as in Model 6.3.

Variables

The list of variables is the same as in Model 6.3 and is extended by the following ones:

b_h An integer variable determining how much extra capacity has to be reserved for backup paths at arc a_h (therefore related to metric ξ_h in Eq. 6.23).

$b_{r,h,g}$ A binary variable to indicate whether for demand d_r , the failed primary path traverses arc a_g , and the corresponding backup path traverses arc a_h .

$b_{h,g}$ An integer variable representing the total capacity needed for backup paths at arc a_h in the case of shared protection provided for working paths traversing the failed arc a_g .

Objective

It is to find pairs of end-to-end node-disjoint working and backup paths for demands from D to assure survivability in scenarios of single node failures by applying “a priori” sharing of backup paths in a way to minimize the total cost defined by Eq. 6.23.

$$\text{minimize } \varphi(x) = \sum_{r=1}^{|D|} \sum_{h=1}^{|A|} \xi_h x_{r,h} + \sum_{h=1}^{|A|} \xi_h b_h \quad (6.23)$$

Constraints

The list of constraints of the model includes formulas (6.16)–(6.20), formula (6.8), as well as is additionally extended by constraints referring to “a priori” shared protection defined by formulas (6.24)–(6.28).

$$x_{r,g} + y_{r,h} \leq 1 + b_{r,h,g}; \quad r = 1, 2, \dots, |D|; h = 1, 2, \dots, |A|; g = 1, 2, \dots, |A|; g \neq h \quad (6.24)$$

$$2b_{r,h,g} \leq x_{r,g} + y_{r,h}; \quad r = 1, 2, \dots, |D|; h = 1, 2, \dots, |A|; g = 1, 2, \dots, |A|; g \neq h \quad (6.25)$$

$$b_{h,g} = \sum_{r=1}^{|D|} b_{r,h,g}; \quad h = 1, 2, \dots, |A|; g = 1, 2, \dots, |A|; g \neq h \quad (6.26)$$

$$b_{h,g} \leq b_h; \quad h = 1, 2, \dots, |A|; g = 1, 2, \dots, |A|; g \neq h \quad (6.27)$$

$$b_{h,g}, b_h \in \mathcal{Z}_+; \quad b_{r,h,g} \in \{0, 1\} \quad (6.28)$$

In particular, formula (6.24) assures that in the case both variables of left-hand side are equal to 1 (which implies that a given working path traverses the failed arc a_g , while the corresponding backup path traverses arc a_h), then variable $b_{r,h,g}$ must also be equal to 1 (i.e., it must indicate this relation).

Formula (6.25), in turn, guarantees that if at least one of the variables $x_{r,g}$ and $y_{r,h}$ is equal to 0 (i.e., if arcs a_g and a_h are not used in parallel by the respective working and backup paths of demand d_r), then $b_{r,h,g}$ should be equal to 0—in order not to indicate the mentioned relation. Formula (6.26) refers to the total amount of spare capacity required at a_h in a particular scenario of arc a_g failure, while formula (6.27) is to provide constraints on the maximum amount of spare capacity needed to be reserved at a_h for all failure scenarios. Formula (6.28) determines the sets of allowed values for variables added to this model.

6.6 Optimization Model for the “A Posteriori” Sharing of Backup Path Resources

This section discusses the formulation of an optimization model for another approach to sharing the backup path resources, called “a posteriori” sharing, introduced by us in [23]. The main idea behind this scheme is to ensure that the length of the backup paths is not increased due to backup path sharing. As shown in [23], the increased length of backup paths (even up to 40%, compared to the case of a dedicated protection scheme) is indeed a clear disadvantage of the conventional “a priori” sharing. To achieve this, contrary to the “a priori” sharing applied at the time of establishing both working and protection paths described in Sect. 6.5, our “a posteriori” sharing procedure is executed after the process of calculation of working and backup paths is completed.

In the “a posteriori” scheme, sharing is applied locally at every single network link and results in the rearranging of the assignment of channels initially assigned to backup paths during path calculation. This also means freeing some of the channels assigned initially at a given link to backup paths at their calculation time. As a result, backup paths traverse the links they utilized before applying the sharing procedure, which guarantees that the backup paths remain the shortest possible, as in the case of dedicated protection. The capacity efficiency of the “a posteriori” scheme follows from the properties of the sharing technique itself, described later in this section.

It is important to note that the “a posteriori” sharing the capacity of backup path links can be applied at each link a_h independently in any order only in the case of a full wavelength conversion available at each network node. Otherwise, in the case of restrictions on possible conversions at certain optical cross connects between input wavelengths λ_i and the related output wavelengths λ_j , our “a posteriori” sharing scheme would also need to reflect these limitations, as discussed in detail in [23].

In the case of a full wavelength conversion possible at each network node, sharing backup path capacities can be applied at links a_h in any order after finishing the working and backup paths calculation. Clearly, for any two backup paths protecting non-disjoint segments of working paths, a failure of a single network element can affect both these working paths. The related backup paths might then need to be activated simultaneously. That is why, to provide 100% restorability of working path capacity, such *conflicting backup paths* cannot share common capacity at any network link.

The purpose of “a posteriori” sharing of backup paths at each arc a_h is, therefore, to:

- (1) Divide the set of backup paths B_h traversing arc a_h into subsets B_h^c in a way that each subset B_h^c contains backup paths that may share the same capacity at that arc (i.e., if these backup paths protect mutually disjoint segments of the related working paths).

- (2) Assign the same capacity at arc a_h (e.g., represented by a given DWDM link channel) to backup paths belonging to the same subset B_h^c .

To optimize the efficiency of the “a posteriori” sharing technique, the number of subsets B_h^c for each arc a_h must be minimized, as it denotes the number of resource units (e.g., DWDM link channels) finally assigned to backup paths at arc a_h after executing the “a posteriori” sharing procedure.

In the case of a full wavelength conversion possible at all network nodes, the optimization model to determine the optimal “a posteriori” sharing of link capacity among backup paths traversing a given arc a_h providing 100% restorability of working path capacity can be formulated as follows.

Model 6.6 to Determine “A Posteriori” Sharing of Backup Paths at Arc a_h (Full Wavelength Conversion Capability)

Symbols

$w = 1, 2, \dots, |B_h|$ Indices of backup paths $\hat{\pi}_w$ traversing a given link
 $c = 1, 2, \dots, \Lambda$ Indices of channels at a_h available for backup paths

Variables

\hat{x}_w^c Takes the value of 1, if backup path $\hat{\pi}_w$ is assigned channel c at a_h ;
 0 otherwise.
 b^c Equals 1, if channel c is assigned to any backup path at a_h ; 0 otherwise.

Objective

For backup paths traversing a given arc a_h , it is to determine the optimal sharing of link channels by these backup paths in a way to minimize the total number of channels assigned to backup paths at a_h given by Eq. 6.29.

$$\text{minimize } \varphi(x) = \sum_{c=1}^{\Lambda} b^c \quad (6.29)$$

Constraints

- (a) On the assignment of only one channel c to each backup path $\hat{\pi}_w$:

$$\sum_{c=1}^{\Lambda} \hat{x}_w^c = 1 \quad (6.30)$$

where: $w = 1, 2, \dots, |B_h|$.

- (b) On the assignment of different channels for conflicting backup paths $\hat{\pi}_w$ and $\hat{\pi}_{w'}$:

$$\hat{x}_w^c + \hat{x}_{w'}^c \leq b^c \quad (6.31)$$

for each pair of conflicting backup paths \hat{x}_w^c and $\hat{x}_{w'}^c$.

(c) On not assigning to backup paths channels at arc a_h already reserved for working paths:

$$\sum_{w=1}^{|B_h|} \sum_{c=1}^{\Lambda} \hat{x}_w^c = 0 \quad (6.32)$$

where indices c refer to channels already reserved for working paths.

(d) On the allowed values of variables:

$$\hat{x}_w^c \in \{0; 1\}; b^c \in \{0; 1\}; \quad w = 1, 2, \dots, |B_h|; c = 1, 2, \dots, \Lambda \quad (6.33)$$

To apply the “a posteriori” sharing of backup paths, the program defined by Eq. 6.29 and constraints provided by formulas (6.30)–(6.33) should be executed for all arcs a_h of network graph G in any order. For each arc a_h , the objective of the model provided by Eq. 6.29 is to minimize the number of channels allocated at a_h to backup paths while assuring that each backup path traversing arc a_h is assigned only one channel (formula (6.30)), any two conflicting backup paths at that arc are assigned different channels (formula (6.31)), and, in general, backup paths are assigned channels not reserved for working paths (formula (6.32)). Since all variables used in this model are binary (formula (6.33)) and all functions are linear, this model belongs to the class of binary linear programming (BLP) optimization models.

The Computational Complexity of “A Posteriori” Backup Path Sharing Scheme

The problem addressed by our “a posteriori” scheme of backup path sharing at a given arc a_h belongs to the class of \mathcal{NP} -hard problems since it is equivalent to the problem of *vertex-coloring* of a *graph of conflicts* known to be \mathcal{NP} -hard [10]. In a graph of conflicts $\Gamma_h(V, E)$:

- V denotes a set of vertices v_i representing backup paths traversing arc a_h .
- E is a set of edges $e_m=(k, l)$ representing conflicts between vertices v_k and v_l . In the case of the “a posteriori” scheme, every edge $e_m=(k, l)$ represents a conflict between the respective backup paths $\hat{\pi}_k$ and $\hat{\pi}_l$, meaning that these backup paths need to be assigned different channels at arc a_h since the respective segments of working paths being protected by these backup paths are not disjoint.

Figure 6.4b illustrates the example graph of conflicts for backup paths traversing a link (8, 10) in the network from Fig. 6.4a. Figure 6.4b illustrates five conflicts referring to backup paths traversing link (8, 10) from Fig. 6.4a.

Indeed, a vertex-coloring of the graph of conflicts is a mapping $\delta: V \rightarrow C$, where V is a set of vertices of Γ_h , while C is a finite set of colors (integer numbers)

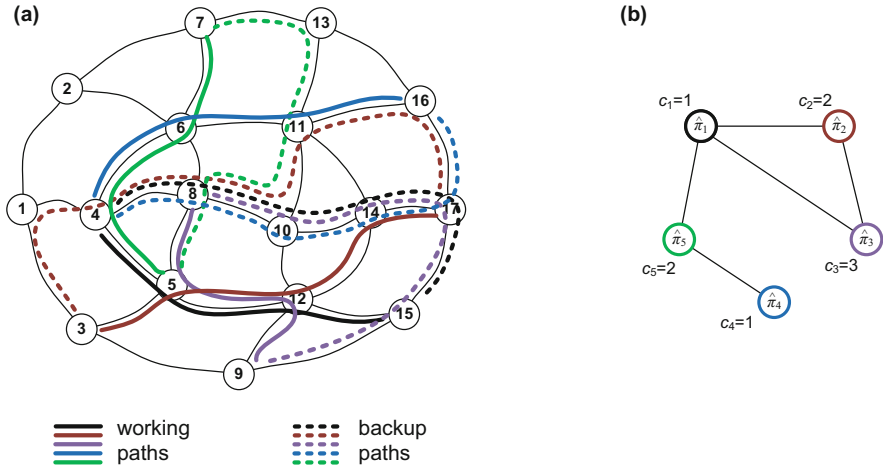


Fig. 6.4 An example network topology with five multi-hop connections (a); the related graph of conflicts referring to the network link (8, 10) with the assignment of colors c_i to its vertices (b)

assigned to vertices v_i in a way that any two neighboring vertices v_k and v_l in Γ_h , i.e., connected by a direct edge $e_m = (k, l)$, are assigned different colors.

The equivalence of our “a posteriori” sharing scheme to the vertex-coloring of a graph of conflicts Γ_h follows from a direct correspondence of colors c with indices of channels finally assigned at a_h to backup paths. The objective of the backup path sharing procedure to minimize the number of channels assigned to backup paths at a given network link is equivalent to determining the minimum number of colors needed to color the vertices of Γ_h , i.e., its *chromatic number* $\chi(\Gamma_h)$ known to be \mathcal{NP} -hard [10].

Example

As a result of executing the “a posteriori” sharing procedure, only three channels need to be allocated to backup paths at link (8, 10) in Fig. 6.4a, since when applying the vertex-coloring of a graph of conflicts Γ_h from Fig. 6.4b, three colors $c \in \{1, 2, 3\}$ are needed to be assigned to vertices of Γ_h . Therefore, after executing the “a posteriori” sharing procedure for the link (8, 10), the amount of capacity needed for backup paths at that link was decreased to $3/5 = 60\%$ of capacity originally required for the separate allocation of link capacity to backup paths.

As this optimization problem is \mathcal{NP} -hard, obtaining the optimal vertex-coloring for graphs of conflicts consisting of more than several vertices becomes computationally infeasible. Therefore, for larger graphs, a possible solution is to use one of the heuristic algorithms for graph coloring, such as the *largest first (LF)* described, e.g., in [10]. In the LF algorithm, vertices of Γ_h are first ordered descending their degrees and then assigned sequentially the lowest possible colors based on this ordering. Therefore, in LF, vertices of higher degrees receive colors before lower-degree vertices.

6.7 Protection Cycles (p -Cycles)

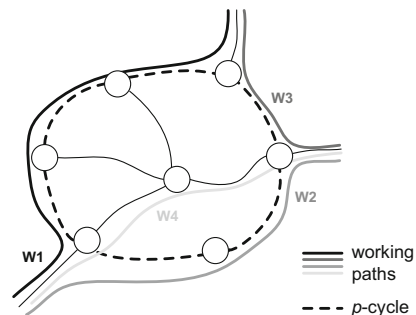
As described in Chap. 4, protection cycles (shortly p -cycles) are ring-like structures designed to protect segments of working paths either traversing these cycles (see paths W1-W3 in Fig. 6.5) or straddling them (as, e.g., path W4 in Fig. 6.5). Originally proposed by Grover and Stamatelakis in [9], they are designed to provide resilience of communications in circuit-switched networks (such as, for instance, DWDM networks) in scenarios of failures affecting the related working paths. Due to pre-configuration of p -cycles completed at the time of their installation in the network (i.e., before the occurrence of a failure), they can offer ring-like fast recovery of the affected working paths in parallel with a high level of link capacity efficiency (since one p -cycle can protect several mutually disjoint segments of working paths either traversing or straddling the p -cycle).

There is a rich set of optimization models available in the literature concerning the optimal allocation of link capacity to p -cycles focused on the minimization of the total amount of capacity assigned to all p -cycles in the network while assuring 100% of restorability (in terms of capacity assigned to p -cycles) for all working paths. These models can differ in terms of specific features of protection cycles (see, e.g., path-protecting p -cycles [13, 14], flow p -cycles [8], node-encircling p -cycles [6], or Hamiltonian p -cycles [28]). Also, the technological characteristics of networked systems often impose additional modeling assumptions.

In this section, we focus on modeling the p -cycles in a way that reflects the technology-related assumptions made in this chapter in Sect. 6.2 referring to the architecture of optical (DWDM) networks with single-fiber bidirectional optical links (each link offering a given number of Λ_h channels for a parallel transmission, represented by two unidirectional arcs in opposite directions), a full wavelength conversion possibility at each transit node, as well as assuming that each working path is allocated one channel exclusively at each traversed link.

In this context, a relatively close to our assumptions seems to be the proposal by Schupke et al. from [29]. In [29], the set of $|K|$ candidate p -cycles is first determined and is next used in the optimization model to set up the protection cycles for working paths. That model is re-formulated in this section as follows.

Fig. 6.5 Example illustration of a p -cycle configured for four working paths W1-W4



Model 6.7 to Determine the Set of p -Cycles for Working Paths

Symbols

$G(N, A)$	Graph representing a directed network.
N	Set of network nodes; $ N $ is the number of network nodes.
A	Set of arcs $a_h = (i, j)$ modeling directed network links. $ A $ is the number of arcs.
Δ_h	Number of DWDM channels available at arc a_h .
k	Index of a given p -cycle; $ K $ is the number of all pre-calculated p -cycles.

Constants

ξ_h	Cost of a unitary capacity (i.e., of a single channel) at arc a_h .
w_h	Number of channels already reserved at arc a_h for working paths.
$p_{k,h}$	Takes the value of 1, if arc a_h belongs to p -cycle k ; 0 otherwise.
$x_{k,h}$	Takes the value of 1, if a working path on arc a_h can be protected (i.e., is protectable) by p -cycle k ; 0 otherwise.

Variables

s_h	Number of channels reserved at arc a_h for protection cycles.
u_k	Number of units of link capacity needed for p -cycle k (i.e., the number of copies of p -cycle k returned as a solution, each copy of p -cycle k occupying a full channel on all its links).

Objective

It is to determine the assignment of p -cycles to the already established working paths, providing 100% restorability for each affected working path in a way to minimize the total cost of installing the protection cycles in the network given by Eq. 6.34.

$$\text{minimize } \varphi(x) = \sum_{h=1}^{|A|} \xi_h s_h \quad (6.34)$$

Constraints

(a) On the total capacity needed for p -cycles at each arc a_h :

$$s_h = \sum_{k=1}^{|K|} p_{k,h} u_k; \quad h = 1, \dots, |A| \quad (6.35)$$

(b) On providing 100% of restorability for all working paths at each arc a_h :

$$w_h \leq \sum_{k=1}^{|K|} x_{k,h} u_k; \quad h = 1, \dots, |A| \quad (6.36)$$

(c) On the total capacity available at arc a_h for working and protection paths:

$$w_h + s_h \leq \Lambda_h; \quad h = 1, \dots, |A| \quad (6.37)$$

(d) On the allowed values of variables:

$$s_h \in \mathcal{Z}_+; \quad h = 1, \dots, |A| \quad (6.38)$$

$$u_k \in \mathcal{Z}_+; \quad k = 1, \dots, |K| \quad (6.39)$$

Constraint (6.35) determines the total protection capacity needed at arc a_h , i.e., required for all u_k copies of all protection cycles k traversing arc a_h . Constraint (6.36), in turn, guarantees that channels already reserved at a_h for working paths are protected by p -cycles, i.e., by reserving at a_h for each selected p -cycle k the number of link capacity units equal to at least $x_{k,h} u_k$. Constraint (6.36) thus assures that the number of channels at a_h to be reserved for the related protection cycles is not lower than the number of channels w_h already reserved for working paths at a_h . Constraint (6.37) is to assure that the number of units of link capacity reserved for working and protection paths at each arc a_h does not exceed the nominal number of capacity units at a_h expressed by Λ_h . The determination of the sets of allowed values for variables is given by constraints (6.38)–(6.39).

It is important to note that although each working path is set up to serve a particular demand d_r , in this model, we do not need to refer to certain demands d_r when establishing the protection cycles for the related working paths. This is because the working paths are already established in the network, which allows us in the model above to operate at the aggregate number of protection units u_k needed for each p -cycle k instead of linking a certain p -cycle with a particular demand d_r . Also, this approach does not impose any constraints on the size of segments of working paths protected by certain p -cycles (these segments can be of any size, i.e., as small as one link of a working path and as large as entire working paths).

6.8 Mathematical Methods Used to Solve the Resilient Routing Optimization Problems

There are several representative monographs covering methods of mathematical programming available, e.g., by Wolsey [33], Murthy [19], Lasdon [15], Minoux [16], and Nemhauser and Wolsey [20]. However, concerning the adaptation of mathematical programming techniques to routing problems, particularly to resilient routing, the most notable one seems to be the book by Pióro and Medhi [22].

Linear programming models investigated in this chapter can be written in a general form as follows:

$$\begin{aligned}
 &\text{minimize } \varphi(\mathbf{x}) = \mathbf{c}\mathbf{x} \\
 &\text{subject to } \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b} \\
 &\mathbf{x} \geq \mathbf{0}
 \end{aligned} \tag{6.40}$$

where:

- \mathbf{c} is a row vector of size $1 \times n$ composed of cost coefficients ξ_j for variables x_j ;
- \mathbf{x} is a column vector of size $n \times 1$ composed of variables x_j ;
- \mathbf{A} is the $m \times n$ matrix, where m is the number of constraints, and n is the number of variables; each element $a_{i,j}$ in \mathbf{A} is the coefficient for variable j in constraint i ;
- \mathbf{b} is the right-hand side column vector of size $m \times 1$.

or shortly:

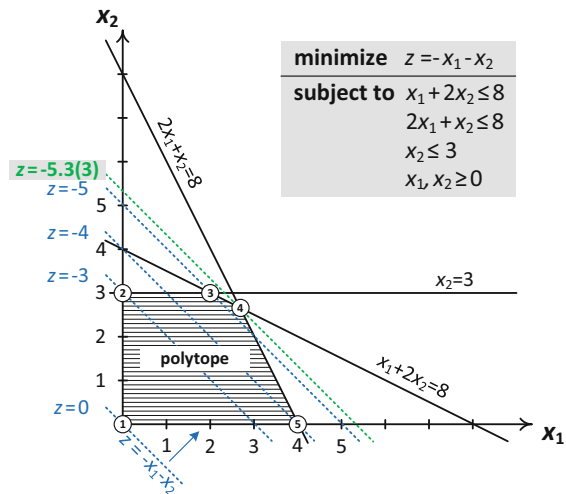
$$\min_{\mathbf{x}} \{ \mathbf{c}\mathbf{x} : \mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0} \}. \tag{6.41}$$

A given point \mathbf{x} satisfies inequalities $\mathbf{A}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \geq \mathbf{0}$, and then it is called a *feasible point*. All feasible points \mathbf{x} form a *polyhedron*.

A feasible point \mathbf{x} , which cannot be expressed as a convex combination $\sum_{k=1}^K \alpha_k \mathbf{x}^k$ of other feasible points $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^K$ (where $\alpha_k \geq 0, \sum_{k=1}^K \alpha_k = 1$), is called an *extreme point* or *vertex*. As discussed in [22], if the problem is bounded, then the minimum of $\varphi(\mathbf{x})$ exists, and it is achieved at at least one of vertices called the *optimal vertices*.

As shown in Fig. 6.6, for the example optimization model, the set of feasible solutions is bounded by five straight lines (thus forming a *polytope*, that is,

Fig. 6.6 An example of a two-variable linear programming problem with the respective optimal value of $-5\frac{1}{3}$ for the objective function z attributed to vertex 4 characterized by coordinates $(x_1, x_2) = (2\frac{2}{3}, 2\frac{2}{3})$



a bounded polyhedron). The vertices of this polytope are marked with circles numbered (1)-(5). In this example, contours of the objective function $z = -x_1 - x_2$ are marked by dashed lines. Fig. 6.6 shows that by moving the contours upward, the optimal (i.e., minimal) value of z equal to $-5\frac{1}{3}$ is achieved when the contour traverses vertex 4 characterized by coordinates $(x_1, x_2) = (2\frac{2}{3}, 2\frac{2}{3})$.

In the remaining part of this section, we discuss the basic methods for solving linear programming problems (simplex and column generation) and (mixed) integer programming problems (branch-and-bound).

6.8.1 Simplex Method

The simplex method invented by Dantzig [5] is a common technique for solving linear programming problems which are given in the standard form, i.e., including only equality constraints and nonnegative variables [15, 19] as given by formula (6.42).

$$\min\{\mathbf{c}\mathbf{x} : \mathbf{A}\mathbf{x} = \mathbf{b}; \mathbf{x} \geq \mathbf{0}\} \quad (6.42)$$

An LP problem formulation originally containing inequality constraints can be easily converted into a standard form by adding a nonnegative *slack variable* for each inequality constraint [22]. Also, each variable x_j , originally not constrained to be nonnegative, should be replaced by the difference of two nonnegative variables:

$$x_j = x'_j - x''_j, \quad x'_j \geq 0, \quad x''_j \geq 0 \quad (6.43)$$

For example, to express the LP problem illustrated in Fig. 6.6 in the standard form given by formula (6.42), we need to add three slack variables, one for each of three inequality constraints. The standard formulation of the LP problem from Fig. 6.6 is as follows:

$$\begin{aligned} \mathbf{minimize} \quad & z = -x_1 - x_2 \\ \mathbf{subject\ to} \quad & x_1 + 2x_2 + x_3 = 8 \\ & 2x_1 + x_2 + x_4 = 8 \\ & x_2 + x_5 = 3 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{aligned} \quad (6.44)$$

In this example, matrix \mathbf{A} has the form:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

In general, in the $m \times n$ matrix \mathbf{A} , if the set of constraints is redundant, some constraints are linear combinations of the others and thus can be removed without any impact on the solution. The *rank* of matrix \mathbf{A} is defined as the maximum number of linearly independent (nonredundant) rows (also equal to the maximum number of linearly independent columns). Therefore, for a nonredundant matrix \mathbf{A} , its rank equals m .

A subset of linearly independent m constraints forms the *basis*. A given basis is characterized by the unique point \mathbf{x} . This point \mathbf{x} is a *basic feasible solution* if its location satisfies all m constraints with equality. This point \mathbf{x} is one of the vertices of the polytope (generally, all basic feasible solutions are vertices of the polytope).

In the example model (6.44), all $m=3$ constraints are nonredundant. This is because matrix \mathbf{A} is a *canonical matrix*, as it includes a unitary matrix \mathcal{I} of $m \times m$ size with respect to the *basic vector* $\mathbf{x}=(x_3, x_4, x_5)$. Therefore, $\text{rank}(\mathbf{A})=3$. For the example model (6.44), this basic vector implies the basic solution $(x_1, x_2, x_3, x_4, x_5) = (0, 0, 8, 8, 3)$.

The simplex method operates on a canonical matrix \mathbf{A} . If matrix \mathbf{A} is initially non-canonical, it should first be transformed into a canonical form. The main idea behind the simplex method is to approach the best solution by iteratively visiting a sequence of vertices (i.e., moving from one basic solution to another basic solution) of the polyhedron. The simplex method walks along the edges of the polyhedron, moving from one vertex to another one characterized by a better value of the objective function z until the optimal vertex (i.e., characterized by the optimal value of z) is reached or an unbounded edge is encountered (implying that there is no solution to the problem).

In a nonredundant matrix \mathbf{A} , where $n > m$, there may be $\binom{n}{m}$ basic nonnegative solutions (i.e., vertices of the polyhedron). Therefore, as the number of vertices in the polyhedron is finite, the algorithm always terminates.

The simplex method consists of two phases:

Phase 1

Obtaining a starting feasible solution (from the set of basic feasible solutions), if one exists, or returning information that no solution is possible (if a region of feasible solutions is empty).

Phase 2

Execution of an iterative procedure to approach the optimal solution initiated from a starting feasible solution obtained in Phase 1. During this phase, consecutive iterations are to identify vertices with improved objective function values. This iterative phase either terminates by reaching the optimal vertex (by identifying that a further improvement of the objective function value is impossible) or recognizing an unbounded edge (implying that there is no solution to the problem).

In the simplex method, a linear program is represented by a *simplex tableau* of the following form:

$$\begin{bmatrix} \mathbf{A} & \mathbf{0} & \mathbf{b} \\ -\mathbf{c}^T & \mathbf{1} & \mathbf{0} \end{bmatrix}$$

Table 6.1 Initial simplex tableau for the problem specified by formula (6.44)

	x_1	x_2	x_3	x_4	x_5	$-z$	b
x_3	1	2	1	0	0	0	8
x_4	2	1	0	1	0	0	8
x_5	0	1	0	0	1	0	3
$-z$	-1	-1	0	0	0	1	0

Identification in Phase 1 of a starting feasible solution can be achieved via a transformation of matrix **A** so that its rearranged form includes:

- The identity matrix I of $m \times m$ size, columns of which refer to the *basic variables* of a solution
- Matrix **B** of $m \times (n - m)$ size associated with *nonbasic variables*

This rearranged form of **A** directly points to a given basic feasible solution when setting the values of nonbasic variables to 0. For instance, the problem given by formula (6.44) translates into the simplex tableau presented as Table 6.1.

This tableau shows that matrix **A** already includes the identity matrix I associated with variables x_3, x_4 and x_5 . Therefore, by setting the values of nonbasic variables (x_1, x_2 in this case) to 0, we obtain the starting feasible solution given by vertex $x^1 = (x_1, x_2, x_3, x_4, x_5) = (0, 0, 8, 8, 3)$, i.e., vertex 1 in Fig. 6.6 with the value of the objective function $z(x^1) = 0$.

In each consecutive iteration of Phase 2, the simplex algorithm visits a vertex that was not previously visited. Visiting the next vertex means identifying another solution for the problem for a different identity matrix I , i.e., formed by another set of m basic variables. To guarantee that the optimal solution is finally reached, the transition from the current vertex to the next one is done so that the value of the objective function for the next vertex is not lower than the respective one for the formerly visited vertex.

To assure this, when transforming the simplex tableau toward identifying another vertex for another identity matrix I , the following rules for the selection (for the purpose of a transition) of the main column and the main row in **A** must be met:

- (A) The main column r can be any column j pointing to a certain variable x_j , for which (a) the value of the coefficient referring to that variable j in the objective function row **z** is negative, and (b) this column j includes at least one positive coefficient $a_{i,j}$ in rows i referring to the basic variables.
- (B) The main row p can be any row i referring to the basic variables, for which coefficient $a_{i,r}$ in the selected main column r is positive, while the ratio $\frac{b_i}{a_{i,r}}$ (where $a_{i,r} \neq 0$) is minimal.

After that, the simplex tableau is transformed in the following way:

- Values of all elements $a_{p,j}$ of the main row p are updated as given in formula (6.45).

Table 6.2 The simplex tableau after applying the transformations in the first iteration of Step 2

	x_1	x_2	x_3	x_4	x_5	$-z$	b
x_3	0	$\frac{3}{2}$	1	$-\frac{1}{2}$	0	0	4
x_1	1	$\frac{1}{2}$	0	$\frac{1}{2}$	0	0	4
x_5	0	1	0	0	1	0	3
$-z$	0	$-\frac{1}{2}$	0	$\frac{1}{2}$	0	1	4

Table 6.3 The simplex tableau after applying the transformations in the second iteration of Step 2

	x_1	x_2	x_3	x_4	x_5	$-z$	b
x_2	0	1	$\frac{2}{3}$	$-\frac{1}{3}$	0	0	$\frac{8}{3}$
x_1	1	0	$-\frac{1}{3}$	$\frac{2}{3}$	0	0	$\frac{8}{3}$
x_5	0	0	$-\frac{2}{3}$	$\frac{1}{3}$	1	0	$\frac{1}{3}$
$-z$	0	0	$\frac{1}{3}$	$\frac{1}{3}$	0	1	$5\frac{1}{3}$

$$a'_{p,j} = \frac{a_{p,j}}{a_{p,r}} \quad (6.45)$$

- Values of all elements of other rows in the simplex tableau are updated by applying the transformation given by formula (6.46):

$$a'_{i,j} = a_{i,j} - a_{p,j} \frac{a_{i,r}}{a_{p,r}} \quad (6.46)$$

In our example, in iteration 1 of Phase 2, column j associated with the basic variable x_1 is selected as the main column, while the row associated with the basic variable x_4 becomes the main row. This will imply that x_4 will be replaced by x_1 in the new identity matrix \mathcal{I} . The simplex tableau gets transformed as presented in Table 6.2.

This updated simplex tableau points at a next vertex $x^2 = (x_1, x_2, x_3, x_4, x_5) = (4, 0, 4, 0, 3)$, i.e., vertex 5 in Fig. 6.6 with the value of the objective function $z(x^2) = -4$.

In iteration 2 of Phase 2, column j associated with the basic variable x_2 is selected as the main column, while the row associated with the basic variable x_3 becomes the main row. This will imply that x_3 will be replaced by x_2 in the new identity matrix \mathcal{I} . The simplex tableau gets further transformed as presented in Table 6.3.

This updated simplex tableau points at a next vertex $x^3 = (x_1, x_2, x_3, x_4, x_5) = (\frac{8}{3}, \frac{8}{3}, 0, 0, \frac{1}{3})$, i.e., vertex 4 in Fig. 6.6 with the value of the objective function $z(x^3) = -5\frac{1}{3}$.

Since all the coefficients in the last row of the recently updated simplex tableau are positive, rule (1) is not met, and, therefore, further selection of the main column is not possible. It means that the recently visited vertex x^3 indicates the best solution obtained for $x_1 = \frac{8}{3}$ and $x_2 = \frac{8}{3}$ characterized by the value of the objective function $z(x^3) = -5\frac{1}{3}$ (as also illustrated in Fig. 6.6).

6.8.2 Branch-and-Bound Method

The *branch-and-bound* (B&B) optimization method [19, 22, 33] is the most efficient general approach for solving mixed-integer linear optimization problems where certain variables are constrained to integer values, as in MIP problems. In fact, commercially available MIP solvers include the B&B method [22].

The name of this method reflects the two main operations performed recursively:

- Splitting (also called branching) of the current search space into disjoint subspaces
- Determination of the lower and upper bounds of the objective function in these subspaces

The operation of the B&B method can be best illustrated by means of a tree called B&B tree. The root of that tree denotes the entire solution space and is linked with other vertices at consecutive lower levels of the tree referring to the given branches (i.e., regions) of the search space.

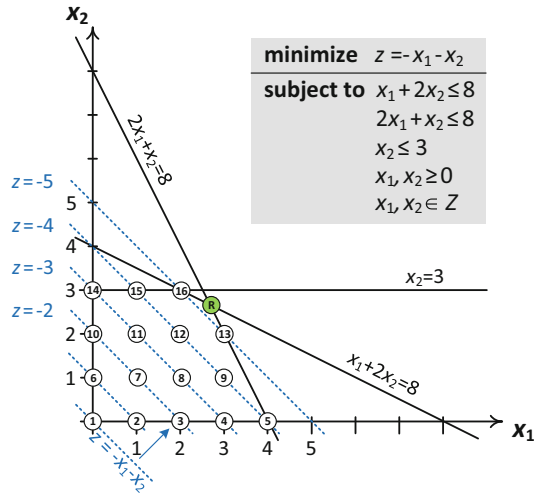
During the execution of the B&B method, vertices are visited in a best-first manner, meaning that before performing the branching operation for a given vertex, all the other vertices located at the same level in the tree are first sequentially visited. When visiting a particular vertex, B&B analyzes the lower and upper bounds of the objective function for the related branch to determine whether this branch can provide a better solution than the best one identified so far. If the answer is negative, the branch is discarded, and no further search within this branch is performed.

The efficiency of this scheme is in its bounding operation, resulting in discarding certain branches of the solution space as soon as it becomes clear that these branches do not contain solutions better than the best one identified so far. The related lower and upper bounds on the objective function determined in branches are thus used to prune the search space. Otherwise, if only branching (i.e., without bounding) was performed, the algorithm would verify all single feasible solutions, as in the brute-force approach.

Figure 6.7 presents the example illustration of a two-variable integer linear programming problem, being an updated version of the problem considered earlier in this chapter in Fig. 6.6, in a sense that here variables x_1 and x_2 are allowed to take integer values only. Therefore, the set of feasible solutions now consists of only sixteen vertices marked as (1)–(16) in Fig. 6.7 instead of the original polytope from Fig. 6.6. As can be seen in Fig. 6.7, there are 16 feasible points, and the optimal value of the objective function z equal to -5 is attributed to two vertices $x^{13} = (3, 2)$ and $x^{16} = (2, 3)$.

Contrary to a brute-force algorithm analyzing the value of all 16 feasible solutions to finally identify the optimal one, the set of feasible solutions analyzed in parallel by the B&B algorithm would be much smaller. For the optimization problem from Fig. 6.7, as illustrated in Fig. 6.8, we start the B&B method by relaxing (i.e., ignoring) the assumption of the integrality of values of variables x_1 and x_2 .

Fig. 6.7 The illustration to an example of a two-variable integer linear programming problem with the respective optimal value of -5 for the objective function z attributed to two vertices 13 and 16 characterized by coordinates $(3, 2)$ and $(2, 3)$, respectively



$$\begin{aligned} &\text{minimize } z = -x_1 - x_2 \\ &\text{subject to } x_1 + 2x_2 \leq 8 \\ &\quad 2x_1 + x_2 \leq 8 \\ &\quad x_2 \leq 3 \\ &\quad x_1, x_2 \geq 0 \\ &\quad x_1, x_2 \in Z \end{aligned}$$

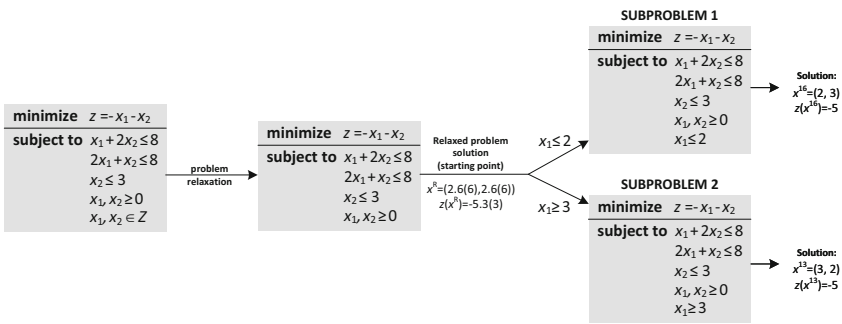


Fig. 6.8 Example illustration of the branch-and-bound algorithm execution for the optimization problem from Fig. 6.7

The solution to this relaxed problem is $z = -5\frac{1}{3}$ at vertex $x^R = (2\frac{2}{3}, 2\frac{2}{3})$ marked in green in Fig. 6.7. Vertex x^R becomes the starting point for the B&B method. Among variables determining the location of the starting point, we commonly take the variable with the largest decimal part. Since in the case of our vertex x^R , both variables x_1 and x_2 have equal decimal parts, we arbitrarily select x_1 and start investigating other possible integer values of x_1 , i.e., either $x_1 \leq 2$ or $x_1 \geq 3$. By doing so, we formulate two new subproblems where, in each case, the original set of the relaxed problem formulas is extended by either $x_1 \leq 2$ or $x_1 \geq 3$. In the case of the first subproblem, its optimal solution with $z = -5$ is achieved for vertex $x^{16} = (2, 3)$. Since both variables of x^{16} have integer values, we can terminate the calculations for this branch and consider $z = -5$ as the current upper bound on the value of the objective function. Otherwise, if not all values of variables were integer,

we would continue branching until identifying a solution to a subproblem compliant with this requirement.

The optimal solution for the second subproblem is provided at vertex $x^{13} = (3, 2)$, with both variables taking integer values. The value of z for this solution is equal to -5 , which, in turn, is the same as the current lower bound on the objective function value. The algorithm terminates here and returns vertices x^{16} and x^{13} as alternate optimal solutions. Also, it is worth noting that to identify these optimal solutions, we did not have to verify any other solution from the set of 16 feasible solutions from Fig. 6.7.

There are several variants of the B&B method depending on the strategy of searching the tree and selecting the variable for branching (see, e.g., branch-and-price or branch-and-cut methods in [22]).

6.8.3 Column Generation Method

Column generation is a technique useful in solving linear programming problems characterized by many variables. In this method, the solution providing the global minimal value of the objective function is approached in subsequent iterations, each operating on larger and larger subsets of variables. The name of this method follows from the fact that each decision variable is a column in the programming tableau. Therefore, expanding the subset of the considered problem variables in subsequent iterations actually widens the \mathbf{A} matrix by adding subsequent columns of coefficients indicating the newly added variables.

The justification for using column generation comes from the observation that for many problems, the optimal solution can be obtained by considering only a small subset of all problem variables. Indeed, as we discussed in Sect. 6.8.1, the optimal solution is determined by the values of the basic variables, while the other (i.e., nonbasic) variables are set to 0. The number of basic variables is given by $\text{rank}(\mathbf{A})$, which, in fact, can often be much smaller than the number of all problem variables n (e.g., n can be exponential in the problem size (graph size)).

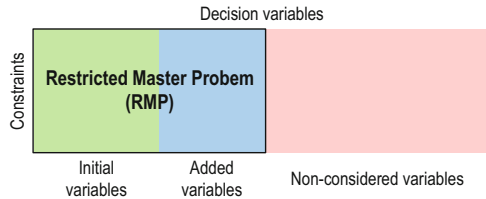
In the column generation algorithm, two problems are considered:

- The *master problem* is the original problem, however, with the set of problem variables limited to a subset of a certain size considered in a given iteration.
- The *column generation subproblem* (also referred to as the *pricing problem*) used to find an “improving variable,” i.e., a variable that, if added to the set of considered variables, improves the value of the objective function in the master problem.

Both problems are solved in the following five-step procedure:

1. Initialize the master problem by determining the initial set of variables.
2. Solve the restricted master problem.
3. Solve the column generation subproblem to find an improving variable.

Fig. 6.9 Illustration of subsets of variables taken into consideration when solving the RMP in column generation method



4. If an improving variable exists, extend the set of variables and go to step 2.
5. If identification of an improving variable was not possible, stop and return the current solution to the master problem as the optimal one.

In the column generation method, as illustrated in Fig. 6.9, we start with an initial number of variables guaranteeing to obtain a feasible solution to the *restricted master problem (RMP)*, i.e., a problem with a confined number of variables. The solution to the column generation subproblem is variable x_j such that if added to the master problem, it results in a decrease in the objective function value in the master problem. If such a variable x_j exists, it is returned as a solution to the column generation subproblem, which extends the subset of added variables of the restricted master problem considered in the next iteration. Otherwise, this means that the recently identified solution to the master problem is returned as the optimal one. Variables not considered during the execution of the column generation method are marked as “non-considered” in Fig. 6.9.

The adaptation of the column generation scheme to solve problems of network routing can be found, e.g., in [22].

6.9 Conclusions

This chapter discussed optimization methods for the problem of resilient routing in capacity-constrained connection-oriented communication systems. In its initial part, we highlighted the main characteristics of network flows and provided the description of the network model used later for modeling the resilient routing problems. In the core part of this chapter, we discussed the formulations of linear programming optimization problems to determine resilient routing through node-/link-disjoint communication paths and commented on the optimization issues of shared protection schemes with the related linear programming models. The final part of this chapter was to explain the operation of the major schemes of mathematical programming that can be useful in solving the discussed optimization problems.

Formulating optimization problems and configuring the related mathematical programming procedures is undoubtedly an art. Advanced experience is indeed needed to fine-tune the properties of the considered programming schemes to match the needs for acceptable computational time values and solution quality.

? Questions

1. Provide the classification of mathematical programming optimization models and characterize each model type shortly.
 2. Characterize the properties of multi- and single-commodity flows and comment on their relation with the shortest path routing.
 3. Compare the properties of node-link and link-path formulations of optimization models for the problem of resilient routing.
 4. Characterize the main features of linear programming models for problems of resilient routing in scenarios of failures of nodes and links.
 5. Describe the conditions necessary to enable backup path sharing.
 6. Explain the pros and cons of the conventional “a priori” scheme for sharing the backup paths in optical transport networks.
 7. Discuss the challenges to fast recovery of affected services under the backup path sharing and a possible strategy for limiting the recovery time in such scenarios.
 8. Explain the concept of p -cycles and compare its properties with dedicated and shared protection schemes.
 9. Describe the main features and phases of the simplex method.
 10. Explain the main properties of the branch-and-bound optimization method.
 11. Characterize the properties of the column generation optimization scheme.
-

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice Hall, New York (1993)
2. Bhandari, R.: Optimal physical diversity algorithms and survivable networks. In: Proceedings of the 2nd IEEE Symposium on Computers and Communications (ISCC'97), pp. 433–441 (1997)
3. Bhandari, R.: Survivable Networks: Algorithms for Diverse Routing. Kluwer Academic Publishers, Dordrecht (1999)
4. Chatterjee, B., Oki, E.: Elastic Optical Networks: Fundamentals, Design, Control, and Management. CRC Press, Boca Raton (2022)
5. Dantzig, G.B.: Linear Programming and Extensions. Princeton University Press, Princeton (1963)
6. Doucette, J., Giese, P., Grover, W.D.: Combined node and span protection strategies with node-encircling p -cycles. In: Proceedings of the 5th International Workshop on Design of Reliable Communication Networks (DRCN'05), pp. 213–221 (2005)
7. Ford, L.K., Fulkerson, D.K.: Flows in Networks. Princeton University Press, Princeton (1962)
8. Grover, W.D., Shen, G.: Extending the p -cycle concept to path-segment protection. In: Proceedings of the IEEE International Conference on Communications (IEEE ICC'03), vol. 2, pp. 1314–1319 (2003)
9. Grover, W.D., Stamatelakis, D.: Cycle-oriented distributed preconfiguration: Ring-like speed with mesh-like capacity for self-planning network restoration. In: Proceedings of the 1998 IEEE International Conference on Communications (ICC'98), pp. 537–543 (1998)

10. Hansen, P.: Graph Coloring and Applications. American Mathematical Society (1999)
11. Ho, P.-H.: State-of-the-art progress in developing survivable routing schemes in mesh WDM networks. *IEEE Commun. Surv. Tutorials* **6**(4), 2–16 (2004)
12. Ho, P.-H., Moutah, H.T.: Shared protection in mesh WDM networks. *IEEE Commun. Mag.* **42**(1), 70–76 (2004)
13. Jaumard, B., Rocha, C., Baloukov, D., Grover, W.D.: A column generation approach for design of networks using path-protecting p -cycles. In: *Proceedings of the 6th International Workshop on Design of Reliable Communication Networks (DRCN'07)*, pp. 1–8 (2007)
14. Kodian, A., Grover, W.D.: Failure-independent path-protecting p -cycles: efficient and simple fully preconnected optical-path protection. *IEEE/OSA J. Lightwave Technol.* **23**(10), 3241–3259 (2005)
15. Lasdon, L.S.: *Optimization Theory for Large Systems*. Dover Publications, New York (2011)
16. Minoux, M.: *Mathematical Programming: Theory and Applications*. John Wiley & Sons, London (1986)
17. Molisz, W., Rak, J.: Region protection/restoration scheme in survivable networks. *Lecture Notes in Computer Science*, vol. 3685, pp. 442–447. Springer, Berlin (2005)
18. Mukherjee, B.: *Optical WDM Networks*. Springer, Berlin (2006)
19. Murthy, K.: *Linear and Combinatorial Programming*. John Wiley & Sons, London (1976)
20. Nemhauser, G.L., Wolsey, L.A.: *Integer and Combinatorial Optimization*. Wiley, London (1999)
21. Papadimitriou, Ch.: *Computational Complexity*. Addison-Wesley, Reading, (1994)
22. Pióro, M., Medhi, D.: *Routing, Flow and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann, Los Altos (2004)
23. Rak, J.: Fast service recovery under shared protection in WDM networks. *IEEE/OSA J. Lightwave Technol.* **30**(1), 84–95 (2012)
24. Rak, J., Hutchison, D., Tapolcai, J., Bruzgiene, R., Tornatore, M., Mas-Machuca, C., Furdek, M., Smith, P.: Fundamentals of communication networks resilience to disasters and massive disruptions. In: *Guide to Disaster-Resilient Communication Networks*, pp. 1–43. Springer, Berlin (2020)
25. Ramamurthy, B., Sahasrabudhe, L., Mukherjee, B.: Survivable WDM mesh networks. *IEEE/OSA J. Lightwave Technol.* **21**(4), 870–883 (2003)
26. Ramaswami, R., Siwarajan, K.N., Sasaki, G.H.: *Optical networks: a practical perspective*. Morgan Kaufmann, Los Altos (2009)
27. Rennie, R., Law, J. (eds): *A Dictionary of Physics*, 8th edn. Oxford University Press, Oxford (2019)
28. Sack, A., Grover, W.D.: Hamiltonian p -cycles for fiber-level protection in semi-homogeneous, homogeneous, and optical networks. *IEEE Netw.* **18**(2), 49–56 (2004)
29. Schupke, D.A., Gruber, C.G., Autenrieth, A.: Optimal configuration of p -cycles in WDM networks. In: *Proceedings of the 2002 IEEE International Conference on Communications (IEEE ICC'02)*, pp. 2761–2765 (2002)
30. Suurballe, J.W.: Disjoint paths in a network. *Networks* **4**(2), 125–145 (1974)
31. Suurballe, J.W., Tarjan, R.E.: A quick method for finding shortest pairs of disjoint paths. *Networks* **14**(2), 325–336 (1984)
32. Vizzaino, J.L., Ye, Y., Lopez, V., Jimenez, F., Musumeci, F., Tornatore, M., Pattavina, A., Krummrich, P.M.: Protection in optical transport networks with fixed and flexible grid: cost and energy efficiency evaluation. *Opt. Switching Netw.* **11**(A), 55–71 (2014)
33. Wolsey, L.A.: *Integer Programming*. John Wiley & Sons, London (2020)

Chapter 7

Efficient Methods to Determine Disjoint Paths for Single Demands



In this chapter, we concentrate on computationally efficient methods for determining the shortest sets of k end-to-end disjoint communication paths crucial in assuring resilient routing. As already discussed in this book, the ability of a networked system to maintain delivery of its services to end users despite the occurrence of failures is much dependent on the existence of alternate paths (called *backup paths*) able to substitute the respective *primary paths* (also referred to as *working paths*) when affected by failures. In scenarios of failures of system nodes/links, those backup paths to remain operational should be node-/link-disjoint with the related working paths.

Since, as discussed in [9], scenarios of single failures (i.e., referring to failures of a single network element at a time) are most probable (involving well over 50% of all failure scenarios), deployment of schemes involving installation of a single backup path being node-/link-disjoint with the related working path often turns out to be sufficient to assure resilient communications in the case of the majority of failure events. However, as the frequency, intensity, and scale of disaster-induced massive failure events such as those triggered by natural disasters or malicious human activities are increasing, cases of simultaneous failures of multiple elements of a networked system are becoming more and more severe, which, in turn, calls for the deployment of relevant schemes able to maintain delivery of services under such circumstances.

In particular, as most natural disasters (e.g., hurricanes, fires, volcano eruptions, and floods) manifest their activity in certain geographical regions, it is often sufficient to assure the *regional disjointness* of the related working and backup paths. This property illustrated in Fig. 7.1a means that a single backup path being *geographically diverse* (i.e., not traversing the same geographical region) with the related working path is sufficient to ensure resilient routing in scenarios of failures of multiple elements of a network occurring in the considered geographical region.

However, in scenarios of malicious human activities, the resulting failures of multiple elements of a networked system are rarely limited to specific geographical

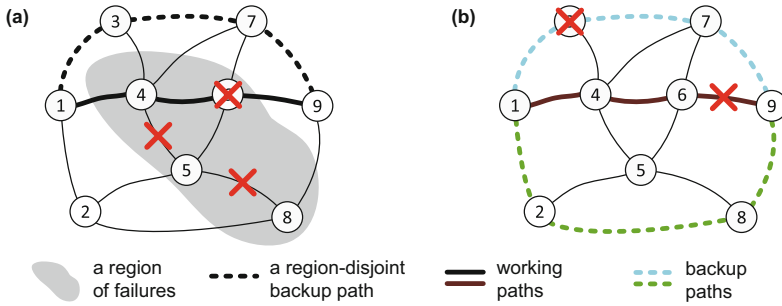


Fig. 7.1 Example schemes of resilient routing for scenarios of multiple failures: confined to a given region (a) and without a regional correlation (b)

regions. To be able to assure 100% of restorability of services in such scenarios of simultaneous uncorrelated failures of multiple $(k-1)$ network elements, schemes of shortest sets of k end-to-end mutually disjoint paths are necessary. For instance, in the example scenario of simultaneous failures of any two arbitrarily chosen network elements illustrated in Fig. 7.1b, a scheme of three disjoint paths between end nodes 1 and 9 is needed, as these two failures can affect at most two paths for this demand. In general, the possibility of determining a set of k end-to-end mutually disjoint paths depends on the minimum degree d_{\min} of a network graph, which should be at least equal to k .

As discussed in [9], schemes for the calculation of sets of disjoint paths for demands d_r from D can be categorized into the following three classes:

- *Min-sum* approaches, where the objective is to find for each demand d_r a shortest set of k end-to-end mutually node-/link-disjoint paths, i.e., the set of disjoint paths for which the sum of costs of these paths is minimal.
- *Min-min* approaches, where the objective is to find for each demand d_r the working path of the minimal cost together with a set of $k-1$ end-to-end node-/link-disjoint paths.
- *Min-max* approaches, where the objective is to find for each demand d_r a set of k end-to-end mutually node-/link-disjoint paths such that the cost of the most expensive path is minimized.

Among these three classes of problems, in networks with constrained link resources, only the min-sum variant can be solved for a given demand d_r in polynomial time [9]. For instance, for a single demand d_r , the problem of finding the pair of working and protection paths of the minimal value of the sum of their cost can be achieved in polynomial time by Surballe's algorithm [20, 21] or its modification—Bhandari's approach [2, 3]. In each of them, a single end-to-end path can be found using Dijkstra's algorithm [5]. These two algorithms are also suitable to calculate the shortest set of k end-to-end node-/link-disjoint paths with the lowest overall cost, necessary if protection against multiple failures (i.e., a simultaneous failure of multiple network elements) is required [17]. Such a scenario can occur,

for instance, if several network links are placed together in a duct that is cut by a third party.

However, it must be noted that the observation made above is valid only for a single demand. As shown in [18], the considered problem of resilient routing becomes \mathcal{NP} -hard if the sets of k disjoint paths for more than one demand are needed to be determined simultaneously. Indeed, concerning the common case of networks with limited resources, the min-sum problem addressed for all demands simultaneously is certainly combinatorial since assigning link capacity to paths of a given demand d_i may impact the availability of capacity at these links for paths of the other demands. The related formulations of optimization problems are discussed in detail in Chap. 6 of this book.

For the other classes of min-min and min-max problems, no known algorithms would return the optimal solutions in polynomial time, even for single demands. Therefore, in all such cases, only using suboptimal heuristic schemes is reasonable.

In the literature, the term *multi-cost network* is used to represent cases with differentiated costs assigned to network links in computations of multiple end-to-end disjoint paths of the same demand (e.g., as in the case of a backup path sharing scheme using differentiated costs ξ_h and ζ_h of arcs a_h for working and backup paths, respectively). On the contrary, in the scheme of a *single-cost network*, the same cost ξ_h of arc a_h is assigned to network links in computations of all disjoint paths of a given demand (e.g., as in [16]).

The problem of finding a set of multiple end-to-end disjoint paths in a multi-cost network was shown to be \mathcal{NP} -hard even for a single demand [22]. Therefore, to obtain reasonable solutions to such problems in a time-efficient way, there could be a need for using specialized heuristic approaches, e.g., as proposed in [15].

In this chapter, our analysis is focused on efficient schemes to determine shortest sets of k end-to-end disjoint paths. However, contrary to Chap. 6 of this book presenting the optimization schemes applicable for the related \mathcal{NP} -hard problems of determining the communication paths for all demands simultaneously, here we investigate the representative methods for solving problems of resilient routing addressing each demand separately. As our analysis in the remaining part of this chapter is limited to the min-sum class of problems, the related schemes discussed here for single demands in single-cost networks will lead to optimal solutions. In contrast, the approach discussed later in this chapter for multi-cost networks is able to deliver only suboptimal results.

In this chapter, we consider networks represented by directed weighted graphs $G(V, E)$, where V is the set of vertices representing network nodes, while E is a set of edges e representing bidirectional network links between certain pairs of network nodes i and j . Bidirectional links can also be well represented by set A of directed arcs $a_h = (i, j)$ in graph G (instead of set E), where each edge e is replaced by two oppositely directed arcs $a_h = (i, j)$ and $a'_h = (j, i)$.

It is worth noting that the representation of network links by directed arcs remains the only possibility for networks with different uplink/downlink nominal capacities or in the case of one-way transmission. Wherever applicable, in the remaining part

of this chapter, in path calculations, the costs of allocation of unitary capacity for paths at arcs a_h are given by ξ_h . Each arc is assigned a nominal capacity c_h referring to the total flow possible via that arc.

The remaining part of this chapter is structured as follows. Section 7.1 explains the details of the common Dijkstra's algorithm for the calculation of the shortest path between a certain pair of end nodes of demand d_r . Although Dijkstra's algorithm itself is not devoted to providing the set of k end-to-end disjoint paths for a given demand, it is explained here because of its essential role in resilient routing algorithms discussed later in this chapter. Sections 7.2 and 7.3 explain and illustrate the most representative schemes to determine a shortest set of disjoint paths in single-cost networks, namely Suurballe's and Bhandari's algorithms. Section 7.4 discusses the properties of the k -Penalty algorithm designed to determine the set of k end-to-end disjoint paths in multi-cost networks. Section 7.5 concludes the chapter.

7.1 Dijkstra's Algorithm

The objective of the algorithm published by Edsger W. Dijkstra in 1959 [5], commonly called *Dijkstra's algorithm*, is to determine a directed path of the smallest overall cost (i.e., the total cost of all traversed arcs) between a given pair of source and destination vertices v_s and v_t in graph $G = (V, A)$ with nonnegative costs of arcs. In particular, if the cost of each arc reflects the arc length, this algorithm provides a solution to the related *shortest path problem (SPP)*. Although this algorithm was originally proposed for directed networks, it can also be used for networks with undirected links if each such link is represented by a pair of oppositely directed arcs $a_h = (i, j)$ and $a'_h = (j, i)$ or by a single edge e .

Dijkstra's algorithm is available in several versions. Apart from its original version aimed at determining the shortest path between a given pair of vertices, one of its major variants is to find the set of shortest paths from a given source vertex v_s to any other vertex in the network graph, forming the *shortest path tree (SPT)* [4].

The area of applications of the algorithm is undoubtedly wide: from solving various transportation and location problems to operation of routing protocols in communication networks. In the latter case, the algorithm is often used, e.g., in OSPF (Open Shortest Path First) [13] or IS-IS (Intermediate System to Intermediate System) [14] routing protocols.

The Main Properties of the Algorithm

The essence of Dijkstra's algorithm operation is in the use of labels l_i representing the current cost of the best path found so far from vertex v_s to vertices v_i , and in updating these labels each time a lower-cost path is identified [4, 9]. Therefore, to find the shortest path between a given pair of source and destination vertices v_s and v_t in a given network graph, Dijkstra's algorithm iteratively produces partial solutions determining the related shortest paths from the source vertex v_s to a subset of intermediate vertices v_i until the cost of the shortest path from v_s to v_t is

identified. However, instead of simply progressing from the source vertex v_s toward the destination vertex v_t , in each iteration, the algorithm visits one vertex v_i from the set of non-visited vertices of graph G characterized by the current lowest cost (i.e., the shortest distance) of the related path to that vertex v_i from the source vertex v_s traversing only the already visited vertices.

The operation of Dijkstra's algorithm makes use of the following features:

- Division of the set of all vertices of G into two disjoint subsets of *non-visited vertices*—vertices with the unknown final costs of the related shortest paths from the source vertex v_s and *visited vertices*, for which the shortest paths from v_s have already been identified.
- The label l_i of each vertex v_i representing the lowest total cost of the shortest path determined so far between source vertex v_s and the considered vertex v_i (i.e., a *tentative distance* of v_i from v_s).
- The *predecessor index* p_i being the index of a vertex directly preceding vertex v_i on a path from v_s .
- The *current vertex* referring to a particular vertex v_i from the set of non-visited vertices characterized by the lowest label value l_i among all non-visited vertices. Its tentative distance (cost) from the source vertex v_s can be considered to be the final one (i.e., the lowest possible). This is also the vertex at which the labels reflecting the distances between vertex v_s and all its neighboring unvisited vertices are recalculated in the next iteration.

The steps of Dijkstra's algorithm to find the shortest path between v_s and v_t are as follows:

1. Mark all vertices of G as belonging to the set of non-visited vertices. Assign the value of 0 to the label l_s (i.e., a tentative distance) of source vertex v_s and infinity to labels l_i of all other vertices v_i in G . For each vertex v_i , set its predecessor index p_i to NULL. Set the initial vertex v_s as the current vertex.
2. For the current vertex, recalculate the tentative distances to all its neighboring unvisited vertices via paths traversing the current vertex. If, for some nodes v_i , these new distances occur to be smaller than the related values of labels l_i , update these labels with the values of the related recalculated distances.
3. Move the current vertex to the set of visited vertices.
4. If destination vertex v_t is marked as visited, terminate and return the path consisting of visited vertices between v_s and v_t as the shortest path.
5. If destination vertex v_t belongs to the set of non-visited vertices, select a new current vertex from the set of non-visited vertices characterized by the lowest label value l_i , and go to Step 2.

Example

Consider an example task to determine the shortest path between source vertex $v_s = 2$ and destination vertex $v_t = 7$ in a network graph from Fig. 7.2a, where costs of each arc are marked in blue. Each vertex v_i in Fig. 7.2a is provided with a pair (l_i, p_i) marked in red, referring to its label l_i and the predecessor index p_i .

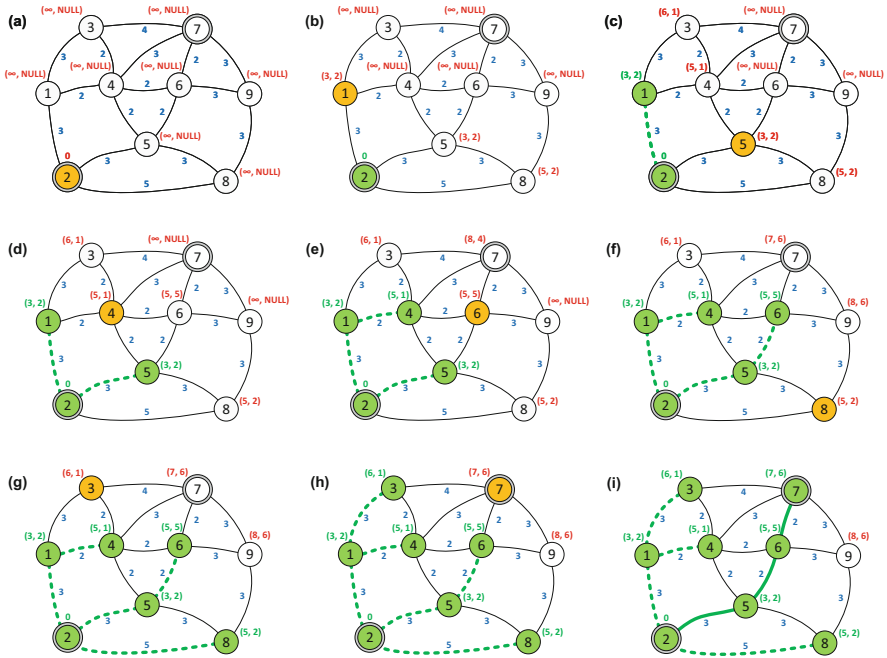


Fig. 7.2 Execution steps of Dijkstra’s algorithm when determining the shortest path between vertices 2 and 7 in the example network topology graph

As shown in Fig. 7.2a, during the execution of the first iteration of Dijkstra’s algorithm, in Step 1 labels l_i of all vertices are assigned the respective tentative costs (equal to 0 for the source vertex v_s and to ∞ for all the other vertices), all vertices are classified as non-visited, and source vertex $v_s = 2$ is set as the current vertex (marked in orange). As illustrated in Fig. 7.2b, in Step 2, labels l_i (tentative distances) for all unvisited neighboring vertices of the current vertex 2 are recalculated, the related predecessor indices p_i are set to 2 wherever labels l_i were updated (since vertex 2 is the current vertex), and the current vertex (i.e., v_s) is moved in Step 3 to the set of visited vertices (marked in green). Vertex 1 is selected in Step 5 as a new current vertex (marked in orange).

During the second iteration of the algorithm, as illustrated in Fig. 7.2c, labels l_i and predecessor indices p_i for all non-visited neighbors (i.e., vertices 3 and 4) of the current vertex 1 are determined, vertex 1 is moved to the set of visited vertices (marked in green), its distance from source vertex v_s is considered lowest possible (and, therefore, final), and vertex 5 of the lowest tentative distance (label $l_5 = 3$) is selected as the current vertex.

During the third iteration of the algorithm, as provided in Fig. 7.2d, the label and predecessor index for the non-visited neighboring vertex 6 of the current vertex 5 are determined, the current vertex 5 is moved to the set of visited vertices, its

distance from source vertex v_s is considered lowest possible (and, therefore, final), and vertex 4 of the lowest tentative distance ($l_4 = 5$) is selected as the current vertex.

As illustrated in Fig. 7.2e, the fourth iteration results in updating the tentative distance only for vertex 7 (setting $l_7 = 8$, $p_7 = 4$), moving the current vertex 4 to the set of visited vertices, and selecting vertex 6 as a new current vertex.

As illustrated in Fig. 7.2f, the fifth iteration results in updating the labels and predecessor indices for vertices 7 and 9, moving the current vertex 6 to the set of visited vertices, and selecting vertex 8 as a new current vertex.

In the sixth iteration, no updates of tentative costs are applied, vertex 8 is moved to the set of visited vertices, and vertex 3 is selected as a new current vertex—see Fig. 7.2g.

In the seventh iteration, no updates of tentative costs are applied, vertex 3 is moved to the set of visited vertices, and vertex 7 is selected as a new current vertex—see Fig. 7.2h.

The assignment of the role of the current vertex to the destination vertex finalizes the execution of the algorithm (following the execution of Step 4 in the eighth iteration). The path (2, 5, 6, 7) shown in Fig. 7.2i is returned as the shortest one (of the total minimal cost of 7).

It is worth noting that if we allow Dijkstra's algorithm to continue its operation until all vertices in G are marked as visited, the algorithm will produce the shortest path tree rooted at vertex v_s .

Proof of the Correctness of Dijkstra's Algorithm

Proving the correctness of Dijkstra's algorithm is based on the following two invariants that apply throughout the algorithm's execution.

- For each visited vertex v_i , the value of l_i determines the cost of the shortest path from v_s to v_i .
- For each non-visited vertex v_i , the value of l_i determines the cost of the shortest path from v_s to v_i through visited vertices only.

Showing the correctness of these invariants is trivial for the initial case of only one visited vertex (i.e., the source vertex v_s). For any size of graph G , the formal proof of the correctness of Dijkstra's algorithm can be provided through induction on the number of vertices already marked as visited. For this purpose, it is sufficient to show that the assumption that the two invariants given above are true for any number k of visited vertices also implies that these invariants are also true when the number of visited vertices grows to $k + 1$.

This, in turn, can be demonstrated by a proof by contradiction. Let us assume that, for a given unvisited vertex v_i chosen to become the next visited vertex, there exists path \mathcal{P}' from v_s to v_i characterized by cost lower than the one identified so far. Then two cases are possible:

1. Path \mathcal{P}' also traverses another non-visited vertex v_j , as illustrated in Fig. 7.3a.
2. Path \mathcal{P}' does not traverse any other non-visited vertex v_j shown in Fig. 7.3b.



Fig. 7.3 The two cases for path \mathcal{P}' considered in the proof of Dijkstra's algorithm correctness

In the first case, following the two invariants given above, the lengths of the shortest paths from vertex v_s to vertices v_i and v_j traversing only visited vertices are given by l_i and l_j , respectively. In this case, we also have $l_i > l_j$, since the cost of a path from v_s to v_i is the sum of two costs: l_j (positive) and the cost ξ_h of arc a_h from vertex v_i to vertex v_j (also positive). However, the algorithm visited vertex v_i earlier than v_j (implying that the inverse relationship $l_i \leq l_j$ is, in fact, true), which contradicts the correctness of the first case.

In the second case, v_j is already a visited vertex. If it is the last one on the shortest path (see Fig. 7.3b), then $l_j + \xi_h \leq l_i$. However, the algorithm visited vertex v_i earlier than v_j , which also contradicts the correctness of the second case. ■

Computational Complexity

Computational complexity of Dijkstra's algorithm depends on the problem size (defined by the number of vertices $|V|$ and arcs $|A|$ in graph G) and methods used to access certain elements in the related data structures. In particular, if the set of vertices V is stored in a list while the respective arcs from A are kept in an adjacency matrix, it can be shown that the time of Dijkstra's algorithm execution is polynomial and bounded by $\Theta(|V|^2)$. This is implied by at most $|V|$ iterations of the algorithm needed to mark vertex v_i as a visited one, as well as the verification of path costs for at most $|V|-1$ vertices of the current vertex v_i . However, this quadratic complexity can be further lowered by the application of efficient methods to access elements of data structures such as, e.g., based on the implementation of Fibonacci heap [7] limiting the complexity to $\Theta(|A| + |V|\log|V|)$.

To conclude, polynomial complexity of Dijkstra's algorithm makes it efficient to produce the end-to-end path of the optimal (i.e., smallest possible) cost (see the proof of the algorithm's correctness). In particular, its at most quadratic computational complexity implies that this algorithm can provide a fast calculation of multi-hop paths for real-world networks.

It is worth noting that apart from Dijkstra's algorithm, other solutions are also available in the literature, such as the Floyd-Warshall algorithm, which finds the shortest paths between all pairs of nodes in a weighted graph [6].

7.2 Suurballe's Algorithm

The algorithm discussed in this section, called *Suurballe's algorithm*, was proposed in [20] to find a shortest pair of link-disjoint paths (i.e., paths having no common

links) between a given pair of source and destination vertices v_s and v_t in a weighted graph $G = (V, A)$, where V is a set of graph vertices, while A is a set of arcs $a_h = (i, j)$, each arc characterized by a nonnegative cost ξ_h . This algorithm aims to determine a pair of paths characterized by the optimal (i.e., minimal) value of the total cost of arcs of both paths, thus being the optimal solution to the min-sum problem.

The major features of Suurballe's algorithm are as follows:

- The use of Dijkstra's algorithm to find a shortest path between v_s and v_t .
- Transformation of graph G into G' in a way to preserve the nonnegativity of the updated costs of arcs, enabling further use of conventional Dijkstra's algorithm again to determine the second path.
- Determination of the final pair of disjoint paths by removing the overlapping edges.

As discussed in [9, 21], the following five stages can be distinguished in the execution of Suurballe's algorithm.

1. Determination of the minimum-cost path \mathcal{P}_1 from source vertex v_s to destination vertex v_t .
2. Transformation of graph G into G' , where new costs ξ'_h of arcs a_h are defined according to formula (7.1).
3. Removal from G' of arcs belonging to path \mathcal{P}_1 directed toward destination vertex v_t .
4. Determination of the minimum-cost path \mathcal{P}_2 from source vertex v_s to destination vertex v_t in the new graph G' .
5. Retrieval of a pair of disjoint paths \mathcal{P}'_1 and \mathcal{P}'_2 through removal from paths \mathcal{P}_1 and \mathcal{P}_2 of common arcs in G' .

$$\xi'_h = \xi_h + \bar{\xi}_{s(h)} - \bar{\xi}_{t(h)} \quad (7.1)$$

where $\bar{\xi}_{s(h)}$ and $\bar{\xi}_{t(h)}$ are costs of the shortest paths (i.e., total distances, if arc costs refer to their lengths) from the source vertex v_s to the source node of arc a_h and from the source vertex v_s to the destination node of arc a_h , respectively.

In general, following formula (7.1), costs ξ'_h of arcs belonging to the shortest path tree rooted at v_s in G are equal to 0. Costs ξ'_h of other arcs (not belonging to the shortest path tree in G) are greater than or equal to zero.

Example

For a demand to determine a pair of link-disjoint paths between vertices $v_s = 1$ and $v_t = 6$ in a graph from Fig. 7.4, in Step 1 of Suurballe's algorithm, the minimum-cost path \mathcal{P}_1 between a pair of end vertices of a demand is found using common Dijkstra's algorithm described earlier in this chapter. In this case, Step 1 results in calculating path $\mathcal{P}_1 = (1, 2, 3, 4, 5, 6)$ marked in green in Fig. 7.4a.

The outcome of the transformation in Step 2 of the original network graph G into graph G' is shown in Fig. 7.4b. This transition includes updating the arc costs

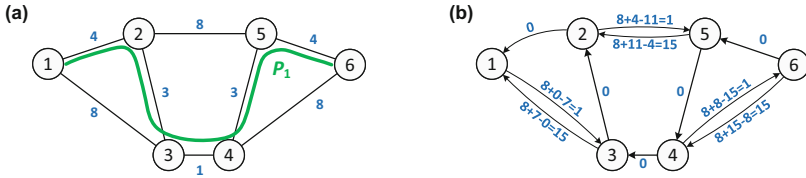


Fig. 7.4 Results of the execution of the first three steps of Suurballe’s algorithm for the example demand to establish a shortest pair of link-disjoint paths between nodes 1 and 6: path \mathcal{P}_1 (marked in green) calculated in Step 1 (a); the structure of graph G' with the related costs of arcs after the execution of Steps 2 and 3 (b)

according to formula (7.1) designed to avoid negative costs and removal of arcs traversed by path \mathcal{P}_1 from v_s to v_t .

In graph G' illustrated in Fig. 7.4b, costs ξ'_h of arcs are nonnegative. They are either equal to 0 (as, e.g., in the case of arcs traversed by path \mathcal{P}_1) or positive. This property, in turn, enables in Step 4 the application of a common variant of Dijkstra’s algorithm (without any modification) also for calculating path \mathcal{P}_2 in graph G' . Path \mathcal{P}_2 marked in orange in Fig. 7.5a has two interlacing arcs with path \mathcal{P}_1 , namely the arc (3, 2) and the arc (5, 4). Therefore, these arcs are excluded in Step 5 during the retrieval of the final pair of disjoint paths \mathcal{P}'_1 and \mathcal{P}'_2 characterized by the total minimal cost of 33 (based on costs ξ_h in graph G) illustrated in Fig. 7.5b.

However, while looking at Fig. 7.5, it may seem that Suurballe’s algorithm can return a pair of node-disjoint paths (which is a property stronger than link-disjointness since any node-disjoint pair of paths is also link-disjoint, but not vice versa), this observation is simply topology-specific. In general, Suurballe’s algorithm can provide at least link disjointness of the returned pair of paths (e.g., a pair of paths returned for graph G in Fig. 7.6).

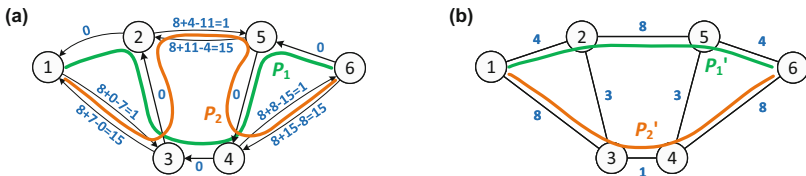
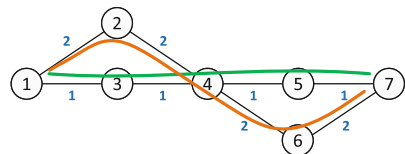


Fig. 7.5 Results of the execution of the last two steps of Suurballe’s algorithm for the example demand to establish a shortest pair of disjoint paths between nodes 1 and 6: path \mathcal{P}_2 (marked in orange) calculated in Step 4 based on arc costs ξ'_h in graph G' (a); the final shortest pair of disjoint paths \mathcal{P}'_1 and \mathcal{P}'_2 obtained in Step 5 from paths \mathcal{P}_1 and \mathcal{P}_2 after removing the interlacing edges (b)

Fig. 7.6 Example pair of link-disjoint paths returned by Suurballe’s algorithm for a pair of vertices 1 and 7



The Correctness of Suurballe's Algorithm

As discussed in detail in [3], to prove the correctness of Suurballe's algorithm establishing a pair of link-disjoint paths in graph G' , it is sufficient to show that the pair of paths of the lowest total cost in graph G' is also the shortest pair of paths in graph G . To provide a brief explanation of this property, let us first analyze the cost of any arbitrary single path \mathcal{P} between vertices v_s and v_t in graph G' which traverses transit vertices v_1, v_2, \dots, v_n . The total cost $\bar{\xi}'_t$ of such a path \mathcal{P} in a graph G' can be defined by formula (7.2).

$$\bar{\xi}'_t = \sum_{a_h \in \mathcal{P}} \xi'_h = (\xi_{(s,1)} + \bar{\xi}_s - \bar{\xi}_1) + (\xi_{(1,2)} + \bar{\xi}_1 - \bar{\xi}_2) + \dots + (\xi_{(n,t)} + \bar{\xi}_n - \bar{\xi}_t) \quad (7.2)$$

where $\xi_{(i,j)}$ is another form for expressing the cost ξ_h of arc $a_h=(i,j)$.

Formula (7.2) can be transformed into the following final form:

$$\bar{\xi}'_t = \sum_{a_h \in \mathcal{P}} \xi_h - \bar{\xi}_t \quad (7.3)$$

Formula (7.3) indeed shows that the cost of any arbitrary path in graph G' is equal to the respective cost of that path in graph G minus the cost $\bar{\xi}_t$ of the shortest path between vertices v_s and v_t in graph G . Therefore, the ranking of various single paths between vertices v_s and v_t by their cost in G' remains the same as the respective ranking of these paths regarding their costs in graph G . This property, in turn, implies that the respective pair of link-disjoint paths of the minimal total cost in graph G' is also of the minimal cost in graph G . Therefore, the task to determine the pair of link-disjoint pair of paths of the total minimal cost in G reduces to the problem of determining the related link-disjoint path pair of a minimal total cost in graph G' . ■

Computational Complexity

The computational complexity of Suurballe's algorithm is polynomial and comparable to the complexity of the standard version of Dijkstra's algorithm analyzed earlier in this chapter. This is because the most time-consuming operation of Suurballe's algorithm is the execution of Dijkstra's algorithm two times (in Steps 1 and 4, respectively). As shown in [21], it is possible to implement Suurballe's algorithm in a way that the upper bound on the time complexity is $O(|A| \log_{1+\frac{|A|}{|V|}} |V|)$.

The application of Suurballe's algorithm in solving problems of resilient routing is considered, e.g., in [8, 19] (multidomain routing), [10] (resilient routing in scenarios of geographically correlated failures), or [1] (survivable routing in optical networks).

7.3 Bhandari's Algorithm

The algorithm proposed by Bhandari in [3] aiming at finding a shortest pair of link-disjoint paths between given vertices v_s and v_t of the total minimal cost is a modified version of Suurballe's algorithm discussed in this chapter.

The sequence of the main steps of Bhandari's algorithm is the same as that of Suurballe's algorithm. However, compared to Suurballe's algorithm, Bhandari's method provides a simpler transformation of graph G . This transformation refers only to a certain subset of graph arcs implied by the arcs of path \mathcal{P}_1 found in Step 1. We should note that, as the transformation provided in Bhandari's algorithm leads to negative costs ξ'_h of certain arcs in graph G' , a dedicated shortest path algorithm is needed in Step 4 (instead of the conventional Dijkstra's algorithm) to determine path \mathcal{P}_2 . This new shortest path algorithm is indeed a modified version of Dijkstra's algorithm.

Before explaining the properties of Bhandari's algorithm, we will first elaborate on the modified Dijkstra's algorithm. After that, we will focus on two more problems, namely to establish a shortest pair of end-to-end node-disjoint paths and a shortest set k of end-to-end mutually node-disjoint paths, respectively.

The Modified Dijkstra's Algorithm

A modified version of Dijkstra's algorithm has been designed to be useful for the determination of shortest paths in graphs with negative costs of arcs (however, without negative cycles). It is different from the conventional version of Dijkstra's algorithm because the labels it assigns to visited vertices may turn out not to be final. These vertices may indeed become non-visited again once a lower cost path to these vertices has been determined. It is worth noting that this behavior can take place only if the costs of some arcs are negative. Otherwise, the modified Dijkstra's algorithm follows the same behavior as its conventional version.

The steps of the modified Dijkstra's algorithm are as follows:

1. Mark all vertices of G as belonging to the set of non-visited vertices. Assign the value of 0 to the label l_s (i.e., a tentative distance) of source vertex v_s and ∞ to labels l_i of all other vertices v_i in G . For each vertex v_i , set its predecessor index p_i to NULL. Set the initial vertex v_s as the current vertex.
2. For the current vertex, recalculate the distances for all its neighboring vertices (both visited and non-visited) via paths traversing the current vertex. If, for some nodes v_i , these new distances occur to be smaller than the related values of labels l_i , replace these labels with the values of the related recalculated distances. For each visited vertex, which received in this step an updated label, move it to the set of non-visited vertices.
3. Move the current vertex to the set of visited vertices.
4. If destination vertex v_t is marked as visited, terminate and return the path consisting of visited vertices between v_s and v_t as the shortest path.

- If destination vertex v_t belongs to the set of non-visited vertices, select a new current vertex from the set of non-visited vertices characterized by the lowest label value l_i , and go to Step 2.

In the modified Dijkstra's algorithm given above, additional operations that do not appear in its conventional version are underlined.

Example Execution of the Modified Dijkstra's Algorithm

In the example task to determine the shortest path between source vertex $v_s = 1$ and destination vertex $v_t = 2$ in a network graph from Fig. 7.7a, where costs of each arc are marked in blue, each vertex v_i in Fig. 7.7a is provided with a pair (l_i, p_i) marked in red, referring to its label l_i and the predecessor index p_i .

As shown in Fig. 7.7a, during the execution of the first iteration of the modified Dijkstra's algorithm, in Step 1 labels l_i of all vertices are assigned the respective tentative costs (equal to 0 for the source vertex v_s and to ∞ for all the other vertices), all vertices are classified as non-visited, and source vertex $v_s = 1$ is set as the current vertex (marked in orange). As illustrated in Fig. 7.7b, in Step 2, labels l_i (tentative distances) for unvisited neighboring vertices of the current vertex 1 (here of vertex 3) are recalculated, the related predecessor index p_i is set to 1 (since vertex 1 is the current vertex), and vertex 1 is moved in Step 3 to the set of visited vertices (marked in green). Vertex 3 is selected in Step 5 as a new current vertex (marked in orange).

During the second iteration of the algorithm, as illustrated in Fig. 7.7c, label l_i and predecessor index p_i for a non-visited vertex 2—a neighbor of the current vertex 3—are updated. Vertex 3 is moved to the set of visited vertices (marked in green), and vertex 2 of the lowest tentative distance (label $l_2=5$) is selected as the current vertex.

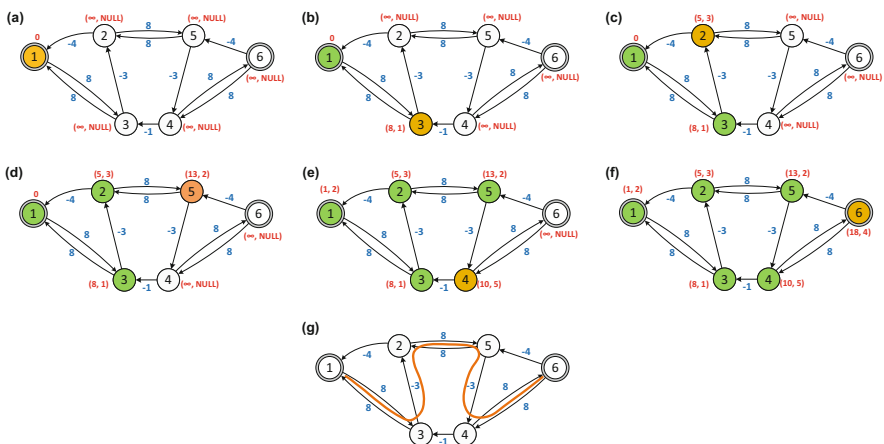


Fig. 7.7 Execution steps of the modified Dijkstra's algorithm when determining the shortest path between vertices 1 and 6 in the example network topology graph

During the third iteration of the algorithm, as provided in Fig. 7.7d, the label and predecessor index for the neighboring vertex 5 of the current vertex 2 are determined. The current vertex 2 is moved to the set of visited vertices, and vertex 5 of the lowest tentative distance of 13 is selected as the current vertex.

As illustrated in Fig. 7.7e, the fifth iteration results in updating the label and predecessor index for vertex 4, moving the current vertex 5 to the set of visited vertices, and selecting vertex 4 as a new current vertex.

In the sixth iteration, as illustrated in Fig. 7.7f, the label and predecessor index for vertex 6 are updated, vertex 4 is moved to the set of visited vertices, and vertex 6 is selected as a new current vertex. The assignment of the role of the current vertex to the destination vertex finalizes the execution of the algorithm (following the execution of Step 4 in the seventh iteration). The path (1, 3, 2, 5, 4, 6) shown in orange in Fig. 7.7g of the total cost of 18 is returned as the shortest one.

The Major Steps of Bhandari's Algorithm

When determining a shortest pair of link-disjoint paths between given two end vertices v_s and v_t , the steps of Bhandari's algorithm are as follows.

1. Determine path \mathcal{P}_1 from vertex v_s to vertex v_t using conventional Dijkstra's algorithm in graph G with arcs characterized by costs ξ_h .
2. Transform graph $G = (V, A)$ into graph $G' = (V, A')$ where the set of arcs A' includes arcs from A not traversed by path \mathcal{P}_1 and characterized by costs $\xi'_h = \xi_h$, as well as arcs directed oppositely to arcs traversed by \mathcal{P}_1 with negative costs $\xi'_h = -\xi_h$.
3. Determine path \mathcal{P}_2 from vertex v_s to vertex v_t using the modified Dijkstra's algorithm in graph G' with costs of arcs ξ'_h .
4. Return the final pair of paths \mathcal{P}'_1 and \mathcal{P}'_2 by merging paths \mathcal{P}_1 and \mathcal{P}_2 and removing common arcs.

Example of Bhandari's Algorithm Execution

In the example illustrated in Fig. 7.8, for a demand to establish a shortest pair of link-disjoint paths between vertices 1 and 6 when executing Step 1 of Bhandari's algorithm, path $\mathcal{P}_1 = (1, 2, 3, 4, 5, 6)$ marked in green in Fig. 7.8a is found. This path is determined by executing conventional Dijkstra's algorithm for graph G with arc costs ξ_h . Next, in Step 2, graph G' is created from the original graph G by replacing the original links traversed by path \mathcal{P}_1 with the arcs in a reverse direction only and making their costs negative, as illustrated in Fig. 7.8b. In Step 3, the modified Dijkstra's algorithm is used in graph G' to determine path $\mathcal{P}_2 = (1, 3, 2, 5, 4, 6)$ marked in orange in Fig. 7.8c. Finally, in Step 4, the shortest pair of disjoint paths $\mathcal{P}'_1 = (1, 2, 5, 6)$ and $\mathcal{P}'_2 = (1, 3, 4, 6)$ of the total cost of 33 shown in Fig. 7.8d is obtained from paths \mathcal{P}_1 and \mathcal{P}_2 .

Determination of a Shortest Pair of Node-Disjoint Paths

A shortest pair of node-disjoint (vertex-disjoint) paths established between given vertices v_s and v_t is a pair of paths with no common transit vertices and characterized by the smallest total cost. As discussed earlier in this book, the requirement on

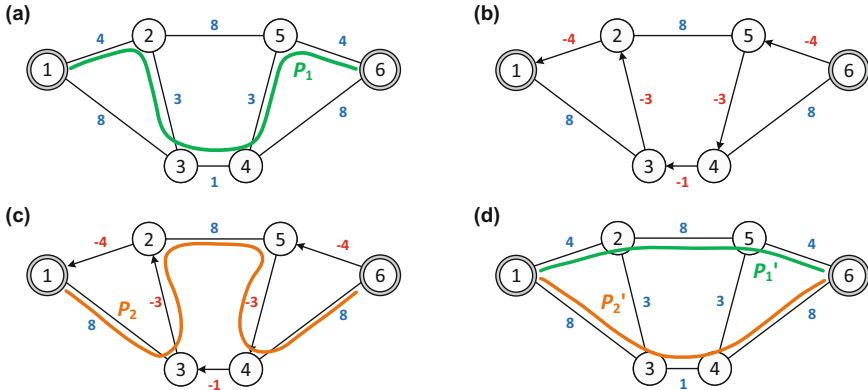


Fig. 7.8 Execution steps of Bhandari's algorithm when determining the shortest pair of paths \mathcal{P}'_1 and \mathcal{P}'_2 between vertices 1 and 6 in the example network topology graph

vertex-disjointness is stronger than the assumption of edge-disjointness, i.e., a pair of vertex-disjoint paths is also edge-disjoint, but not vice versa.

To obtain a shortest pair of vertex-disjoint paths using Bhandari's algorithm, an update of graph G transformation procedure is needed. This necessity is illustrated in Fig. 7.9 showing results of execution of Step 4 of Bhandari's algorithm when determining a shortest pair of link-disjoint paths between vertices 1 and 6. Figure 7.9 illustrates two possible variants of path \mathcal{P}_2 after the execution of Step 4 of Bhandari's algorithm. However, we can see that although both variants lead to returning a final shortest pair of link-disjoint paths, only in the case of variant (b) in Fig. 7.9, the final shortest pair of paths \mathcal{P}'_1 and \mathcal{P}'_2 returned by Bhandari's algorithm is vertex-disjoint.

The final shortest pair of paths \mathcal{P}'_1 and \mathcal{P}'_2 obtained from paths \mathcal{P}_1 and \mathcal{P}_2 shown in Fig. 7.9 is vertex-disjoint when path \mathcal{P}_2 determined in Step 4 of Bhandari's algorithm execution has at least one common arc with path \mathcal{P}_1 (e.g., arc (4, 3) in Fig. 7.9b). This observation, in turn, forms the basis of the *vertex splitting* operation proposed in [3] added to the graph transformation procedure concerning all transit vertices of path \mathcal{P}_1 to assure the vertex-disjointness of the final shortest pair of paths.

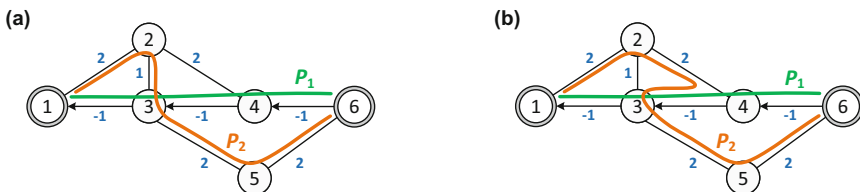
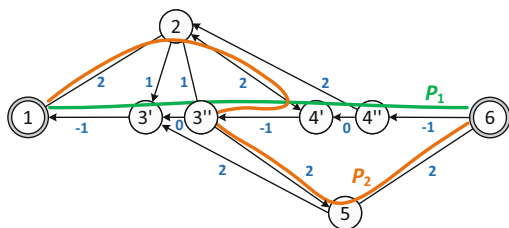


Fig. 7.9 Two alternate versions of paths \mathcal{P}_2 marked in orange obtained during the execution of Bhandari's algorithm when determining the shortest pair of paths between vertices 1 and 6

Fig. 7.10 Illustration of paths \mathcal{P}_1 and \mathcal{P}_2 determined in Step 4 of Bhandari's algorithm between vertices 1 and 6 in a transformed graph G' with node splitting operation applied to transit nodes of path \mathcal{P}_1



This vertex-splitting operation consists of the following three steps.

1. Replacing each transit vertex v_i of path \mathcal{P}_1 by two vertices v'_i and v''_i .
2. Connecting each pair of vertices v'_i and v''_i by a zero-cost arc in opposite direction to path \mathcal{P}_1 .
3. Replacing each edge between the original transit vertex v_i of path \mathcal{P}_1 and a non-transit vertex v_j of \mathcal{P}_1 by a pair of oppositely directed arcs, each such arc of cost $\xi'_h = \xi_h$ as shown in Fig. 7.10. As a result of this operation:
 - Arcs that were incident into a given transit vertex v_i in graph G are now incident into the respective vertex v'_i in graph G' .
 - Arcs that were incident out of a given transit vertex v_i in graph G now became incident out of the respective vertex v''_i in graph G' .

As illustrated in Fig. 7.10, since the vertex splitting routine forces path \mathcal{P}_2 to traverse some arcs originally traversed by path \mathcal{P}_1 (arc $(4', 3'')$ in this case), the final shortest pair of paths \mathcal{P}'_1 and \mathcal{P}'_2 to be returned by Bhandari's algorithm is indeed vertex-disjoint.

Finding a Shortest Set of k Disjoint Paths Using Bhandari's Algorithm

As discussed earlier in this book, a set of k end-to-end node/link-disjoint paths determined between a given pair of vertices v_s and v_t can provide continuity of transmission between that pair of end nodes in scenarios of simultaneous failures of $k-1$ network nodes (links). Determination of such a set of k disjoint paths can be realized by using Bhandari's algorithm iteratively with the related transformations of the original graph G into graph G' done in each i -th iteration considering the properties of $i-1$ disjoint paths found in former iterations.

For instance, for a demand to establish a shortest set of three link-disjoint paths between a given pair of vertices 1 and 6 in Fig. 7.11, in the first iteration of Bhandari's algorithm a pair of link-disjoint paths is first determined. This is done by calculating path \mathcal{P}_1 (Fig. 7.11a), followed by providing path \mathcal{P}_2 for the transformed graph G' (Fig. 7.11b) and returning these two paths as the result of the first iteration. Next, in the second iteration, a new transformed graph G' is first created (see Fig. 7.11c), path \mathcal{P}_3 marked in red in Fig. 7.11c is determined, and the shortest set of three link-disjoint paths \mathcal{P}'_1 , \mathcal{P}'_2 , and \mathcal{P}'_3 is finally returned, as illustrated in Fig. 7.11d. The characteristics of graph G make the final shortest set of paths also vertex-disjoint (although the requirement on vertex-disjointness was not imposed in this example).

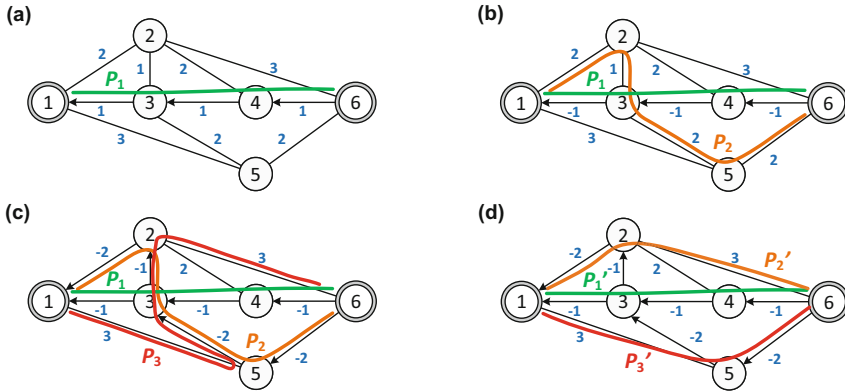


Fig. 7.11 Illustration of results of consecutive iterations of Bhandari’s algorithm to determine a shortest set of $k = 3$ link-disjoint paths between vertices 1 and 6

The three main features of this iterative scheme can be thus summarized as follows:

- For a demand to establish a shortest set of k disjoint paths between a given pair of vertices v_s and v_t , $k - 1$ iterations of Bhandari’s algorithm are required.
- During the execution of a given i -th iteration, transformation of graph G is applied in the context of $i - 1$ disjoint paths found in the previous iteration.
- After executing a given iteration, the number of disjoint paths determined so far is extended by one.

Computational Complexity of Bhandari’s Algorithm

Similar to Suurballe’s algorithm, the computational complexity of Bhandari’s method is also polynomial and bounded above by $O(|V|^2)$. The primary reason is that both these algorithms rely on executing a certain number of times Dijkstra’s algorithm (or its modified version) of $O(|V|^2)$ complexity, which is the most time-consuming part of the execution of these algorithms.

7.4 k -Penalty Algorithm to Determine Sets of Disjoint Paths in Multi-cost Networks

Methods useful in the determination of a pair (or a set of k) disjoint paths between a given pair of vertices v_s and v_t discussed so far in this chapter operate under the assumption that the same cost ξ_h of each arc a_h of graph G is valid when determining both the primary and backup paths. Such a network property is often called the *single-cost network* case [20].

As already discussed in this chapter, the problem of finding a pair (or a set of k) disjoint paths for a single demand to provide transmission between a given pair of vertices v_s and v_t in a way that the total cost of the resulting set of paths is minimized can be solved optimally by methods of polynomial computational complexity, such as Suurballe's or Bhandari's algorithms.

However, it must be noted that using these algorithms to determine the shortest sets of paths for a set of demands (instead of only one demand) does not guarantee reaching the optimal solution. This is due to the \mathcal{NP} -hardness of the latter problem, already discussed in Chap. 6 of this book.

Also, the costs ξ_h of arcs may not be defined in the same way for all paths traversing these arcs. For instance, they may be formulated differently for different types of paths. Such a situation occurs, e.g., in the case of the backup path sharing scheme discussed in detail in Chap. 4 of this book, where the cost of arcs in backup path calculation may be lower than the respective one applied when determining the primary path. This, in turn, leads to the concepts of either *dual-cost networks* (where each arc a_h is provided with two variants of its cost ξ_h^1 and ξ_h^2) or, generally speaking, *multi-cost networks*, allowing each arc to be characterized by k different costs of edges used for determining each of k disjoint paths. However, even in the simpler case of dual-cost networks, the problem of determining a pair of paths of the lowest total cost is \mathcal{NP} -hard even for a single demand [11, 22].

In this part of the chapter, we explain the characteristics of our k -Penalty method from [15] designed to determine a set of k node-disjoint paths between a given pair of vertices v_s and v_t dedicated to multi-cost networks, where each arc a_h may be assigned k costs $\xi_h^1, \xi_h^2, \dots, \xi_h^k$ for each of k paths. This approach is relatively similar to the *active path first (APF)* scheme highlighted, e.g., in [12].

Similar to the APF method, the primary path between a given pair of vertices v_s and v_t is determined first. However, notable differences between k -Penalty and APF method can be noticed later when determining the backup communication paths. In particular, in the APF method, to assure mutual disjointness of all k paths of demand, when finding each next (i -th) path, APF sets to ∞ the costs of links traversed by all former $i-1$ paths (when link disjointness is required) or the costs of all links incident to all transit nodes of all former $i-1$ paths (when nodal disjointness is needed). Setting the costs of these links to ∞ excludes them from calculations of further paths.

However, this feature raises the risk for the occurrence of a *trap problem* being a scenario when the algorithm fails to determine a set of disjoint paths despite the availability of network resources, as illustrated in Fig. 7.12. In particular, after determining path \mathcal{P}_1 in Fig. 7.12a and setting all costs of edges traversed by \mathcal{P}_1 to ∞ (thus excluding them from graph G , as given in Fig. 7.12b), calculation of path \mathcal{P}_2 is not possible, although sufficient network resources are available and, therefore, allowing for calculation of a pair of node-disjoint paths \mathcal{P}_1 and \mathcal{P}_2 shown in Fig. 7.12c.

To eliminate the trap problem, in our k -Penalty scheme the costs of arcs traversed by previous $i - 1$ paths (arcs incident to transit nodes of previous $i - 1$ paths) in the case of a link (nodal) disjointness are not set to ∞ but are increased at the beginning

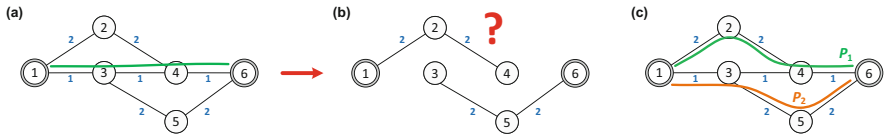


Fig. 7.12 Illustration of a trap problem for a demand to establish two link-disjoint paths between vertices 1 and 6

of a given i -th iteration by the total cost of previously determined $i - 1$ disjoint paths. Traversing these “forbidden” links will thus cost more (i.e., implying that a certain “penalty” has to be paid) to try to prevent the next path from traversing the arcs utilized by the already found $i - 1$ paths of a demand.

For the example demand to establish a set of three node-disjoint paths between vertices 1 and 10 in Fig. 7.13, this means that after determining path \mathcal{P}_1 marked with a dashed black line in Fig. 7.13a, the total cost of all arcs incident to transit nodes of path \mathcal{P}_1 is increased by the total cost of that path, i.e., by 5. Therefore, as given in Fig. 7.13b, before determining path \mathcal{P}_2 , the increase of arc cost by 5 refers to arcs (1, 2), (2, 5), (5, 7), (7, 9), (9, 10) traversed by path \mathcal{P}_1 and other arcs incident to transit nodes of path \mathcal{P}_1 .

If a given i -th path happens to be not disjoint with previously determined $i - 1$ paths of a demand, the costs of all “conflicting” arcs (i.e., arcs traversed jointly with previously calculated $i - 1$ paths or arcs incident to transit nodes of previously determined $i - 1$ paths in the case of a link and nodal disjointness, respectively) are permanently increased by the total cost of i -th path, all calculated paths are removed, and the algorithm starts its execution from the beginning.

In our example, this indeed happens to path \mathcal{P}_2 in Fig. 7.13b, so as illustrated in Fig. 7.13c, both paths \mathcal{P}_1 and \mathcal{P}_2 are removed, the costs of all conflicting arcs (i.e., arcs (4, 9), (7, 9), and (9, 10) in this case) are increased by the total cost of path \mathcal{P}_2 (i.e., by 17), and the algorithm restarts the calculation of paths.

The algorithm returns the set of k disjoint paths possibly encountering several conflicts or (rarely) returns no solution after reaching the predefined maximum number of allowed conflicts. In our example, the algorithm encounters a second conflict as given in Fig. 7.14a, but after that, it finally manages to return the set of three node-disjoint paths illustrated in Fig. 7.14d.

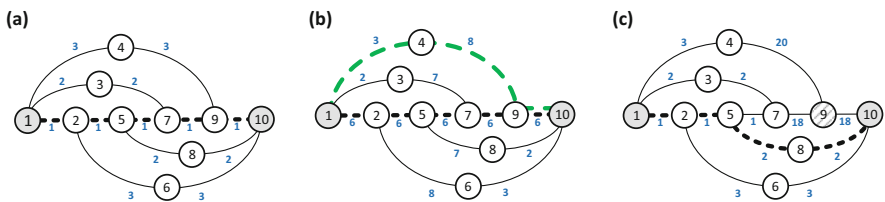
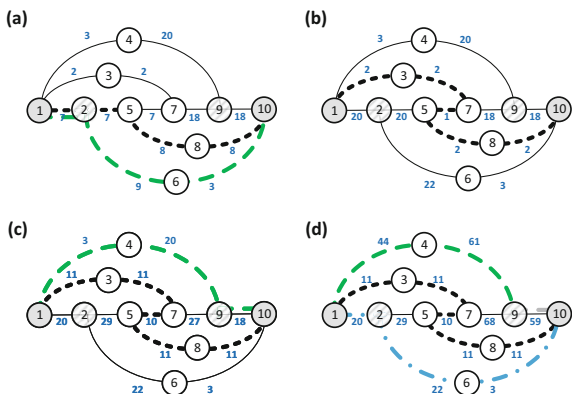


Fig. 7.13 Illustration initial steps of the execution of the k -Penalty algorithm for a demand to establish two node-disjoint paths between vertices 1 and 10

Fig. 7.14 Illustration of further steps of the execution of the k -Penalty algorithm for a demand to establish two node-disjoint paths between vertices 1 and 10



7.5 Summary

In this chapter, our analysis was focused on efficient algorithms to determine the shortest pairs (or shortest sets of k) of end-to-end paths for single demands. A particular focus was on approaches aimed at optimizing, for given demands, the total cost of disjoint paths (i.e., according to the assumptions of the min-sum objective). As discussed in this chapter, computationally efficient methods exist for this objective, such as Suurballe’s algorithm or Bhandari’s scheme, which are able to provide optimal solutions. This assumption is true for the so-called single-cost network scenario, where costs ξ_h of arcs a_h do not depend on particular paths to be established.

However, a notable set of problem variants (e.g., referring to scenarios of multi-cost networks or optimizing the total cost of shortest set of paths simultaneously for all demands) belongs to the class of \mathcal{NP} -hard problems implying the need to use suboptimal heuristics for larger problem instances.

? Questions

1. Explain the regional disjointness/geographical diversity property of communication paths and provide example scenarios of its application.
2. Explain the differences between the min-sum, min-min, and min-max objectives for the problem to determine the shortest pairs of end-to-end disjoint paths.
3. Discuss the purpose and properties of Dijkstra’s algorithm.
4. Explain the purpose of Suurballe’s algorithm and illustrate the main steps of its execution.
5. Discuss the purpose and illustrate the main steps of Bhandari’s algorithm execution.
6. Compare the properties of Suurballe’s and Bhandari’s algorithms.

7. Which algorithm would you recommend using for the calculation of a shortest set of k end-to-end node-disjoint paths of the total minimal cost (i.e., min-sum)? Justify your answer.
 8. Explain the differences between single-cost and multi-cost network scenarios and associate each of them with at least one known practical problem.
 9. Explain the scenario of the trap problem and discuss the factors that may trigger this problem.
 10. Discuss possible ways to mitigate the trap problem by algorithms of resilient routing.
-

References

1. Beshir, A., Kuipers, F., Van Mieghem, P., Orda, A.: On-line survivable routing in WDM networks. In: Proceedings of the 2009 21st International Teletraffic Congress (ITC'09), pp. 1–8 (2009)
2. Bhandari, R.: Optimal physical diversity algorithms and survivable networks. In: Proceedings of the 2nd IEEE Symposium on Computers and Communications (ISCC'97), pp. 433–441 (1997)
3. Bhandari, R.: Survivable Networks: Algorithms for Diverse Routing. Kluwer Academic Publishers, Dordrecht (1999)
4. Chen, W.-K.: Theory of Nets: Flows in Networks. John Wiley & Sons, Hoboken (1990)
5. Dijkstra, E.W.: A note on two problems in connexion with graphs. *Numer. Math.* **1**, 269–271 (1959)
6. Floyd, R.W.: Algorithm 97: shortest path. *Commun. ACM* **5**(6), 345 (1962)
7. Fredman, M.L., Tarjan, R.E.: Fibonacci heaps and their uses in improved network optimization algorithms. In: Proceedings of the 25th Annual Symposium on Foundations of Computer Science, pp. 338–346 (1984)
8. Gao, C., Cankaya, H.C., Jue, J.P.: Survivable inter-domain routing with Suurballe-based intra-domain disjointness information in multi-domain optical networks. In: Proceedings of the OFC/NFOEC'12, pp. 1–3 (2012)
9. Gomes, T., Jorge, L., Girao-Silva, R., Yallouz, J., Babarzi, P., Rak, J.: Fundamental schemes to determine disjoint paths for multiple failure scenarios. In: Guide to Disaster-Resilient Communication Networks, pp. 1–43. Springer, Berlin (2020)
10. Gour, R., Kong, J., Ishigaki, G., Yousefpour, A., Hong, S., Jue, J.P.: Survivable routing in multi-domain optical networks with geographically correlated failures. In: Proceedings of the 2017 IEEE Global Communications Conference (GLOBECOM'17), pp. 1–6 (2017)
11. Liu, Y.: Spare capacity allocation method analysis and algorithms. PhD thesis. University of Pittsburgh (2001)
12. Liu, Y., Tipper, D., Siripongwutikorn, P.: Approximating optimal spare capacity allocation by successive survivable routing. *IEEE/ACM Trans. Networking* **13**(1), 198–211 (2005)
13. Moy, J.: OSPF version 2, Request for Comments 2328, <https://www.rfc-editor.org/rfc/rfc2328.html> (1998)
14. Oran, D. (Eds.): OSI IS-IS Intra-domain routing protocol. Request for Comments 1142, <https://www.rfc-editor.org/rfc/rfc1142.html> (1990)
15. Rak, J.: k -Penalty: a novel approach to find k -disjoint paths with differentiated path costs. *IEEE Commun. Lett.* **14**(4), 354–356 (2010)

16. Rak, J.: Fast service recovery under shared protection in WDM networks. *IEEE/OSA J. Lightwave Technol.* **30**(1), 84–95 (2012)
17. Rak, J., Molisz, W.: A new approach to provide the differentiated levels of network survivability under a double node failure. In: *Proceedings of the 11th International Conference on Transparent Optical Networks (ICTON'09)*, pp. 1–4 (2009)
18. Ramamurthy, B., Sahasrabudde, L., Mukherjee, B.: Survivable WDM mesh networks. *IEEE/OSA J. Lightwave Technol.* **21**(4), 870–883 (2003)
19. Samonaki, M., Serna, C.B., Mas-Machuca, C.: Survivable node-disjoint routing in multi-domain networks. In: *Proceedings of the IEEE International Conference on Communications (IEEE ICC'23)*, pp. 4578–4583 (2023)
20. Suurballe, J.W.: Disjoint paths in a network. *Networks* **4**(2), 125–145 (1974)
21. Suurballe, J.W., Tarjan, R.E.: A quick method for finding shortest pairs of disjoint paths. *Networks* **14**(2), 325–336 (1984)
22. Xu, D., Chen, Y., Xiong, Y., Qiao, CH., He, X.: On the complexity of and algorithms for finding the shortest path with a disjoint counterpart. *IEEE/ACM Trans. Networking* **14**(1), 147–158 (2006)

Part III
Case Studies

Chapter 8

Resilience of Future Internet Communications



Over the last 40 years, we have been observing a gradual evolution of the Internet from an academic network toward a widespread commercial architecture. Indeed, the classic Internet, designed in the 1970s by Vinton G. Cerf and Robert E. Kahn [14] as a network of networks, evolved from its predecessor—the ARPANET academic network connecting computing sites at universities across the USA [43].

The Internet was originally meant to be a computer communication network of datagram orientation only (i.e., mainly for conservative data traffic usage). Afterward, it has been progressively adapted to meet the evolving diverse expectations of end users concerning services and daily use applications to enhance the quality of life [9]. In particular, owing to the observed convergence of telecommunications, media, and information technology, the Internet is now becoming an integrated system enabling accessing, distributing, processing, storing, and managing the content [60].

However, the main architectural changes to the Internet architecture have been mostly the “last minute” fixes/updates, while important modifications have recently become practically infeasible [61]. Besides, the conventional Internet has already reached its limits where even minor improvements do not have much chance to succeed. Therefore, a comprehensive Internet transformation from a simple “host-to-host” packet exchange environment toward a complex networking paradigm built around the content and end users instead of network nodes is inevitable [55]. Following [60], major challenges driving the research efforts toward the Internet of the future include:

- Identification of a large set of network nodes
- Scalability and efficiency of network and mobility management
- Quality of Service
- Security
- Resilience
- Energy efficiency

Since, without doubt, future knowledge society will be built on the Internet communications base, any limitations referring to the efficiency of the Internet must be defeated. Otherwise, end users may not be able to fully benefit from several emerging technologies, e.g., advanced wireless/mobile communications, broadband optical networking, huge storage capacity, or innovative techniques, including sensors and energy sources [60].

All these demands have driven the research community to design the respective *Future Internet (FI)* solutions within various research activities intensively supported in the last decade, for instance, by the European Commission [25], National Science Foundation (NSF) in the USA [52], and others. As a result, one/two Future Internet Assemblies [53] have been organized every year since 2008 to discuss the outcomes of numerous ongoing FI research projects, as well as to summarize their achievements in the respective FIA books (see, for instance, [6, 23, 67]).

Apart from the European activities in this area, including, e.g., 4WARD [63], FIRE [27], GEANT2 [30], or IIP [29], there have also been other important initiatives from the USA (e.g., FIA [28], FIND [52, 54], GENI [34], MobilityFirst [48], or NDN [49]), Japan (e.g., AKARI [3]), and China (e.g., CERNET [15]).

It is worth noting that there is no standardized/publicly accepted definition for the Future Internet. Instead, it is mainly described by a set of relevant capabilities not existing in the classic Internet architecture. As discussed in [6, 23, 24, 55, 56, 61], the list of desired functionalities of the Future Internet architecture includes the following:

- *Content-oriented networking (CON)* being an opposite solution to the conventional host-to-host information delivery, as the primary utilization of the Internet visibly evolves into data/content distribution. A widely observable trend is to design the architecture of the Future Internet “around people” instead of around machines [55], implying the need to update the IP layer to provision content distribution and making information (rather than conventional IP addresses) the primary search goal.
- *Cloud computing/communications*. Combining data centers and computation possibilities into the cloud to form a “computing utility” available over the Internet is seen as an efficient solution to provide the global-scale resource and computation capabilities.
- Novel *Human-computer interaction (HCI)* techniques driven by the availability of cheap sensor technology that may soon revolutionize the way humans interact with computers (i.e., via human gestures, as well as displays integrated with objects of everyday use).
- Real-time access to huge-scale multimedia content (known as the *Big Data* paradigm), e.g., to 3D and cognitive content, virtual, and augmented world.
- *Users acting as service providers*, e.g., selling photos, or operating as stream broadcasters. Other examples include inter-vehicular communications (as discussed in Chap. 10 of this book), where a system installed in a vehicle may automatically inform other vehicles about accidents, ice on the road, etc.

- *Personalized services* including individualized (or context-aware) search results, person-(group-)-oriented services targeting specific interest groups.
- *Mobility-centric orientation enabling ubiquitous access to information anytime and anywhere*. Due to the observed shift from stationary (PC-based) computing to mobile computing, as well as the convergence of heterogeneous networks, mobility is one of the key functionalities of the Future Internet. It should be thus considered as a norm, rather than an exception.
- *Interconnection of devices, sensors, etc.* (known as the *Internet of Things—IoT* concept) into networks of diverse physical objects, such as vehicles, mobile phones, etc.
- *Networks programmability* offered by virtualized Software-Defined Networks with network control functions being directly programmable and decoupled from forwarding [62, 73].
- *Security* mechanisms forming an inherent part of the FI architecture (as opposed to functioning as an additional overlay in the classic Internet), which is justified from both technical and economic reasons.
- *Energy efficiency*. The gradual growth of Internet traffic volume increases energy consumption by networking equipment to accommodate the demands. One of the solutions to save energy may be switching off the devices or putting them into sleep mode in inactive periods.
- *Availability and disruption tolerance*. The Internet is currently viewed as an important element of critical infrastructure (similar to, e.g., fresh water supplies or power grids). Therefore, the architecture of the Future Internet should also be resistant to disruptions of any kind, providing an alternate means for content distribution/processing in the face of failures and guaranteeing fast recovery of affected network elements.

Another classification of FI main research areas from [60] is presented in Fig. 8.1. In particular, issues of Future Internet resilience are included in areas #2 (Future Internet modeling, analysis, and design) and #3 (Future Internet network architectures), respectively.

To support these functionalities, one of the possible ways proposed by the research community is the so-called *clean-slate* concept, in which applying certain solutions may be done under the assumption that other parts of the architecture remain unchanged [26, 55]. Therefore, deploying a number of clean-slate solutions may not necessarily lead to a new architecture of the Internet. Besides, redesign of the Internet architecture should utilize the best practices from the past and be evolvable and flexible to accommodate future demands [55].

In the clean-slate paradigm, there are practically no restrictions on the architectural design of the Future Internet. However, since today’s Internet connects billions of nodes and supports millions of applications, even though the new architecture is expected to be revolutionary, its application should be done on an evolutionary basis. In particular, “new technology” nodes should be able to communicate over the existing infrastructure. Researchers are convinced that the Future Internet must be designed dynamically and modularly, allowing for further adaptive changes [9].

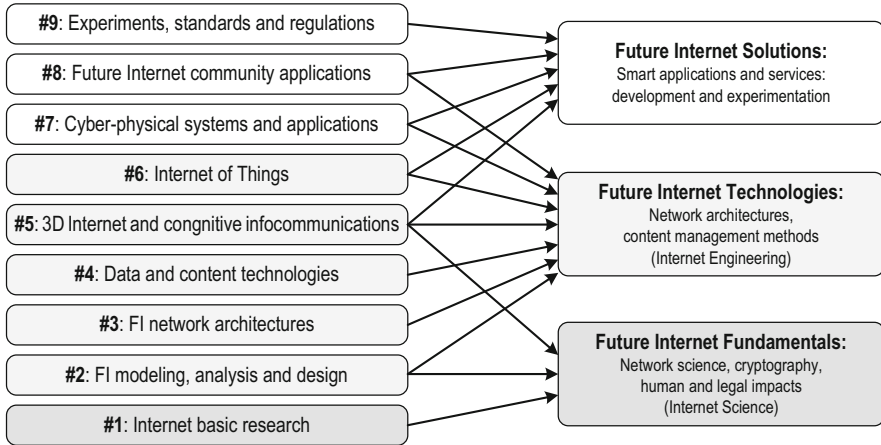


Fig. 8.1 Future Internet research areas in relation to their goals from [60]

In the remaining part of this chapter, we will discuss in detail the key research topics and requirements for the FI architecture (Sect. 8.1), present our solutions to network resource provisioning necessary to provide network resilience (Sect. 8.2), and describe in Sect. 8.3 three proposals to improve the resilience of content-oriented FI networking. The chapter is summarized in Sect. 8.4.

8.1 Key Research Topics and Requirements for the Future Internet Architecture

Considering the architectural requirements of the Future Internet, a distinction between providers of a network infrastructure (i.e., physical resources) and providers of network services becomes apparent. It justifies the need for *virtual networks* (VNs) implementation. Such a scheme allowing for leasing physical network resources (e.g., node processing power, link capacity, etc.) to deploy the end-to-end services, as well as having a certain control on the usage of these leased resources (being one of the main foundations of virtual local area networks (VLANs), virtual private networks (VPNs), or overlay networks [18]), has now evolved concerning the Future Internet architecture into the *virtualization* scheme [11, 68].

Following [64], *network virtualization* benefits from decoupling the single role of common *Internet service providers* (ISPs) into two independent entities: *infrastructure providers* (InPs) managing the physical infrastructure of networks and *service providers* (SPs) offering the end-to-end services via virtual networks

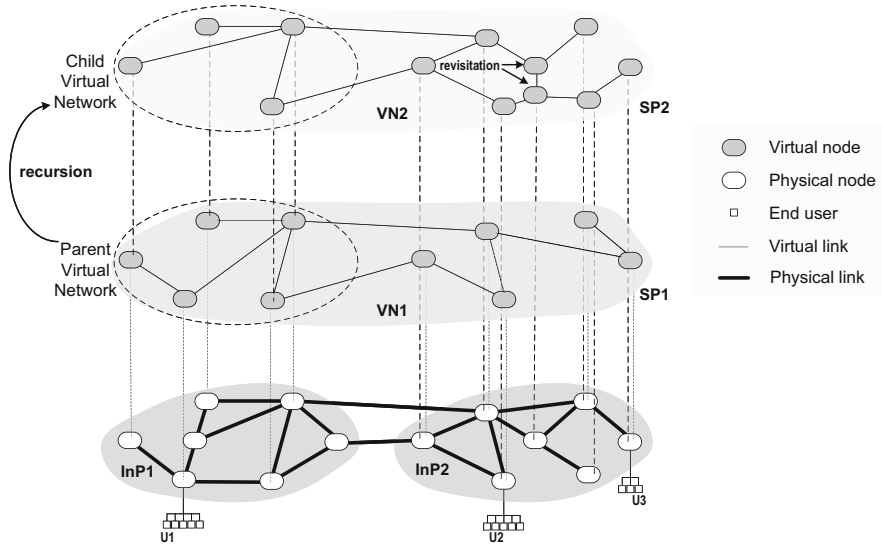


Fig. 8.2 Example network virtualization environment (NVE) with virtual network VN1 created on top of InP1 and InP2 resources and VN2 additionally implementing partial parent-child relationship with VN1

created and managed by them based on aggregating resources from multiple InPs.¹ In such a virtualized networking scheme, the set of multiple heterogeneous network architectures owned by different service providers that can be utilized to form a virtual network by the InP is often referred to as the *network virtualization environment (NVE)* [18], as presented in Fig. 8.2.

A virtual network is the basic entity in any NVE. It consists of *virtual nodes* (hosted on a given physical node) linked together by *virtual links* typically provided by paths in the physical network utilizing the respective resources from the physical layer (mainly link capacities and processing power of transit physical nodes). Therefore, end users can benefit in the NVE from multiple virtual networks managed by different SPs for a number of services.

Following [18], network virtualization implies:

- *Coexistence* of many virtual networks of different SPs utilizing physical resources from at least one InP [7]
- *Inheritance* allowing child VNs to inherit the architectural attributes of their parent VNs [43]

¹ In general, the idea of identifying the separate roles of InPs and SPs is not new (it has been proposed for the *information society* paradigm before).

- *Recursion* being a parent-child relationship for virtual networks (see Fig. 8.2) creating the VN hierarchy (i.e., VNs built on top of other VNs), often referred to as *nesting* [45]
- *Revisitation* enabling hosting multiple virtual nodes from a given VN by a single physical node [64]

Network virtualization leading to transformation into logical networks built on top of the existing physical network infrastructure can be thus viewed as an evolved form of the overlay networking concept. Like the original idea of overlays, deploying new network virtualization environments does not require changes to the underlying physical network once it is set up to support network virtualization [18]. Therefore, virtualization is expected to be a scalable scheme that offers relatively easy and inexpensive means to configure communication environments for end-to-end services.

A proper evaluation of Future Internet solutions requires utilization of *large-scale testbeds* [55]. However, several ongoing (and completed) projects related to FI architectures use either small testbeds (e.g., of a national scale), multiple heterogeneous testbeds (e.g., multiple testbeds with differentiated schemes deployed), or simply infrastructure of the classic Internet, as well as testbeds of previous research project not related to FI architectures.

In a network virtualization environment, a proper reservation of physical network resources is necessary for provisioning end-to-end services by service providers to meet the stringent Quality of Service requirements. As such, it is also essential to support resilient routing (for instance, by efficient reservation of network resources for alternate paths establishment) in the face of differentiated challenges and should be considered an essential part of any Future Internet architecture.

Therefore, in Sect. 8.2, we will highlight the concept of network resource provisioning for virtualization environments proposed in the example framework of one of the major European research projects on Future Internet architecture by researchers from Polish technical universities and research centers in 2010–2013, called Future Internet Engineering [29]. In particular, solutions to the network resource provisioning problem implemented in “System IIP”—the core part of the designed FI architecture—allow for automatic reservation of physical network resources for coexisting virtual networks of differentiated transmission types.

The respective network resource provisioning module we implemented for System IIP includes three Integer Linear Programming models introduced to obtain the optimal solutions to the respective network resource provisioning problems. This module, being an important part of the management system, is to be utilized periodically to update the resource provisioning solutions to respond to changes in end-to-end demands over time.

8.2 Network Resource Provisioning Concepts in the “System IIP” Future Internet Architecture

Among several completed and ongoing projects related to the Future Internet architecture design, the Polish initiative called Future Internet Engineering resulted in the four-layer architecture of the so-called System IIP, comprising in the bottom-up order: L1—physical infrastructure layer, L2—virtualization layer, L3—Parallel Internets layer, and L4—virtual networks layer [12, 13]. This architecture, characterized by the ability to adjust its properties based on the required transmission scheme, was designed to provide the coexistence of differentiated types of Parallel Internets (PIs) within one physical infrastructure, including IPv6 with Quality of Service (IPv6_QoS), Content-Aware Network (CAN), and Data Stream Switching (DSS), as shown in Fig. 8.3.

In this section, we focus on the Future Internet resource provisioning issues, particularly concerning architectural aspects of the L1/L2 resource provisioning module we implemented in the System IIP architecture. Allocation of requested resources is provided here periodically in a static way. Therefore, before each consecutive update of the network resource provisioning solution, a traffic matrix is prepared in advance. Additional constraints (e.g., on link propagation delay concerning given PIs) may also be introduced.

The objective of the network resource provisioning module is to assign elementary resources (such as link capacity or node processing power) to three investigated Parallel Internets and to the management system enabling virtualization of nodes and links [16, 20]. The following three schemes aimed at providing the optimal solution to the respective Linear Programming (LP) problems were implemented in the System IIP architecture, as described in [36].

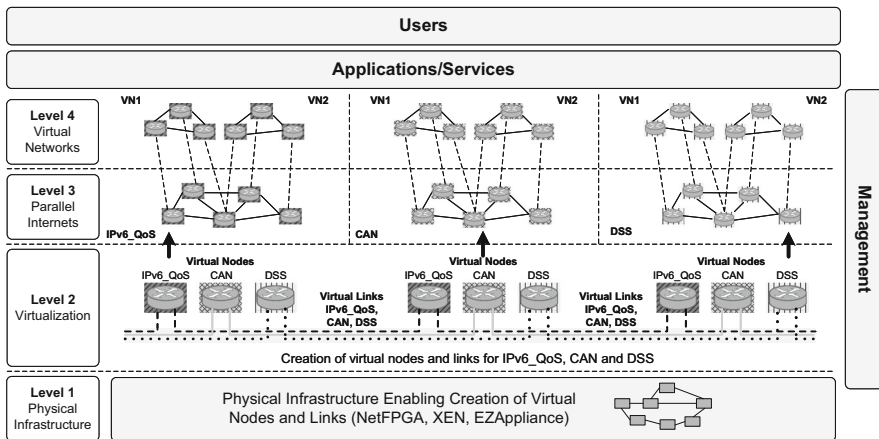


Fig. 8.3 Architecture of System IIP from [12]

Model 8.1 Formulation of Link Bandwidth Utilization Optimization Problem Respecting Basic Requirements on Routing (LBUO)

Symbols

$G(N,A)$	Directed network, where N and A are the sets of network nodes and directed arcs, respectively; each network link is represented by two opposite arcs $a_h = (i, j)$ and $a'_h = (j, i)$; and $ N $ and $ A $ are the numbers of network nodes and arcs, respectively.
T	Set of transit (forwarding) nodes
$N \setminus T$	Set of edge nodes
M	Set of instances of Parallel Internets (here, referring to IPv6_QoS, CAN, and DSS Internets, respectively; $ M = 3$),
D_m	Set of demands r for each m -th Parallel Internet, $r = 1, 2, \dots, D_m $

Constants

c_h	Total capacity available at arc a_h
$\hat{c}_{m,h}$	The lower bound on capacity (i.e., fraction of link capacity) required at arc a_h for m -th instance of PI
$c_{r,m}$	Volume of demand r from m -th instance of PI
$s_{r,m}$	Source node of demand r from m -th instance of PI
$t_{r,m}$	Destination node of demand r from m -th instance of PI

Variables

$x_{m,h} \geq 0$	Capacity assigned for m -th PI at arc a_h
$z_{r,m,h} \geq 0$	Capacity assigned for demand r of m -th PI at arc a_h

Objective

It is to minimize the total bandwidth consumption for delivering the traffic defined by formula (8.1).

$$\min \varphi(x) = \sum_{m \in M} \sum_{h \in A} x_{m,h} \quad (8.1)$$

Constraints

1. To assure that the amount of flow leaving node n via arc a_h for m -th Parallel Internet is the same as the amount of flow received at the other end of arc $a_h = (i, j)$:

$$\sum_{n: a_h=(i,n) \in A} x_{m,h} = \sum_{n: a_h=(n,j) \in A} x_{m,h}; \quad m \in M; \quad h \in A \quad (8.2)$$

2. To provide flow conservation rules at transit nodes for total capacities assigned to each m -th PI:

$$\sum_{\substack{h \in \{h: a_h = (t, j) \in A; \\ j = 1, 2, \dots, |N|; j \neq n\}}} x_{m,h} = \sum_{\substack{h \in \{h: a_h = (i, t) \in A; \\ i = 1, 2, \dots, |N|; i \neq n\}}} x_{m,h}; \quad m \in M; \quad t \in T \quad (8.3)$$

3. On the lower bound on the aggregate capacity assigned to m -th PI at arc a_h :

$$x_{m,h} \geq \hat{c}_{m,h} c_h \quad m \in M; \quad h \in A \quad (8.4)$$

4. On the upper bound on the total flow passing via network links for all PIs:

$$\sum_{m \in M} x_{m,h} \leq c_h; \quad h \in A \quad (8.5)$$

5. To provide flow conservation rules for demands r of each m -th PI:

$$\sum_{\substack{h \in \{h: a_h = (n, j) \in A; \\ j = 1, 2, \dots, |N|; j \neq n\}}} z_{r,m,h} - \sum_{\substack{h \in \{h: a_h = (i, n) \in A; \\ i = 1, 2, \dots, |N|; i \neq n\}}} z_{r,m,h} = \begin{cases} c_{r,m}, & \text{if } n = s_{r,m} \\ -c_{r,m}, & \text{if } n = t_{r,m} \\ 0, & \text{otherwise} \end{cases} \quad (8.6)$$

where $r \in D_m$, $m \in M$, and $n \in N$.

6. To guarantee that the aggregate flow transported via arc a_h for all demands of m -th PI does not exceed the capacity reserved for this PI at arc a_h :

$$\sum_{r \in D_m} z_{r,m,h} \leq x_{m,h}; \quad m \in M; \quad h \in A \quad (8.7)$$

We also implemented another objective function aimed at maximizing the total residual (free) capacity at all arcs, as given in Eq. 8.8. This objective is suitable when determining the capacity assignment in a way to increase the residual capacity margin (necessary, e.g., to apply the resilience schemes based on backup paths).

$$\max \varphi(x) = \sum_{h \in A} \left(c_h - \sum_{m \in M} x_{m,h} \right) \quad (8.8)$$

The next model implemented in System IIP is an extension to the LBUO model by additional constraints referring to node resource optimization issues. Therefore, it also includes constraints on node resources (here related to node processing power).

Model 8.2 Extension of the LBUO Model Including Basic Requirements on Node Resource Utilization Optimization Issue (LBNR)

Symbols

The set of symbols is the same as in the LBUO model.

Constants

Compared to the LBUO model, the list of constants is additionally extended by the following:

- $\theta_{m,h}(\rho_{m,h})$ Consumption of node processing power measured per unit capacity for m -th PI defined for outgoing (incoming) arc a_h
- ϕ_n Aggregate processing power at node n

Variables

The list of variables is the same as in the LBUO model and extended by the following:

- $\wp_{m,n} \geq 0$ Amount of resources reserved to process flows from m -th PI at node n (in MFlops)

Objective

It is to minimize the total processing power to deliver the traffic defined by formula (8.9).

$$\min \varphi(x) = \sum_{m \in M} \sum_{n \in N} \wp_{m,n} \quad (8.9)$$

Constraints

The set of constraints includes formulas (8.2)–(8.7) and is additionally extended by constraint (8.10) referring to calculation of node n processing power utilization related to the portion of capacity reserved for each m -th PI and formula (8.11) providing the upper bound on the total processing power available at node n to serve all demands.

$$\wp_{m,n} = \sum_{\substack{h \in \{h: a_h = (n, j) \in A; \\ j = 1, 2, \dots, |N|; j \neq n\}}} \theta_{m,h} x_{m,h} + \sum_{\substack{h \in \{h: a_h = (i, n) \in A; \\ i = 1, 2, \dots, |N|; i \neq n\}}} \rho_{m,h} x_{m,h}; \quad m \in M; \quad n \in N \quad (8.10)$$

$$\sum_{m \in M} \wp_{m,n} \leq \phi_n; \quad n \in N \quad (8.11)$$

The last of the three network resource provisioning models implemented in System IIP includes additional constraints on the maximum allowed transmission delay for delay-sensitive streams. In this model, any potential path is verified

concerning its end-to-end delay, defined as the sum of delays along all network arcs a_h forming the path. Therefore, in this case, any valid solution must consist of paths compliant with upper bounds on end-to-end delay.

Model 8.3 Extension of LBNR Model Including Additional Constraints on End-to-end Delay (LBDC)

Symbols

The set of symbols is the same as in the LBUO model.

Constants

Compared to LBUO and LBNR models, the list of constants is additionally extended by:

- p_h Upper bound on transmission delay along arc a_h
- $p_{m,r}$ Upper bound on end-to-end transmission delay for demand r from m -th Parallel Internet
- G Arbitrarily chosen large value

Variables

The list of variables is the same as in the LBNR model and additionally includes the following:

- $v_{r,m,h}$ Equals 1 if arc a_h is used to forward the traffic referring to demand r of m -th PI and 0 otherwise

Objective

The same as in the LBUO model (i.e., Eq. 8.1).

Constraints

The set of constraints includes formulas (8.2)–(8.7) and (8.10)–(8.11) and is extended by formula (8.12) to guarantee that the end-to-end transmission delay for any demand r from m -th Parallel Internet does not exceed a predefined upper bound, as well as formula (8.13) combined with constant G necessary to bind the respective binary variable $v_{r,m,h}$ with the continuous variable $z_{r,m,h}$.

$$\sum_{h \in A} v_{r,m,h} p_h \leq p_{m,r}; \quad r \in D_m; \quad m \in M \quad (8.12)$$

$$z_{r,m,h} \leq v_{r,m,h} G; \quad r \in D_m; \quad m \in M; \quad h \in A \quad (8.13)$$

All three problems were generally proved to be \mathcal{NP} -complete in [37]. Therefore, for larger problem instances, it is necessary to use one of the suboptimal heuristic approaches, e.g., the one we proposed in [37].

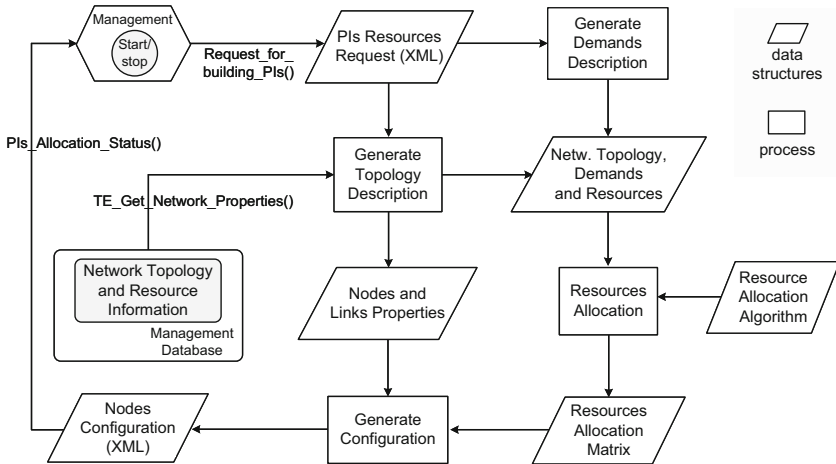


Fig. 8.4 The functional diagram of network resource provisioning module in System IIP architecture from [37]

Owing to the utilization of the implemented network resource provisioning module concerning the core network (i.e., characterized by little fluctuations of the aggregate flows over time), it is reasonable to activate the resource provisioning procedure once every several hours/days. Figure 8.4 presents a functional diagram of the network resource provisioning module in the System IIP architecture.

Three introduced Linear Programming models of network resource provisioning implemented by us in System IIP have been validated for the real large-scale testbed deployed in the IIP project and passed all necessary tests. Similar approaches to determine the optimal solution to the network resource provisioning problem are often applied in the design of resilient network architectures to decide on not only resource provisioning concerning the primary communication paths but also concerning backup routes, as discussed in detail in Sect. 8.3 for the information-centric networking concept (the paradigm of one of PIs addressed in this chapter).

8.3 Fault Tolerance of Content-Oriented Networking

Owing to the remarkable increase in Internet traffic in recent years [1], as well as further predictions of expected exponential increase (mainly attributed to the exchange of various forms of objects, including video, music, and other documents), Future Internet architecture should be characterized by built-in efficient and scalable techniques of content distribution. Indeed, contrary to conventional host-centric communications based on named hosts, the *content-oriented networking (CON)* concept (often referred to as *data-oriented networking* [32, 44] or *information-centric networking (ICN)* [5, 66, 74]) to provide access to *named data objects*

(*NDOs*) [1, 51], focuses on objects of practically any kind that people wish to store and access as the main elements to be addressed. Although the idea itself is not new (see, e.g., solutions of peer-to-peer information exchange from [17, 31]), there is no such built-in mechanism available for the current Internet.

Following [1], an *NDO*—the main abstraction in information-centric networking—does not depend on location, storage method, etc. Therefore, its name is considered an identity regardless of its physical location. Naming an object in information-centric networking is thus as important as issues of naming a host in a conventional scheme. Object names should be unique since they are used for identification independent of their location.

Several copies of an *NDO* stored in the Internet should thus be equivalent. It means that any node that holds a copy of an object should be able to provide it to the requesting node if a node with the original *NDO* is unavailable (for instance, due to node failure or a failure of a transit link/node of a communication path). It is essential to ensure a reliable content distribution in a failure-prone environment, especially with sparse connectivity or high-speed mobility [19].

Considering routing issues, there are several approaches to retrieving information from the source nodes of the content. Among them, it is essential to mention the strategy implemented in the Data-Oriented Network Architecture project [44], where content is published into the network by the sources. Nodes hosting the data have to register themselves at “resolution handlers” that next forward the requests to them from the requesting nodes. Data is further delivered from the source node:

- Via the reverse path of a request
- As information cached at one of the transit nodes (some nodes can use cache memory to act as sources of object copies once they have forwarded the content to the requesting nodes)
- Over a shorter (i.e., a more direct) route

Under *content-centric networking (CCN)*, content is published at original nodes [32]. Therefore, routing is needed to disseminate information about the location of the content around the network.

In general, the considered scheme allowing for serving the content by one of many potential servers, each one storing a copy (also called a replica) of the original object, is referred to as *anycasting* in the literature [38]. This paradigm will be investigated in detail in the later part of this section, where we focus on improving the resilience of information-centric networking and present our approaches from [59, 71, 72] to protect against failures of network elements using alternate paths to such a replicated content.

8.3.1 The Concept of Survivable Anycasting

Anycasting, a one-to-one-of-many transmission technique [47] commonly utilized by a number of services, including Content Delivery Networks (CDNs), Domain

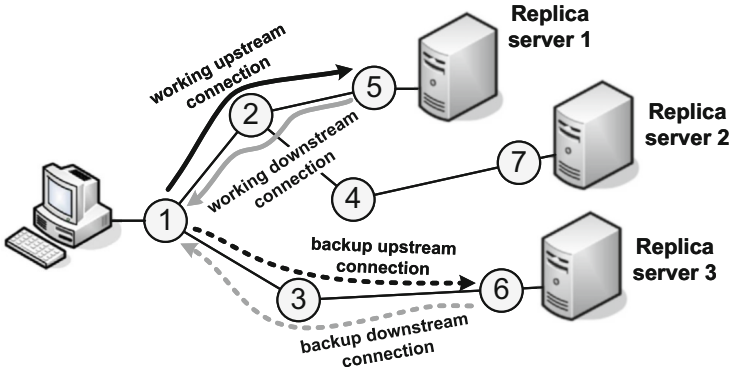


Fig. 8.5 An example of survivable anycast routing with a backup path leading to another replica server

Name System (DNS), peer-to-peer (P2P) systems, or video streaming, due to possibility to retrieve the content from one of many locations, decreases the overall network load and latency, compared to the common unicast (i.e., one-to-one) transmission. Anycasting can also provide survivability of stored information since, unlike in unicasting, in the case of a failure of a node hosting the content, information can be retrieved from another replica server (as, e.g., in Fig. 8.5) [70].

Our proposal from [72] presented here aims at optimizing the routing of anycast and unicast flows with a particular focus on assuring the survivability of the affected traffic. Such a joint optimization scheme is reasonable due to the coexistence of these transmission types in contemporary networks. For instance, the growing popularity of content delivery networking [65, 75] is responsible for 20% share of Internet traffic currently served by the Akamai system [2].

In the case of anycast traffic, to provide survivability against single failures of end nodes, the content has to be stored in parallel at two different replica servers accessible using node-disjoint paths [69]. For unicast traffic, a conventional end-to-end path protection scheme can be employed. The novelty of our approach, compared to other results available in the literature (e.g., [8, 22, 46, 49, 69]), is in the application of a single backup path method aimed at providing 100% protection for both anycast and unicast demands.

In this section, we present an optimization model to protect against single link failures (i.e., establishing link-disjoint paths), as well as failures of replica nodes (by utilization of different primary and backup replicas). The model is related to the physical infrastructure of optical networks, which can be well justified by common utilization of WDM technology in backbone networks.² Therefore, in this section, we consider a directed network $G(N, A)$, where N is a set of nodes, and A is a set

² This approach can be easily adapted for other networking solutions (e.g., for overlay anycasting by replacing the term “optical channel capacity” with the capacity of a virtual link).

of directed arcs. Each arc $a_h \in A$ is characterized by cost ξ_h (referring to the length of arc a_h) and offers Λ unidirectional channels, each of a standard capacity. Replica servers are located at nodes selected in advance in the network planning phase.

All network flows are modeled as non-bifurcated multicommodity flows. In this model, we assume that for each demand r , the requested capacity equals the capacity of a single WDM channel (i.e., $c_r = 1$). In anycast communications, we have upstream and downstream demands (referring to sets D^{US} and D^{DS} , respectively). Each anycast demand r is related to a given client node (being the source s_r /destination t_r node of the upstream/downstream demand, respectively).

Each anycast upstream (downstream) demand $r \in D^{US}(D^{DS})$ has to be associated with the respective downstream (upstream) anycast demand (denoted as $\tau(r)$) referring to the same client node. As shown in Fig. 8.5, both associated anycast demands r and $\tau(r)$ must be related to the same replica node. Since all replica servers located in the network are assumed to provide the same content, working and backup paths can lead to any two of them. The proposed ILP model is defined as follows:

Symbols

N	Set of network nodes
n	Network node
A	Set of arcs representing directed links
h	Arc index
D	Set of demands
$D^{UN} (D^{AN})$	Set of unicast (anycast) demands
$D^{DS} (D^{US})$	Set of anycast downstream (upstream) demands
r	Demand index
$\tau(r)$	Index of a demand associated with demand r

Constants

$s_r(t_r)$	Source (destination) node of r -th demand. For downstream anycast demands, we are given only the destination nodes t_r , while for upstream anycast demands, only source nodes s_r are defined
c_h	Capacity of arc a_h , here given by the number Λ of unidirectional optical channels
ξ_h	Cost (length) of arc a_h
u_n	Equals 1 if node n is a replica node; 0 otherwise
$\chi_{r,n}$	Equals 1 if node n is the closest replica for anycast demand r ; 0 otherwise

Variables

$x_{r,h}$	Equals 1 if arc a_h is used by the working path of r -th demand; 0 otherwise
$y_{r,h}$	Equals 1 if arc a_h is used by the backup path of r -th demand; 0 otherwise
$\kappa_{r,n}$	Equals 1 if a replica server located at node n is selected as a working replica of r -th anycast demand; 0 otherwise

$v_{r,n}$ Equals 1, if a replica server located at node n is selected as a backup replica of r -th anycast demand; 0 otherwise

Objective

It is to minimize the total cost of delivery of flows using working and backup paths given by formula (8.14).

$$\min \varphi(x) = \sum_{r \in D} \sum_{h \in A} \xi_h (x_{r,h} + y_{r,h}) \quad (8.14)$$

Constraints

1. To provide flow conservation rules of working paths of unicast demands:

$$\sum_{\substack{h \in \{h: a_h = (n, j) \in A; \\ j = 1, 2, \dots, |N|; j \neq n\}}} x_{r,h} - \sum_{\substack{h \in \{h: a_h = (i, n) \in A; \\ i = 1, 2, \dots, |N|; i \neq n\}}} x_{r,h} = \begin{cases} 1, & \text{if } n = s_r \\ -1, & \text{if } n = t_r; \quad r \in D^{UN}; \quad n \in N \\ 0, & \text{otherwise} \end{cases} \quad (8.15)$$

2. To provide flow conservation rules of backup paths of unicast demands:

$$\sum_{\substack{h \in \{h: a_h = (n, j) \in A; \\ j = 1, 2, \dots, |N|; j \neq n\}}} y_{r,h} - \sum_{\substack{h \in \{h: a_h = (i, n) \in A; \\ i = 1, 2, \dots, |N|; i \neq n\}}} y_{r,h} = \begin{cases} 1, & \text{if } n = s_r \\ -1, & \text{if } n = t_r; \quad r \in D^{UN}; \quad n \in N \\ 0, & \text{otherwise} \end{cases} \quad (8.16)$$

3. To provide flow conservation rules of working paths of anycast downstream demands:

$$\sum_{\substack{h \in \{h: a_h = (n, j) \in A; \\ j = 1, 2, \dots, |N|; j \neq n\}}} x_{r,h} - \sum_{\substack{h \in \{h: a_h = (i, n) \in A; \\ i = 1, 2, \dots, |N|; i \neq n\}}} x_{r,h} = \begin{cases} -1, & \text{if } n = t_r \\ \kappa_{r,n}, & \text{if } n \neq t_r; \quad r \in D^{DS}; \quad n \in N \end{cases} \quad (8.17)$$

4. To provide flow conservation rules of backup paths of anycast downstream demands:

$$\sum_{\substack{h \in \{h: a_h = (n, j) \in A; \\ j = 1, 2, \dots, |N|; j \neq n\}}} y_{r,h} - \sum_{\substack{h \in \{h: a_h = (i, n) \in A; \\ i = 1, 2, \dots, |N|; i \neq n\}}} y_{r,h} = \begin{cases} -1, & \text{if } n = t_r \\ v_{r,n}, & \text{if } n \neq t_r; \quad r \in D^{DS}; \quad n \in N \end{cases} \quad (8.18)$$

5. To provide flow conservation rules of working paths of anycast upstream demands:

$$\sum_{\substack{h \in \{h: a_h = (n, j) \in A; \\ j = 1, 2, \dots, |N|; j \neq n\}}} x_{r, h} - \sum_{\substack{h \in \{h: a_h = (i, n) \in A; \\ i = 1, 2, \dots, |N|; i \neq n\}}} x_{r, h} = \begin{cases} 1, & \text{if } n = s_r \\ -\kappa_{r, n}, & \text{if } n \neq s_r; \quad r \in D^{US}; \quad n \in N \end{cases} \quad (8.19)$$

6. To provide flow conservation rules of backup paths of anycast upstream demands:

$$\sum_{\substack{h \in \{h: a_h = (n, j) \in A; \\ j = 1, 2, \dots, |N|; j \neq n\}}} y_{r, h} - \sum_{\substack{h \in \{h: a_h = (i, n) \in A; \\ i = 1, 2, \dots, |N|; i \neq n\}}} y_{r, h} = \begin{cases} 1, & \text{if } n = s_r \\ -v_{r, n}, & \text{if } n \neq s_r; \quad r \in D^{US}; \quad n \in N \end{cases} \quad (8.20)$$

7. To provide a proper selection of replica nodes:

$$\kappa_{r, n} \leq u_n; \quad r \in D^{AN}; \quad n \in N \quad (8.21)$$

$$v_{r, n} \leq u_n; \quad r \in D^{AN}; \quad n \in N \quad (8.22)$$

8. To guarantee that the associated upstream and downstream anycast demands use the same corresponding replica node for working paths:

$$\kappa_{r, n} = \kappa_{\tau(r), n}; \quad r \in D^{DS}; \quad n \in N \quad (8.23)$$

9. To guarantee that the associated upstream and downstream anycast demands use the same corresponding replica node for backup paths:

$$v_{r, n} = v_{\tau(r), n}; \quad r \in D^{DS}; \quad n \in N \quad (8.24)$$

10. To provide that exactly one node is selected as the working replica node for each anycast demand:

$$\sum_{n \in N} \kappa_{r, n} = 1; \quad r \in D^{AN} \quad (8.25)$$

11. To assure that exactly one node is selected as the backup replica node for each anycast demand:

$$\sum_{n \in N} v_{r, n} = 1; \quad r \in D^{AN} \quad (8.26)$$

12. On finite arc capacity:

$$\sum_{r \in D} (x_{r,h} + y_{r,h}) \leq c_h; \quad h \in A \quad (8.27)$$

13. To provide link disjointness of working and backup paths of anycast demands:

$$(x_{r,h} + y_{r,h}) \leq 1; \quad r \in D; \quad h \in A \quad (8.28)$$

14. To guarantee link disjointness of the respective working path and backup path of the associated anycast demand:

$$(x_{\tau(r),h} + y_{\tau(r),h}) \leq 1; \quad r \in D^{AN}; \quad h \in A \quad (8.29)$$

The objective is to minimize the overall cost of the flow (formula (8.14)) subject to constraints (8.15)–(8.29). In the model given by formulas (8.14)–(8.29), there is no constraint referring to the physical separation of working and backup replica servers (i.e., they may be hosted at either the same or different nodes). Therefore, the model (8.14)–(8.29) is called Any Replica (AR) here.

Our investigations are also extended by:

- An additional constraint (8.30) to provide disjointness of working and backup replica servers (forming the Disjoint Replica (DR) model defined by formulas (8.14)–(8.30))
- Constraint (8.31) to assure that for each anycast demand, working and backup replica servers are hosted by the same node (Common Replica (CR) model given by formulas (8.14)–(8.29) and (8.31))
- Constraint (8.32) to assure that working and backup replica servers are located in the nearest vicinity for each anycast demand—forming the Nearest Replica (NR) model [42] by formulas (8.14)–(8.29) and (8.32)

$$\sum_{n \in N} (\kappa_{r,n} + v_{r,n}) \leq 1; \quad r \in D^{AN} \quad (8.30)$$

$$\kappa_{r,n} = v_{r,n}; \quad r \in D^{AN}; \quad n \in N \quad (8.31)$$

$$\kappa_{r,n} = v_{r,n} = \chi_{r,n}; \quad r \in D^{AN}; \quad n \in N \quad (8.32)$$

Simulation Results and Conclusions

Verification of characteristics of four introduced models focusing on evaluation of the total network cost (defined as given in formula (8.14)), and values of computational time, was performed for four example networks, namely, the NSF Network, COST 239 Network, Italian Network, and US Long-Distance Network from Fig. 8.6. All links were assumed to have $\Lambda = 160$ channels of equal capacity.

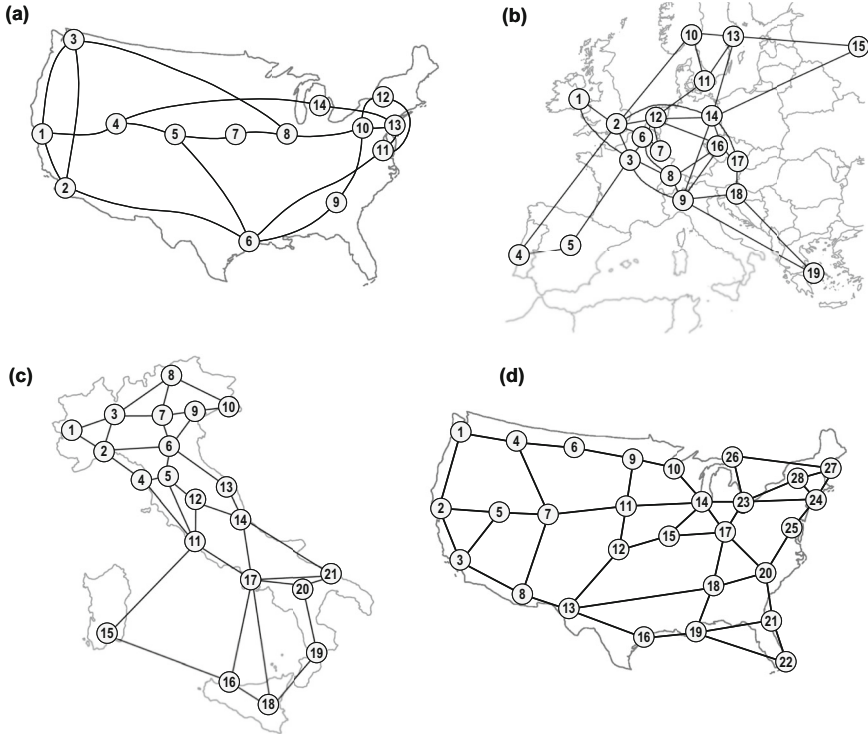


Fig. 8.6 Network topologies used in the analysis: NSF Network (a), COST 239 Network (b), Italian Network (c), and US Long-Distance Network (d)

Table 8.1 Locations of replica servers (node indices)

Network	2 replicas	4 replicas
NSF	6, 10	4, 5, 6, 10
COST 239	2, 14	2, 3, 9, 14
Italian	6, 17	6, 7, 11, 17
US Long-Distance	14, 17	7, 14, 17, 23

Nodes had a full wavelength conversion capability (i.e., at each transit node, flows arriving at any wavelength λ_i of the incoming link could be switched onto any wavelength λ_o of the outgoing link).

Two scenarios referring to the number of replica servers were investigated, i.e., 2 and 4, as shown in Table 8.1, with replica servers located at nodes of a relatively high degree (i.e., defined as the number of neighboring nodes).

The set of anycast demands (D^{AN}) contained all network nodes. The set of unicast demands (D^{UN}) included the respective number of randomly chosen pairs of nodes (with node indices following the uniform distribution) such that the anycast ratio (i.e., the number of anycast demands $|D^{AN}|$ divided by the total number of demands $|D|$) was equal to 30%.

In each simulation determined by a replica model, number of replica servers, and network topology, computations were performed for 50 different sets of demands D generated randomly (following the uniform distribution of node indices). An analysis of multiple scenarios of network load, replica servers count, and other extensions of our ILP model is given in [72].

Table 8.2 presents the average execution time for each analyzed topology and replica model. As shown in Table 8.2, all four models are characterized by comparable values of the average execution time. The only exception is the CR model, for which the average execution time is about two times greater than for the other models. This is due to additional constraints (8.31), including working and backup replica variables.

Figures 8.7 and 8.8 present the average network costs calculated based on formula (8.14), as well as their relation with the number of available replica servers. Independent of the replica model, increasing the number of replica servers decreases the overall cost of a network (as a consequence of the observed decrease in the average total length of established paths). Indeed, when increasing the number of available replica servers, the average minimal distance between replica servers and client nodes becomes smaller.

Regarding the characteristics of analyzed models, the AR approach outperforms the other ones. This is due to its flexibility (i.e., it does not impose additional constraints on replica servers selection). The performance of the other models depends on network characteristics and the number of available replica servers.

Table 8.2 Average execution time

Network	DR [s]	CR [s]	NR [s]	AR [s]
NSF	0.41	2.80	0.43	0.43
COST 239	1.38	2.53	1.44	1.41
Italian	1.69	3.98	1.68	1.67
US Long-Distance	3.34	5.55	3.37	3.40

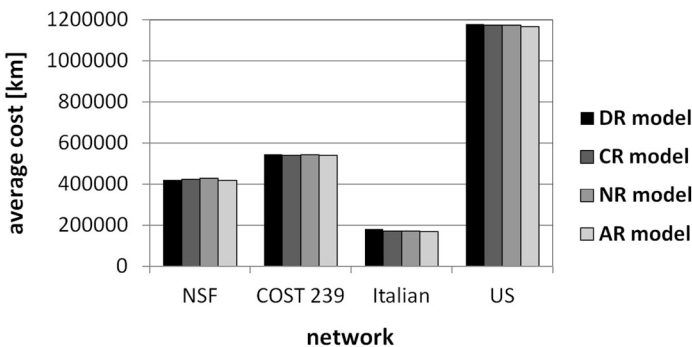


Fig. 8.7 Average network cost for two replica servers

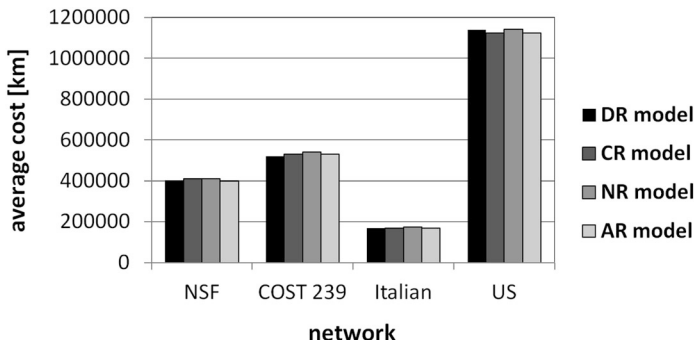


Fig. 8.8 Average network cost for four replica servers

As discussed in Chap. 4 of this book, providing preplanned protection against failures by alternate paths increases the cost of the original solution (i.e., the one without backup paths) by over 100%, since backup paths are commonly longer than the corresponding working paths. Therefore, to reduce the overall cost of a solution, the concept of survivable anycast and unicast routing will be extended in the next section by sharing the backup path capacities.

8.3.2 Shared Protection for Survivable Anycasting

As discussed in Chap. 2, to decrease the ratio of network redundancy necessary to provide 100% protection of flows after failures of nodes (or links), one may apply the concept of sharing the backup paths resources (i.e., link capacities) under the condition that the respective working paths being protected are mutually node-/link-disjoint [4, 41]. This section presents our proposal from [71] of sharing the backup path resources for routing anycast and unicast demands with protection against a single link failure.

So far, the concept of backup path sharing has been investigated mainly for the case of unicast traffic protection [39–41, 58]. Considering backup path resource sharing for survivable anycast routing (as illustrated in Fig. 8.9), recent models to find the optimal solution available in the literature have been formulated using only the link-path formulation (i.e., with a limited number of predefined candidate backup paths) [33]. This, in fact, leads to suboptimal results since, in link-path formulation, not all possible backup paths are analyzed.

In this section, we introduce the Integer Linear Programming formulation of the backup path sharing problem defined using the node-link notation, enabling the investigation of all possible backup paths and, consequently, allowing us to reach optimal results. This model, being an extension of the respective one from Sect. 8.3.1, is defined as follows.

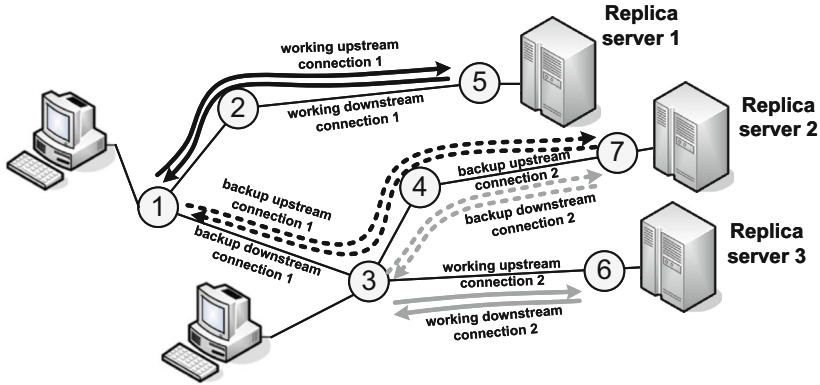


Fig. 8.9 Example of survivable anycast routing with different backup replica servers. Sharing the backup path capacities may be performed at links (3, 4) and (4, 7)

Symbols

The set of symbols is the same as in Sect. 8.3.1 and is extended by the following:

c_r Volume (capacity) of demand r

Variables

The set of variables is the same as in Sect. 8.3.1 and is extended by the following:

$b_{r,h,g}$ Is equal to 1 if after a failure of arc a_g , the channel of arc a_h is used by a backup path of r -th demand, and 0 otherwise

$b_{h,g}$ Spare capacity required at arc a_h in the case of link a_g failure (integer value)

b_h Aggregate spare capacity to be reserved for backup paths at arc a_h (integer value) to protect against a failure of each single link

Objective

It is to minimize the total cost of delivery of flows using working and backup paths given by formula (8.33).

$$\min \varphi(x) = \sum_{r \in D} \sum_{h \in A} \xi_h c_r x_{r,h} + \sum_{h \in A} \xi_h b_h \tag{8.33}$$

Constraints

1. To provide flow conservation rules of working and backup paths of unicast demands; flow conservation rules for downstream and upstream anycast demands: formulas (8.15)–(8.20)
2. To provide a proper selection of replica nodes: formulas (8.21)–(8.22)

3. To assure that the associated upstream and downstream demands use the same corresponding replica node for working and backup paths: formulas (8.23)–(8.24)
4. To guarantee that exactly one node is selected as a working and backup replica node for each anycast demand: formulas (8.25)–(8.26)
5. On finite arc capacity:

$$\sum_{r \in D} c_r x_{r,h} + b_h \leq \Lambda; \quad h \in A \quad (8.34)$$

6. To provide link disjointness of working and backup paths: formulas (8.27)–(8.28)
7. To obtain shared protection concerning the considered backup paths:

$$x_{r,g} + y_{r,h} \leq 1 + b_{r,h,g}; \quad r \in D; h \in A; g \in A; g \neq h \quad (8.35)$$

$$2b_{r,h,g} \leq x_{r,g} + y_{r,h} \quad r \in D; h \in A; g \in A; g \neq h \quad (8.36)$$

8. To provide bounds on arc spare capacity:

$$b_{h,g} = \sum_{r \in D} c_r b_{r,h,g}; \quad h \in A; g \in A; g \neq h \quad (8.37)$$

$$b_{h,g} \leq b_h; \quad h \in A; g \in A; g \neq h \quad (8.38)$$

9. To assure location of working and backup replica servers at the nearest nodes: formula (8.32)

If we replace formula (8.38) with the following formula (8.39), we obtain the model without shared protection, since b_h is then defined simply as the sum of backup capacities over all link failures.

$$b_h = \sum_{g \in A} b_{h,g} \quad (8.39)$$

To summarize, the above formulas can be used to obtain the four following models investigated in detail in the later part of this section:

- SBPP-AR: Any Replica model; shared protection: formulas (8.33), (8.15)–(8.20), (8.23)–(8.28), (8.34)–(8.38)
- SBPP-NR: Nearest Replica model; shared protection: formulas (8.33), (8.15)–(8.20), (8.23)–(8.28), (8.32), (8.34)–(8.38)
- noSBPP-AR: Any Replica model; dedicated protection: formulas (8.33), (8.15)–(8.20), (8.23)–(8.28), (8.34)–(8.37), (8.39)
- noSBPP-NR: Nearest Replica model; dedicated protection: formulas (8.33), (8.15)–(8.20), (8.23)–(8.28), (8.32), (8.34)–(8.37), (8.39)

Simulation Results and Conclusions

Numerical experiments aimed to evaluate the efficiency of the introduced shared protection schemes in terms of (1) the total cost of a solution, (2) the length and hop count of established paths, all as a function of the anycast ratio (defined as the proportion of anycast traffic to the total traffic—i.e., anycast and unicast), (3) the number of replica servers available in the network (2 or 3)—as given in Table 8.3, and (4) two analyzed scenarios (AR and NR) of replica server locations.

Considering the anycast ratio, we investigated the values from the set (10%, 20%, . . . , 80%). Twenty-four different demand sets (comprising three demand sets per each anycast ratio value) were generated randomly (using the uniform distribution function of indices of demand nodes). The numbers of anycast and unicast demands per each demand set were in ranges of 8–28 and 7–44, respectively. To obtain a given value of anycast ratio, demand volumes c_r were selected from the range 1–9. Two cases of replica servers count (2 and 3, respectively) and four analyzed variants of our ILP model in total gave 192 different experiments, all performed for the analyzed NSF network from Fig. 8.6a.

Experiments were also prepared to evaluate the performance of shared backup capacity models compared to schemes without backup capacity sharing. Therefore, the first set of results, presented in Fig. 8.10, refers to the average overall cost of solutions (based on formula (8.33)) in terms of ratios $cost^{SBPP}/cost^{noSBPP}$ as a function of the anycast ratio parameter. The average value of this coefficient (obtained for all experiments) was 0.64, meaning that shared backup path approaches outperformed the respective “no sharing” ones by 36%. As shown in Fig. 8.10, the difference between the analyzed approaches decreases with the increase of the anycast ratio parameter since, under anycasting, one of the end nodes of demand is also related to one of the replica servers located at a limited number of

Table 8.3 Locations of replica servers (node indices)

2 replica servers	4 replica servers
6, 10	4, 6, 10

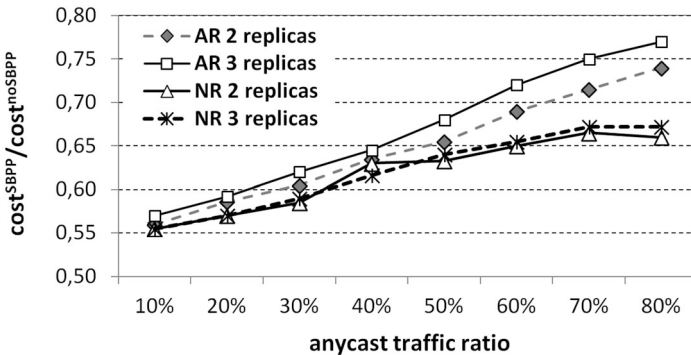


Fig. 8.10 Average cost ratios between SBPP and noSBPP solutions

Table 8.4 Average ratios between SBPP and noSBPP schemes

Number of replica servers	2	3	2	3
Replica scenario	AR	AR	NR	NR
Cost	0.65	0.67	0.62	0.62
Capacity utilization	0.60	0.61	0.56	0.57
Anycast working path length	1.01	1.06	1.01	1.03
Anycast backup path length	2.00	2.09	1.79	1.91
Anycast working path hops	1.00	1.00	1.01	1.02
Anycast backup path hops	1.54	1.60	1.43	1.56
Unicast working path length	1.01	1.01	1.01	1.05
Unicast backup path length	1.71	1.68	1.78	1.86
Unicast working path hops	1.01	1.01	1.00	1.03
Unicast backup path hops	1.49	1.43	1.53	1.55

network nodes. This, in turn, limits the possibility of backup path sharing (following the general backup capacity sharing rule).

As shown in Fig. 8.10, increasing the number of replica servers (here from 2 to 3) also reduces the gap between SBPP and noSBPP models, since with the increase of the number of replica servers, working paths become shorter (due to the physical location of replica servers closer to the client nodes). Therefore, with the increase in the number of replica servers, the average path hop count decreases, which implies fewer possibilities of backup capacity sharing.

Table 8.4 presents the average ratios between SBPP and noSBPP models for all analyzed parameters. In general, there is no visible impact of the scenario of replica server location on the presented ratios independent of analyzed metrics. Considering the cost metric, the Any Replica (AR) model is characterized by lower values of the cost difference (expressed by larger values of the SBPP/noSBPP ratio) since AR, being more flexible than the Nearest Replica (NR) scheme, can benefit from switching the traffic to another replica server after the failure (not possible for the NR model implying location of working and backup replicas of demand at the same closest network node).

Characteristics of the capacity utilization metric are similar, i.e., with the increase of the anycast traffic ratio, and the number of replicas, the difference between SBPP and noSBPP scenarios (42%, on average), becomes less visible.

The most crucial result refers to the average length of backup paths, which is about 70–100% greater for SBPP schemes compared to noSBPP approaches for both anycast and unicast demands. This is due to the backup path cost included in the objective function (Eq. 8.33) reflecting only the extra capacity that has to be reserved for backup paths (i.e., the fraction of backup capacity without the possibility of sharing). Therefore, links with sharable backup capacity are preferred in backup path computations. Backup paths may thus traverse many links of “zero” cost, which increases their hop count.

As shown in Fig. 8.11, with the increase of the anycast traffic ratio, the 3 replica/2 replica ratio considering cost and capacity parameters decreases, implying

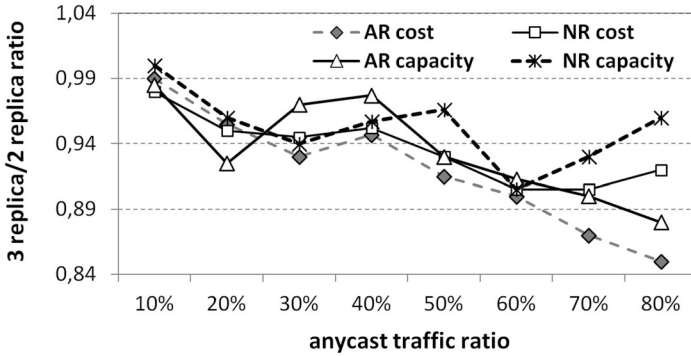


Fig. 8.11 Average ratios of results between 3 and 2 replica servers

the growth of the difference of cost and capacity parameters. This is a natural consequence of adding a new replica server, leading to more efficient results in reducing the average path length (observed with the increase of the anycast traffic ratio). The obtained results confirm the remarkable capacity efficiency of our shared protection scheme at the price of the increased length of backup paths.

8.3.3 Protection of Information-Centric Communications Against Intentional Failures

The majority of proposals available in the literature are related to protection against random failures, being the implication of hardware faults, software defects, or simply human errors, all typically characterized by uniform distribution function of failure probabilities (i.e., failure probabilities independent of network element characteristics). Only a few papers address the issue of protection against failures resulting from malicious activities, referred to as *attacks*, typically affecting the most important network elements (i.e., nodes/links of relatively high degree/capacity switching/storing large amounts of data). The problem is of utmost importance since attacking such elements frequently causes severe losses (which is actually the main aim of attackers).

Such differentiation of severity of attack outcomes can be observed mainly for networks of irregular topology (obtained due to an uncontrolled network growth), for which the node degree distribution does not comply with the uniform law. Following the Barabási and Albert rule of *preferential attachment* of new nodes from [10], when adding a node to the network, it is more probable to link it with an existing one of high rather than low degree, as given in formula (8.40). In case of such an uncontrolled growth, network topologies commonly gradually evolve toward irregular ones (as illustrated in Fig. 8.12) with asymptotic power law degree distribution of node degrees k given by formula (8.41) [10]. Examples include, e.g., topology of the Internet with $\gamma = 2.22$ [76].

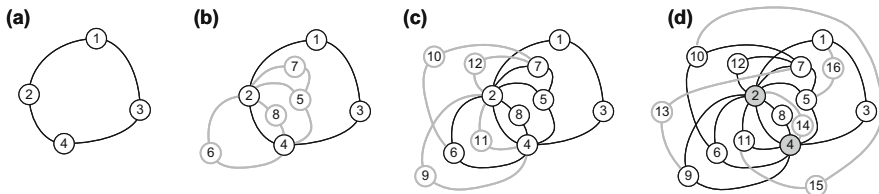


Fig. 8.12 Example evolution of the network topology from (a) illustrated in steps (b)–(d) following the preferential attachment rule

$$\Pi(n) = \frac{\text{deg}(n)}{\sum_j \text{deg}(j)} \tag{8.40}$$

$$P(k) \sim k^{-\lambda} \tag{8.41}$$

It is important to notice that under the conventional shortest path routing, many shortest paths traverse such high-degree nodes (also called *central nodes*) and are at high risk of being affected by an attacker. Therefore, shortest path routing is not a proper solution for networks of dynamically evolving topologies. This is especially true for the current Internet, which is owned by multiple providers without any common policy on topology evolution. It is thus crucial to provide Future Internet with routing mechanisms preventing communication paths from attacks.

This section describes our approach from [59] called “resistant-to-attacks” (RA), designed to protect anycast and unicast flows against malicious activities targeted at network nodes. It uses a path protection scheme to ensure the protection of each working path by a dedicated backup path against a single node failure. To reduce the impact of attacks, in our approach:

- Working paths are established using a dedicated metric of link cost (different than the conventional metric of distance applied by us in backup path computations only) to make them omit nodes of high degree
- Replica servers are located at low-degree nodes to reduce the losses resulting from attacks.

The vulnerability of communication paths to attack-based disruptions changes as the network topology is subject to evolution over time. Therefore, it is crucial to introduce a routing scheme that dynamically adjusts its properties in response to network topology changes. To address this objective, in working path computations, we propose to use the metric of link costs based on *betweenness centrality* (BC) coefficient [35] defined for any node n as given in formula (8.42), providing a proper estimation of a node centrality ratio, and thus being an essential indicator of node vulnerability to attacks.

$$BC(n) = \sum_{p \neq q} \frac{sp_n(p, q)}{sp(p, q)} \quad (8.42)$$

where

- $sp_n(p, q)$ is the number of the shortest paths between nodes p and q (of the same minimal length) traversing node n ;
- $sp(p, q)$ is the number of the shortest paths between nodes p and q (of the same minimal length).

In particular, we define the cost ξ_h of arc a_h in working path computations as the average value of the normalized betweenness centrality parameter (BC^*) of nodes i and j incident to arc a_h , as given in formula (8.43). Since the cost of any link incident to a high degree node should be high as well, working paths calculated based on costs (8.43) are thus expected not to traverse such central nodes (as, e.g., nodes 6, 11, and 17 in Fig. 8.13) and, as a result, be less vulnerable to attack-based disruptions.

$$\xi_h = \xi_{i,j} = \frac{BC^*(i) + BC^*(j)}{2} \quad (8.43)$$

where

$$BC^*(n) = \frac{BC(n)}{\max_i BC(i)} \quad (8.44)$$

For the purpose of backup path computations, the cost ζ_h of any network arc a_h is defined here by formula (8.45) as the normalized length of this arc.

$$\zeta_h = \frac{s_h}{\max_i s_i} \quad (8.45)$$

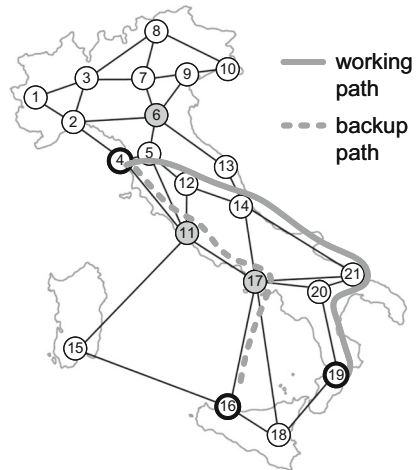
Backup paths are thus established as the shortest ones. Although they are allowed to transit high-degree nodes (as shown in Fig. 8.13), they are used in relatively short periods (for a temporary recovery until the time of manual repair of failed elements).

Similar to Sects. 8.3.1–8.3.2, under anycast routing, working and backup paths may lead to different replica servers to protect against a failure of a replica node (Fig. 8.13).

The ILP model necessary to find the solution to our optimization problem is the same as the Disjoint Replica model from Sect. 8.3.1 defined by formulas (8.14)–(8.30) with the only one exception for the objective function (8.14) here replaced with formula (8.46).

$$\min \varphi(x) = \sum_{r \in D} \sum_{h \in A} \xi_h x_{r,h} + \sum_{r \in D} \sum_{h \in A} \zeta_h y_{r,h} \quad (8.46)$$

Fig. 8.13 Example anycast routing following the proposed approach; Italian network from Fig. 8.6c



However, the considered problem defined by formulas (8.15)–(8.30) and (8.46) is \mathcal{NP} -complete due to \mathcal{NP} -completeness of a simpler task to find $|D|$ working paths only (i.e., without protection) in capacity-constrained networks [50]. Therefore, for larger problem instances, it is necessary to use a heuristic approach to obtain the suboptimal results in a reasonable time. As stated in [57], in the case of multi-cost networks (i.e., when for any link, different link costs are assigned to working and backup path links—as considered in this section), the problem is \mathcal{NP} -complete even for a single demand.

The heuristic scheme from Fig. 8.14, proposed for the general case of establishing the set of k end-to-end node-disjoint paths for a given demand, is similar to the Active Path First (APF) approach [57]. After initialization of Steps 1–3 for each demand, it first tries to calculate the working path using any algorithm of the shortest path computation (e.g., Dijkstra’s from [21]). However, in backup path computations, contrary to the APF scheme, in our approach, to provide nodal disjointness of transmission paths, the costs of the respective *forbidden arcs* traversed by the working path are increased by a large value (instead of being set to infinity). This update is to prevent from entering into the *trap problem* (i.e., the case when the algorithm fails to establish the next disjoint path of a demand, even though it would be feasible for a given topology).

In particular, in the case of establishing k end-to-end node-disjoint paths, before finding the next disjoint path j , for each previously calculated path η_i , the cost of any forbidden arc is first increased by the total cost of path η_i calculated based on the matrix of backup link costs c^j (Step 4). However, after finding the next path (η_j) of a demand in Step 5 and detecting that more than one of the already calculated paths of a demand traverse a given arc a_n , the cost of such a *conflicting arc* is permanently increased by the total cost of path η_j in all matrices c^i (calculated based on arc costs from c^i), and the execution starts from the beginning (Step 6).

INPUT

- A demand (with index r) to determine the set of k end-to-end node-disjoint paths (each unicast demand is determined by a pair of nodes (s_r, t_r) , while each anycast demand is given by a client node s_r to be connected to working and backup replica servers located at different nodes)
- Matrices c^1, c^2, \dots, c^k of arc costs $\zeta_h^1, \zeta_h^2, \dots, \zeta_h^k$ (defined for computations of consecutive disjoint paths of r -th demand)
- The upper bound it_upper on the number of allowed conflicts

OUTPUT The set $\{\eta_1, \eta_2, \dots, \eta_k\}$ of k end-to-end node-disjoint paths

VARIABLES c^{mp} auxiliary matrix of arc costs ζ_h^{mp}
 j index of the current path
 ic conflict counter

- Step 1 Set $ic := 1$.
 Step 2 Set $j := 1$.
 Step 3 For each network arc a_h , set $\zeta_h^{mp} := \zeta_h^j$.
 Step 4 For each path η_i from the set of previously found $j-1$ paths of a demand and for each arc a_h , if a_h is a *forbidden arc** of path η_i , then increase the arc cost ζ_h^{mp} by the total cost ζ^i of η_i in c^j .
 Step 5 Find path η_j using the Dijkstra's algorithm and the costs matrix c^{mp} .
 Step 6 If η_j is not disjoint with the previously found $j-1$ paths of r -th demand then:
 Step 6.1 Increase the costs ζ_h^l of each *conflicting arc*** a_h of η_j by the total cost ζ^j of η_j in all matrices c^l . After that, delete the found paths.
 Step 6.2 Set $ic := ic + 1$.
 Step 6.3 if $ic > it_upper$ then
 terminate and reject the demand,
 else go to Step 2.
 else increment j .
 Step 7 If $j > k$ then terminate and return the found set of paths
 else go to Step 3.
-

* In case of required nodal disjointness of the set of k end-to-end paths of a demand, a forbidden arc of η_i is an arc that is incident to any transit node of η_i

** In case of required nodal disjointness of the set of k end-to-end paths of a demand, arc a_h is a conflicting arc of a given path η_j , if it is incident to any common transit node of η_j also used by any other of previous $j-1$ paths

Fig. 8.14 Heuristic approach to find the set of k end-to-end node-disjoint paths

After several possible conflicts, the method is expected to terminate successfully (as shown later in this section). Our scheme's time complexity depends on the base approach of path computations. If Dijkstra's algorithm from [21] is utilized for this purpose, the overall complexity is bounded from above by $O(|N|^2)$, where $|N|$ is the number of network nodes.

This scheme is used here to find $k = 2$ end-to-end node-disjoint paths.

Simulation Results and Conclusions

Characteristics of the proposed RA approach referring to link capacity utilization ratio, length of working and backup paths, total number of connections broken due to attacks, and the time of connection restoration were evaluated using simulations and compared with the reference results of the common approach to establish working and backup paths using the metric of distance (here called "non-resistant-to-attacks"—NA approach).

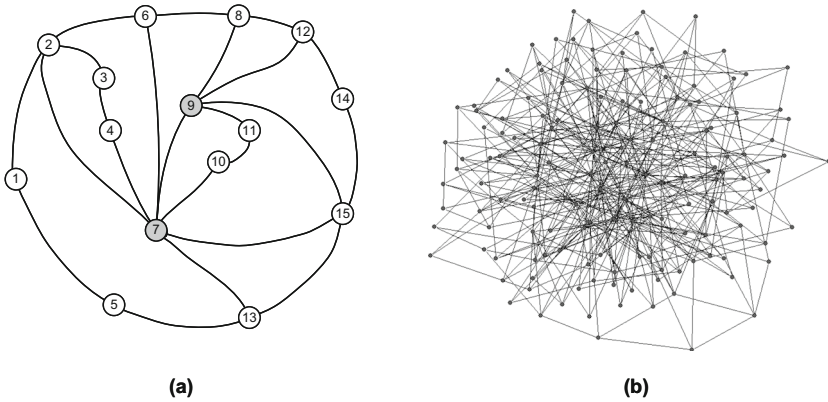


Fig. 8.15 Network topologies used in simulations: ASF Network (a) and BA-150 Network (b)

The time of connection restoration was calculated based on [50]. Experiments were performed using CPLEX 11.0 solver (to obtain the ILP-based optimal results), as well as the heuristic method from Fig. 8.14 for topologies of two irregular networks shown in Fig. 8.15 (achieved using the Barabási-Albert approach of topology generation [10]). Concerning anycast and unicast demands:

- Demanded capacity was assumed to be unitary (equal to the channel capacity).
- 100% of the requested capacity was required to be available for each demand after failures of single nodes.
- Working paths were protected by dedicated backup paths (i.e., without sharing link capacities reserved for backup paths).

Three scenarios of network load were investigated. In each case, the analyzed sets of demands D^{AN} comprised all network nodes. However, concerning unicast demands, the analyzed sizes of demand sets were adjusted in a way to receive three ratios of anycast demands ($|D^{AN}|/|D|$) equal to 10%, 20%, and 30%. Any pair of demand end nodes was always chosen randomly using the uniform distribution function of node indices. Considering the number of replica servers available in the network, we investigated three cases of 2, 3, and 4 replica servers hosted by nodes of the highest (common NA model) and the lowest (our RA model) degree, respectively.

A single simulation comprised 50 different sets of demands for a given network topology and the number of available replica servers. The probability of node failures was proportional to the values of the normalized betweenness centrality coefficient defined for network nodes by Eq. 8.44.

One of the objectives of the simulations was to evaluate the efficiency of our heuristic method in comparison with the results of ILP modeling. This analysis is presented in Fig. 8.16 for ASF network from Fig. 8.15a (with assumed $\Lambda = 40$ channels available at each network link) in terms of the total link capacity per

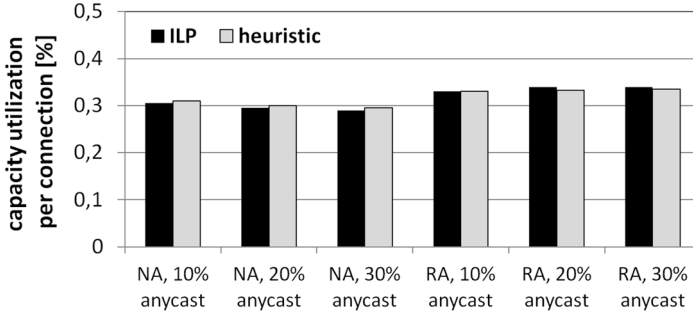


Fig. 8.16 Ratios of total link capacity utilization per connection for ASF network from Fig. 8.15a achieved for different network loads (number of replica servers: 2)

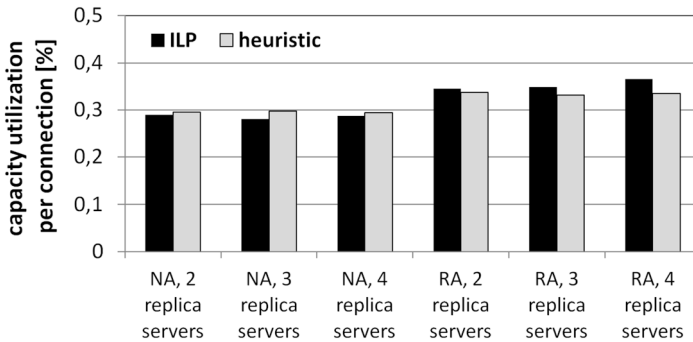


Fig. 8.17 Ratios of total link capacity utilization per connection for ASF network from Fig. 8.15a achieved for different numbers of replica servers (anycast ratio: 30%)

connection necessary to serve the demands as a function of the network load (Fig. 8.16) and the number of replica servers (Fig. 8.17).

The results show that the amount of capacity necessary to serve the demands (per connection) for the heuristic approach was similar to the optimal ILP solution. Our technique sometimes required even less capacity (up to 2.49% less). However, this was an implication of the inconsistency of the proposed formula (3.46) with the hop count metric. In general, our RA scheme required about 10% more capacity than the reference NA algorithm.

The next set of experiments was aimed at evaluating characteristics of the proposed approach related to working and backup path length, the total number of connections broken due to attacks, as well as the average time of connection restoration. Due to the size of the investigated network (BA-150 network from Fig. 8.15b with three replica servers and $\Lambda = 160$ channels available at each link), evaluation was feasible for the heuristic approach only.

For our RA approach, the average length of working paths was up to 2.26 times greater than the common NA scheme (because in the RA scheme, working paths

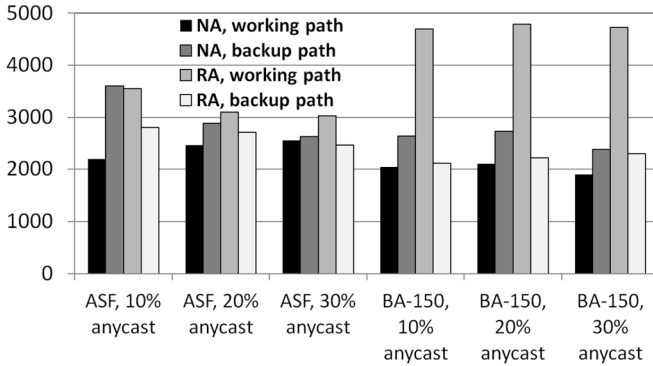


Fig. 8.18 Average length of paths

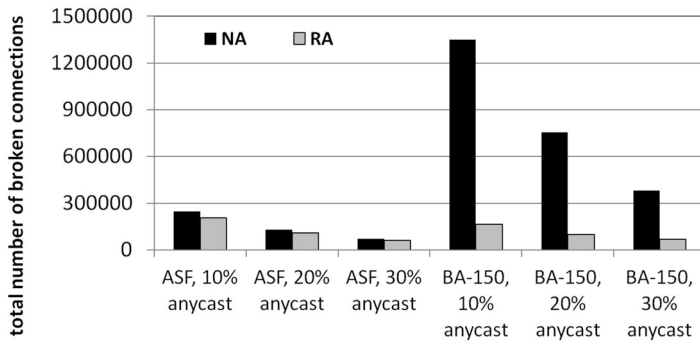


Fig. 8.19 Total number of broken connections

tried to omit high-degree nodes). On the contrary, RA backup paths were about 25% shorter than the respective NA ones (Fig. 8.18).

Since RA working paths were established in a way to omit nodes of high degree, characteristics referring to the number of connections broken due to attacks from Fig. 8.19 show a significant advantage of our scheme (i.e., a 7.47-fold advantage), compared to the reference NA approach. Finally, the achieved average values of service restoration time (which, due to the three-way handshake procedure, commonly depend on lengths of working and backup paths [50]) were similar for both approaches (see Fig. 8.20).

To conclude, the proposed approach to establishing working paths in a way to omit nodes of a high degree results in a remarkable decrease in the number of connections affected after attacks at a price of only an insignificant increase in the length of working paths. The dynamic properties of our scheme make it a suitable solution at any stage of network evolution.

A detailed analysis of our approach characteristics, including, e.g., presentation of 95% confidence intervals for the analyzed parameters, is available in [59].

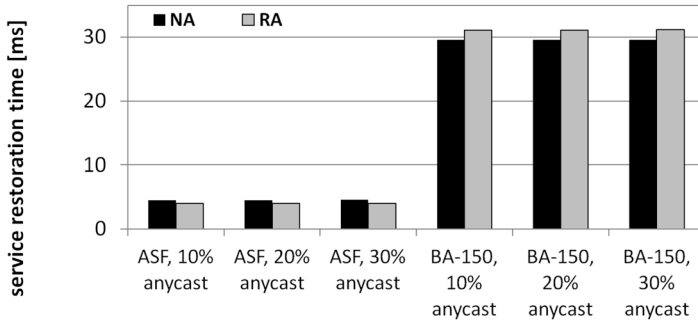


Fig. 8.20 Average service restoration time

8.4 Summary

The diversity of Future Internet desired functionalities, routing paradigms, and challenges threatening the normal operation of any global network altogether make the resilience of FI communications a complex issue. Considered by many to be an important part of a critical infrastructure expected to offer uninterrupted service anytime and anywhere, Future Internet needs to be provided with efficient solutions to assure service continuity under both random and intentional failures.

To address this issue, in this chapter, we first presented the efficient solutions to the routing and network resource provisioning problem deployed by us in one of European research projects on Future Internet architecture, called Future Internet Engineering. Next, we focused on the resilience of content-oriented networking (being an important paradigm for the Future Internet) and introduced three new concepts of survivable routing of unicast, and anycast flows for (1) dedicated and (2) shared protection under random failures of nodes/links and (3) dedicated protection of flows under attack-based disruptions.

Obtained results confirmed the efficiency of our techniques in assuring the uninterrupted routing of FI demands in differentiated scenarios, including dedicated protection (Sect. 8.3.1), shared protection (Sect. 8.3.2 with the achieved 36% reduction of redundancy ratio, compared to the case of dedicated protection) in random failure scenarios, and a significant improvement in terms of reduction of the number of connections broken due to attacks (characterized by a remarkable 7.47-fold advantage over the conventional routing scheme, as shown in Sect. 8.3.3).

References

1. Ahlgren, B., Dannewitz, Ch., Imbrenda, C., Kutcher, D., Ohlman, B.: A survey of information-centric networking. *IEEE Commun. Mag.* **50**(7), 26–36 (2012)
2. Akamai project: <http://www.akamai.com>. Accessed on 08 Mar 2015

3. Akari architecture design project: http://www.nict.go.jp/en/photonic_nw/archi/akari/akari_top_e.html. Accessed on 08 Mar 2015
4. Ali, M.: Shareability in optical networks: beyond bandwidth optimization. *IEEE Opt. Commun.* **42**(2), s11–s15 (2004)
5. Al-Naday, M.F., Reed, M.J., Trossen, D., Yang, K.: Information resilience: source recovery in an information-centric network. *IEEE Network* **28**(3), 36–42 (2014)
6. Álvarez, F., Cleary, F., Daras, P., Domingue, J., Galis, A., Garcia, A., Gavras, A., Karnourkos, S., Krco, S., Li, M.-S., Lotz, V., Mueller, H., Salvadori, E., Sassen, A.-M., Schaffers, H., Stiller, B., Tselentis, G., Turkama, P., Zahariadis, T. (Eds.): *The Future Internet – Future Internet Assembly (FIA 2012): From Promises to Reality*. Aalborg, 9–11 May, 2012. *Lecture Notes in Computer Science*, vol. 7281. Springer, Berlin (2012)
7. Anderson, T., Peterson, L., Shenker, S., Turner, J.: Overcoming the Internet impasse through virtualization. *IEEE Comput.* **38**(4), 34–41 (2005)
8. Awerbuch, B., Brinkmann, A., Scheideler, C.: Anycasting in adversarial systems: routing and admission control. *Lect. Notes Comput. Sci.* **2719**, 1153–1168 (2003)
9. Balasubramaniam, S., Leibniz, K., Lio, P., Botvich, D., Murata, M.: Biological principles for Future Internet architecture design. *IEEE Commun. Mag.* **49**(7), 44–52 (2011)
10. Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. *Science* **286**, 509–512 (1999)
11. Botero, J.F., Hesselbach, X., Fischer, A., de Meer, H.: Optimal mapping of virtual networks with hidden hops. *Telecommun. Syst.* **51**(4), 273–282 (2012)
12. Burakowski, W.: Role of network virtualization in designing Future Internet. In: *Proceedings of the 15th Telecommunications Network Strategy and Planning Symposium (Networks'12)*, pp. 1–3 (2012)
13. Burakowski, W., et al.: IIP System specification level 1 and 2, POIG IIP project deliverable (2011)
14. Cerf, V.G.: The day the Internet age began. *Nature* **461**(7268), 1202–1203 (2009)
15. China Education and Research Network: <http://www.edu.cn/english/>. Accessed on 24 Nov 2014
16. Cholda, P., Gozdecki, J., Kantor, M., Wielgosz, M., Pach, A.R., Wajda, K., Rak, J.: Provisioning concepts of the IIP Initiative. In: *Proceedings of the 13th International Conference on Transparent Optical Networks (ICTON'11)*, pp. 1–4 (2011)
17. Chou, H.-Z., Wang, S.-C., Kuo, S.-Y., Chen, I.-Y., Yuan, S.-Y.: Randomised and distributed methods for reliable peer-to-peer data communication in wireless ad hoc networks. *IET Commun.* **1**(5), 915–923 (2007)
18. Chowdhury, N.M., Boutaba, R.: Network virtualization: state of the art and research challenges. *IEEE Commun. Mag.* **47**(7), 20–26 (2009)
19. D'Ambrosio, M., Fasano, P., Marchisio, M., Vercellone, V., Ullio, M.: Providing data dissemination services in the Future Internet. In: *Proceedings of the IEEE Global Communications Conference (IEEE GLOBECOM'08)*, pp. 1–6 (2008)
20. Dedecker, P., Hoebeke, J., Moerman, I., Moreau, J., Demeester, P.: Network virtualization as an integrated solution for emergency communication. *Telecommun. Syst.* **52**(4), 1859–1876 (2013)
21. Dijkstra, E.: A note on two problems in connexion with graphs. *Numer. Math.* **1**, 269–271 (1959)
22. Din, D.: Anycast routing and wavelength assignment problem on WDM network. *IEICE Trans. Commun.* **E88-B**(10), 3941–3951 (2005)
23. Domingue, J., Galis, A., Gavras, A., Zahariadis, T., Lambert, D., Cleary, F., Daras, P., Krco, S., Mueller, H., Li, M.-S., Schaffers, H., Lotz, V., Alvarez, F., Stiller, B., Karnourkos, S., Avessta, S., Nilsson, M. (Eds.): *The Future Internet – Future Internet Assembly 2011: Achievements and Technological Promises*. *Lecture Notes in Computer Science*, vol. 6656. Springer, Berlin (2011)

24. European Commission: Council decision establishing the specific program implementing HORIZON 2020 – the framework programme for research and innovation (2014–2020). Brussels, 2011. Work Programme 5.i. Leadership in technologies. Draft Discussion Doc., pp. 86–86 (2013)
25. European Commission: <http://ec.europa.eu>. Accessed on 21 Nov 2014
26. Feldmann, A.: Internet clean-slate design: what and why? *ACM SIGCOMM Comput. Commun. Rev.* **37**(3), 59–64 (2007)
27. FIRE: Future Internet Research and Experimentation: <http://cordis.europa.eu/fp7/ict/fire/>. Accessed on 24 Nov 2014
28. Future Internet Assembly: <http://www.future-internet.eu/home/future-internet-assembly.html>. Accessed on 20 Nov 2014
29. Future Internet Engineering (IIP) Initiative: <http://www.iip.net.pl>. Accessed on 24 Nov 2014
30. GEANT2 project: <http://www.geant2.net/>. Accessed on 24 Nov 2014
31. Gedik, B., Liu, L.: A scalable peer-to-peer architecture for distributed information monitoring applications. *IEEE Trans. Comput.* **54**(6), 767–782 (2005)
32. Ghodsi, A., Koponen, T., Rajahalme, J., Sarolahti, P., Shenker, S.: Naming in content-oriented architectures. In: Proceedings of the ACM SIGCOMM’11 Workshop on Information-Centric Networking, pp. 1–6 (2011)
33. Gladysz, J., Walkowiak, K.: Optimization of survivable networks with simultaneous unicast and anycast flows. In: Proceedings of the RNDM’09 @ International Conference on Ultra Modern Telecommunications & Workshops (ICUMT’09), pp. 1–6 (2009)
34. Global Environment for Network Innovations (GENI) project: <http://www.geni.net/>. Accessed on 24 Nov 2014
35. Goh, K.-I., Oh, E.S., Jeong, H., Kahng, B., Kim, D.: Classification of scale free networks. arXiv:cond-mat/0205232, v2 (2002)
36. Gozdecki, J., Kantor, M., Wajda, K., Rak, J.: A flexible provisioning module optimizing utilization of resources for the future internet IIP initiative. In: Proceedings of the 15th International Telecommunications Network Strategy and Planning Symposium (NETWORKS’12), pp. 1–6 (2012)
37. Gozdecki, J., Kantor, M., Wajda, K., Rak, J.: Methods of network resource provisioning for the Future Internet IIP Initiative. *Telecommun. Syst.* **61**, 235–246 (2016)
38. Habib, M.F., Tornatore, M., De Leenheer, M., Dikbiyik, F., Mukherjee, B.: Design of disaster-resilient optical datacenter networks. *IEEE/OSA J. Lightwave Technol.* **30**(16), 2563–2573 (2011)
39. Ho, P.-H., Mouftah, H.T.: A framework for service-guaranteed shared protection in WDM mesh networks. *IEEE Commun. Mag.* **40**(2), 97–103 (2002)
40. Ho, P.-H., Tapolcai, J., Mouftah, H.T.: Diverse routing for shared protection in survivable optical networks. In: Proceedings of the IEEE Global Communications Conference (IEEE GLOBECOM’03), vol. 5, pp. 2519–2523 (2003)
41. Ho, P.-H., Tapolcai, J., Cinkler, T.: Segment shared protection in mesh communications networks with bandwidth guaranteed tunnels. *IEEE/ACM Trans. Networking* **12**(6), 1105–1118 (2004)
42. Hofmann, M., Beaumont, L.: *Content Networking: Architecture, Protocols, and Practice*. Morgan Kaufmann, San Francisco (2005)
43. IEEE Communications Society: *A Brief History of Communications*, 2nd edn. IEEE, Piscataway (2012)
44. Koponen, T., Chawla, M., Chun, B.-G., Ermolinskiy, A., Kim, K.H., Shenker, S., Stoica, I.: A data-oriented (and beyond) network architecture. In: Proceedings of the ACM Annual Conference of the Special Interest Group on Data Communication (ACM SIGCOMM’07), pp. 181–192 (2007)
45. Kounavis, M.E., Campbell, A.T., Chou, S., Modoux, F., Vicente, J., Zhuang, H.: The Genesis Kernel: a programming system for spawning network architectures. *IEEE J. Sel. Areas Commun.* **19**(3), 511–526 (2001)

46. Low, C.P., Tan, C.L.: On anycast routing with bandwidth constraint. *Comput. Commun.* **26**(14), 1541–1550 (2003)
47. Metz, C.: IP anycast point-to-(any) point communication. *IEEE Int. Comput.* **6**(2), 94–98 (2002)
48. MobilityFirst Future Internet Architecture project: <http://mobilityfirst.winlab.rutgers.edu/>. Accessed on 24 Nov 2014
49. Molisz, W., Rak, J.: Region protection/restoration scheme in survivable networks. *Lect. Notes Comput. Sci.* **3685**, 442–447 (2005)
50. Mukherjee, B.: *Optical WDM Networks*. Springer, Berlin (2006)
51. Named Data Networking project: <http://www.named-data.net>. Accessed on 24 Nov 2014
52. National Science Foundation: <http://www.nsf.gov>. Accessed on 24 Nov 2014
53. NSF Future Internet Architecture project: <http://www.nets-fia.net>. Accessed on 24 Nov 2014
54. NSF NeTS FIND Initiative: <http://www.nets-find.net>. Accessed on 24 Nov 2014
55. Pan, J., Paul, S., Jain, R.: A survey of the research on future Internet architectures. *IEEE Commun. Mag.* **49**(7), 26–36 (2011)
56. Petcu, D., Galis, A., Karnouskos, S.: The Future Internet cloud: computing networking and mobility. Introduction to chapter on computing and mobile clouds. In: *The Future Internet – FIA 2013: Validated Results and New Horizons*, pp. xiii–xv (2013)
57. Rak, J.: k -Penalty: a novel approach to find k -disjoint paths with differentiated path costs. *IEEE Commun. Lett.* **14**(4), 354–356 (2010)
58. Rak, J.: Fast service recovery under shared protection in WDM networks. *IEEE/OSA J. Lightwave Technol.* **30**(1), 84–95 (2012)
59. Rak, J., Walkowiak, K.: Reliable anycast and unicast routing: protection against attacks. *Telecommun. Syst.* **52**(2), 889–906 (2013)
60. Sallai, G.: Chapters of Future Internet research. In: *Proceedings of the 4th International Conference on Cognitive Infocommunications (CogInfoCom'13)*, pp. 161–166 (2013)
61. Schoenwaelder, J., Fouquet, M., Rodosek, G.D., Hochstatter, I.C.: Future Internet = Content + services + management. *IEEE Commun. Mag.* **47**(7), 27–33 (2009)
62. Software-defined networking: the new norm for networks. White paper, Open Networking Foundation (ONF), April 2012: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>. Accessed on 08 Mar 2015
63. The FP7 4WARD project: <http://www.4ward-project.eu/>. Accessed on 25 Nov 2014
64. Touch, J.: Dynamic Internet Overlay deployment and management using the X-bone. *Comput. Networks* **36**(2–3), 117–135 (2001)
65. Triukose, S., Wen, Z., Rabinovich, M.: Content delivery networks: how big is big enough? *ACM SIGMETRICS Perform. Eval. Rev.* **37**(2), 59–60 (2009)
66. Trossen, D., Parisi, G.: Designing and realizing an information-centric Internet. *IEEE Commun. Mag.* **50**(7), 60–67 (2012)
67. Tselentis, G., et al. (Eds.): *Towards the Future Internet – emerging trends from European research*. Future Internet Assembly (FIA 2010). IOS Press, Amsterdam (2010)
68. Turner, J., Taylor, D.: Diversifying the Internet. In: *Proceedings of the IEEE Global Communications Conference (IEEE GLOBECOM'05)*, vol. 2, pp. 765–760 (2005)
69. Walkowiak, K.: Anycast communications, a new approach to survivability of connection-oriented networks. *Commun. Comput. Inform. Sci.* **1**, 378–389 (2007)
70. Walkowiak, K.: Anycasting in connection-oriented computer networks: models, algorithms and results. *Int. J. Appl. Math. Comput. Sci.* **20**(1), 207–220 (2010)
71. Walkowiak, K., Rak, J.: Shared backup path protection for anycast and unicast flows using the node-link notation. In: *Proceedings of the IEEE International Conference on Communications (IEEE ICC'11)*, pp. 1–6 (2011)
72. Walkowiak, K., Rak, J.: Simultaneous optimization of unicast and anycast flows and replica location in survivable optical networks. *Telecommun. Syst.* **52**(2), 1043–1055 (2013)
73. Xia, W., Wen, Y., Foh, C.H., Niyato, D., Xie, H.: A survey on software-defined networking. *IEEE Commun. Surv. Tutorials* **17**(1), 27–51 (2015)

74. Xylomenos, G., Ververidis, C.N., Siris, V.A., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K.V., Polyzos, G.C.: A survey of information-centric networking research. *IEEE Commun. Surv. Tutorials* **16**(2), 1024–1049 (2014)
75. Yin, H., Liu, X., Min, G., Lin, Ch.: Content delivery networks: a bridge between emerging applications and future IP networks. *IEEE Network* **24**(4), 52–56 (2010)
76. Zhou, S., Mondragon, R.J.: The rich-club phenomenon in the Internet topology. *IEEE Commun. Lett.* **8**(3), 180–182 (2004)

Chapter 9

Resilience of Wireless Mesh Networks



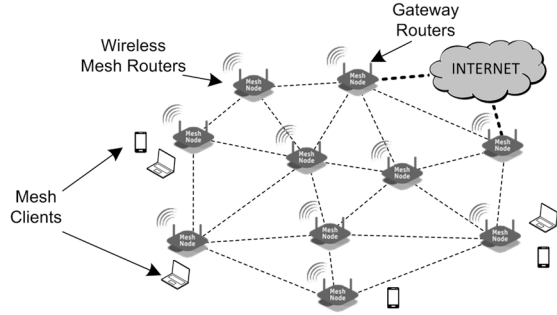
The second case study considered in this book refers to *Wireless Mesh Networks* (WMNs) formed by stationary mesh routers organized in a mesh topology [3, 22], providing transportation of flows originating from mesh clients (with little or no mobility). As presented in Fig. 9.1, WMN nodes have *mesh* capability, meaning their functioning is not restricted only to local data transmission. Instead, they can also relay information belonging to flows from other WMN nodes in a multi-hop fashion [18, 25]. If equipped with necessary functionality at specific nodes (i.e., gateways), WMNs may also be utilized to provide connectivity with external networks, e.g., the Internet [5, 8, 68].

Most WMN architectures are based on the IEEE 802.11 standard, defining how wireless devices can be mutually interconnected to create a mesh network [26]. Compared to Wi-Fi solutions, the mesh structure of these networks implies a substantial enhancement in terms of the coverage area, connectivity, and scalability improvement, as well as simplifying deployment and maintenance activities [18, 68]. WMN end users are also provided with single-domain connectivity instead of switching between Wi-Fi hot-spots. It has been proved that the grid organization of WMN nodes provides up to 50% higher throughput than a random node placement [68].

Due to the utilization of the 71–86 GHz band [29, 39, 66], as well as highly directional antennas, the effective transmission rate can be as high as 1–10 Gb/s per millimeter-wave link with a transmission range of at least several kilometers [64, 72]. Therefore, WMNs can be seen as a promising alternative to wired local or even metropolitan area networks providing the last few miles of connectivity, especially in sparsely populated rural areas [22, 42].

It is also possible to equip each WMN router with *MIMO* technology (i.e., *multiple-input multiple-output*) utilizing multiple orthogonal channels [8]. This, in turn, leads to a further substantial increase in the network capacity [31, 71]. MIMO transmission is essential in urban areas encountering signal distortions, where such systems help amplify and rebuild signal levels, while directional antenna settings

Fig. 9.1 Example architecture of a Wireless Mesh Network including wireless mesh routers, mesh clients, and gateway routers



visibly reduce interference between neighboring channels [68]. What is similarly essential is that WMNs can provide connectivity among users without direct *Line of Sight (LOS)* links.

WMNs have also been shown to offer low connection costs in the backhaul area [8]. That is why using WMN solutions (e.g., instead of applying the fiber optic technology) is well justified for economic and practical reasons. It mainly refers to mobile operators not having their own fiber infrastructure, who otherwise would have to either deploy their own fiber network (which is very expensive in rural areas [20]) or try to lease capacity from other network providers. Deployment of WMNs has also been proposed to obtain affordable access networks for underdeveloped regions [42].

In the last decade, many research teams have been addressing the problems of capacity planning, placement of WMN nodes, routing, channel assignment, power control, topology control, etc. These problems are indeed very closely linked due to the nature of wireless interference. Therefore, when designing a WMN network, a joint consideration of these problems provides much better results in practice than in the case of a separate analysis. A comprehensive overview of joint design problems is presented in [42].

Several WMN installations are already in use in Europe, Australia, and the USA [17], deployed using equipment provided by, e.g., TerraNet, ArubaNetworks, or Motorola [4, 37, 62]. Example WMN architectures include city-wide (or campus-wide) networks in Las Palmas, Spain and Corpus Cristi [65], Cambridge, Massachusetts, USA [68], Houston, USA [49], Oulu, Finland [59], Madison, USA [69], or Dartmouth, USA [24], with the number of nodes ranging from tens to hundreds, and the area of coverage measured in tens of square kilometers.

Apart from inheriting the typical characteristics of the general ad hoc networking concept (i.e., decentralized design, distributed communications), WMNs are known to exhibit characteristics that are novel in the wireless context but rather typical to wired networks, i.e., stationary nodes, no LOS connectivity, high capacity, and no limitations referring to node energy consumption [42].

Considering the transmission of information, we can even say that WMNs possess the most wired network characteristics, with the only apparent exception being the time-varying link stability. Therefore, applying the hop count metric

for routing purposes in WMNs is inefficient (as shown in [13]). To respond to the dynamic characteristics of WMN links, several routing metrics have been proposed, the most important ones including the expected transmission count (EXT) [12], expected transmission time (ETT) [16], a metric of interference and channel switching (MIC) [70], or multichannel routing (MCR) [32]. They were designed to support WMN routing algorithms, e.g., AODV-ST [46], opportunistic Ex-OR [9], multipath routing [19], geographic routing [33], hierarchical routing [48], or multi-radio routing [32].

However, by incorporating the mentioned metrics into either a single-path or multipath routing [18], the impact of time-varying disruptions leading to partial/complete degradation of the effective capacity of WMN links can be reduced only in a reactive way. Proactive protection against failures (commonly known to achieve better performance, e.g., concerning the reduction of lost traffic after failures) is a relatively new research direction for WMNs. The problem is indeed essential since independent of the cause of failure (whether the result of an accident, forces of nature, or an intentional attack [63]), data and revenue losses encountered at high transmission rates of several Gb/s may undoubtedly be severe.

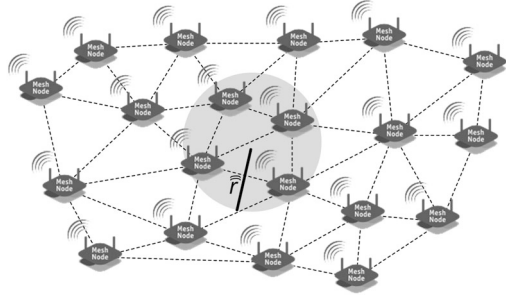
This chapter focuses on the failures of both WMN nodes and links. In particular, failures of WMN links can be covered by failure scenarios of the respective incident nodes (the topic addressed in Sect. 9.1). Failures of WMN links are commonly transient (i.e., not observed after the interval of a negative factor duration).

Although a significant part of the research is related to scenarios of isolated random failures of single nodes resulting from software errors or physical faults [1], such an assumption is improper for WMNs in many realistic scenarios. Example cases comprise natural disasters like earthquakes, volcano eruptions, tornadoes, or malicious human activities, including, e.g., bomb explosions [35], resulting in spatial correlation of failures of WMN nodes. WMN links are, in turn, very vulnerable to heavy precipitation, responsible for a remarkable signal attenuation.

In such cases, it is commonly assumed that the extent of negative outcomes depends on the characteristics of a particular event, with the major factor being the distance of a network element from the failure epicenter. This, in turn, gives rise to the region failure scenario [30, 38, 51, 52] addressing simultaneous failures of multiple nodes located close enough to suffer from the results of the event. Following [52], regions of failure can be defined concerning either network topology or geometry. The latter approach, i.e., a geometrical representation of a failure region determined by a circular radius area, shown in Fig. 9.2, is mainly used due to the predominant role of a node distance from the event epicenter [51, 52].

In particular, to the best of our knowledge, no survivability measures are available to evaluate the performance of WMNs under region failures leading to simultaneous failures of multiple WMN nodes (as well as related links). Also, very few proposals refer to proactive protection of WMN flows against link failures. To provide the respective solutions, Sect. 9.1 introduces the appropriate survivability measures for WMNs, while Sect. 9.2 introduces a new approach to proactive protection against weather-based region disruptions based on automatic antenna alignment features. Section 9.3 concludes this chapter.

Fig. 9.2 Example of a failure region: dark gray circle centered at the epicenter of disruptions, and characterized by a given radius \hat{r} , represents the area of possible failures of WMN nodes



9.1 Measures of Wireless Mesh Networks Survivability

Due to the dependency of region-based failures on multiple characteristics, region failures need a detailed evaluation concerning their influence on the ratio of WMN performance degradation (e.g., measured in terms of the fraction of flow surviving failures of WMN nodes located inside a given failure region).

In this section, we present our approach to WMN regional failure assessment from [45] based on three introduced measures of WMN survivability for a circular region failure scenario under the random location of failure epicenters, i.e.:

- Region failure survivability function (RFS) being the cumulative probability of all region failure scenarios δ occurrence, for which at least ψ percent of flows are successfully served after failures
- p -fractile region failure survivability function (PFRS), providing information on total flow reduction to at most ψ percent after a failure at certain probability p
- Expected percentage of total flow delivered after a region failure as a function of region radius \hat{r} (EPFD)

Apart from providing a means of assessment of a given WMN to region disruptions, these measures are also proposed to enable comparisons of characteristics between different WMNs. To the best of our knowledge, besides our methodology from [45], no other relevant techniques are available in the literature appropriate for measuring the vulnerability of WMNs to regional failures of differentiated radiuses \hat{r} of failure regions.

The methodology of network survivability evaluation is well-established concerning wired networks (see, e.g., [21, 47, 53, 55, 61, 67]). Only a few proposals are available for wireless networks focusing, e.g., on the connectivity of network topology as a measure of fault tolerance [50]. Connectivity can be generally used to provide a binary answer to whether the network is k -connected, i.e., able to provide transmission continuity after a simultaneous failure of $k-1$ nodes. This idea has been extended to cover, e.g., average connectivity [7], distance connectivity [6], or path connectivity [23].

However, most existing proposals for WMN evaluation are unsuitable for a regional failure scenario with faults assumed to occur only in bounded areas. To address this problem, the respective region-based connectivity was proposed (see, e.g., [35, 50–52]). Concerning the scenario of circular failure regions, we can distinguish the models of:

- Deterministic failures (e.g., the single circular model from [52]), where any node located within the failure region is assumed to always fail with probability 1.
- Probabilistic failures with a probability of a node failure due to a disruptive event depending on the node distance from the failure epicenter [35]. This failure probability is assumed to decrease when the node distance from the failure epicenter increases.

Probabilistic models seem to provide more accurate results due to the common nondeterministic characteristics of natural disasters or attacks, resulting in failures of nodes located within failure regions with a certain probability. It is worth noting that available probabilistic approaches are not limitation-free. For instance, in [35], the size of a failure region (given by radius \hat{r}) is assumed to be constant. Another constraint in [35] is that the probability of a node failure (even though decreasing with the increase of a node distance from the failure epicenter) is constant in each i -th area between two consecutive concentric annuluses (see Fig. 9.3a), which results in over- or underestimating the node failure probability values in some areas.

Considering proposals of WMNs characteristics evaluation under region failures, several approaches have been introduced (e.g., [50–52]) to determine whether transmission in WMNs is possible between pairs of non-faulty nodes. To the best of our knowledge, our proposal described in this section is the first one to introduce the WMN survivability measures for the case of varying region radiuses \hat{r} , and using the continuous function of node failure probability (see Fig. 9.3b and Eq. 9.3) that covers the models from [35, 52] as special cases.

It is worth noting that similar survivability measures have been proposed in the literature so far only for random failure scenarios in wired networks (see, e.g., [36]). However, they were designed for failures of network elements assumed to

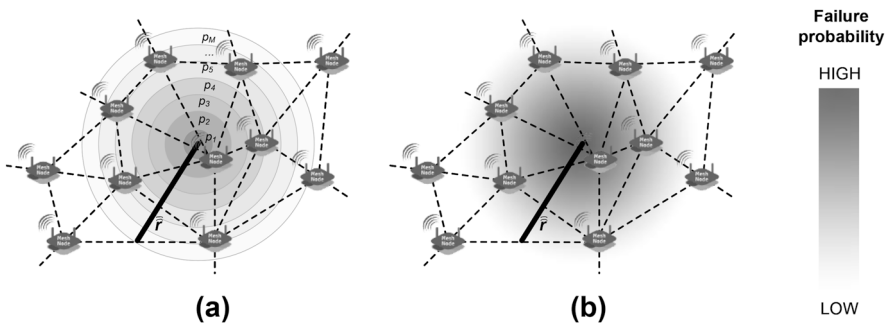


Fig. 9.3 Visualization of region failure probabilities: (a) from [35], and (b) the proposed one

be statistically independent and equally probable, which is entirely in contrast to characteristics of WMN regional failures.

In the remaining part of this section, we first present details of the assumed network model (Sect. 9.1.1), followed by the proposed measures to evaluate the vulnerability of WMNs to regional failures (Sect. 9.1.2). Next, we describe the methodology of WMN survivability evaluation (Sect. 9.1.3) and comment on the results of simulations performed, for the example network topologies (Sect. 9.1.4).

9.1.1 Network Model

In this chapter, we model the WMN topology by graph $G = (N, A)$, where N represents the set of WMN stationary nodes (following [42]), while A denotes the set of directed arcs $a_h = (i, j)$. Each WMN link between neighboring nodes i and j is represented by two arcs in opposite directions. Additional information refers to the location of each node n defined by coordinates (\bar{x}_n, \bar{y}_n) . Despite the assumed stationary characteristics of network nodes, the methodology of network assessment presented in this section can also be easily adapted to the case of mobile nodes (if performed concerning the instant topology of a network at time t).

The available capacity of any WMN link is a result of multiple factors, the most important ones being medium access protocol implementation, interchannel interference implied by the respective link scheduling algorithm [11, 18], or time-varying factors including, e.g., weather-based disruptions caused by heavy rain falls (general propagation conditions) [27]. Since the effective capacity of any WMN link changes over time, it is reasonable to perform evaluations at a given time t , i.e., assuming that the capacity of arc a_h is equal to $c_h(t)$.

The set of demands D consists of demands d_r defined by ordered triples (s_r, t_r, c_r) , i.e., described by source and destination nodes s_r and t_r , and the demanded capacity c_r .

Two matrices are used in our model description: A_{nn} and D_{nn} . Node-to-node incidence matrix A_{nn} provides information on connectivity with elements defined by formula (9.1).

$$a_{i,j} = \begin{cases} 1, & \text{if arc } a_h = (i, j) \in A_{nn} \\ 0, & \text{otherwise} \end{cases} \quad (9.1)$$

Information about aggregate capacities required for flows (commodities) between given pairs of end nodes is stored in elements $d_{s,t}$ of matrix D_{nn} .

$$d_{s,t} = (s_r, t_r, c_r) \quad (9.2)$$

During evaluations, the location of a failure epicenter is chosen at random (i.e., following the uniform distribution function of failure epicenter coordinates) within the smallest rectangular area containing the network. We assume a probabilistic

failure scenario with a disruptive event affecting nodes localized within a given radius \hat{r} from the failure epicenter. In particular, in our model:

- Radius \hat{r} of a failure circular region is uniformly distributed over $(0, \hat{r}_{\max})$, where \hat{r}_{\max} is equal to half of the largest Euclidean distance between any two nodes in the network;
- Probability $P(\hat{r}_n)$ of node n failure is given by a decreasing continuous function of the distance \hat{r}_n between node n and the failure epicenter (see Fig. 9.3b and Eq. 9.3). $P(\hat{r}_n)$ is thus the generalization of the respective formula from [35].

$$P(\hat{r}_n) = \begin{cases} -\frac{\hat{r}_n}{\hat{r}} + 1 = -\frac{\sqrt{(\bar{x}_n - \bar{x})^2 + (\bar{y}_n - \bar{y})^2}}{\hat{r}} + 1 & \text{if } \hat{r}_n \leq \hat{r} \\ 0, & \text{otherwise} \end{cases} \quad (9.3)$$

where:

- (\bar{x}_n, \bar{y}_n) are coordinates (location) of node n ;
- (\bar{x}, \bar{y}) are coordinates (location) of the failure epicenter;
- \hat{r} is the radius of a failure region;
- \hat{r}_n is the distance of node n from the failure epicenter.

It is reasonable to introduce the WMN node failure probability function as given in Eq. 9.3 since, following [35], the negative impact of real physical attacks (e.g., bomb explosions or electromagnetic pulse (EMP) attacks), as well as natural disasters (earthquakes, floods, etc.), attenuates gradually with the increase of the distance of WMN nodes from the failure epicenter. As given in [35], the maximum value of node failure probability can be assumed to be equal to 1 for locations of nodes matching exactly the failure epicenter. Its lowest value of 0 is, in turn, attributed to nodes located at a distance \hat{r}_n not smaller than \hat{r} from the failure epicenter.

It is worth noting that this gradual attenuation of $P(\hat{r}_n)$ values with the increase of the distance \hat{r}_n can be disturbed by several environmental factors, e.g., topography or node protection characteristics. However, if we neglect them to simplify the analysis (following [35]), the decrease of probability $P(\hat{r}_n)$ of node n failure becomes linear with the increase of node n distance from the epicenter of disruptions, as introduced in Eq. 9.3.

9.1.2 Proposed Measures to Evaluate the Survivability of WMNs

The following notation is used in the remaining part of Sect. 9.1:

- δ a regional failure scenario given by the set of nonoperational nodes (after the outage)

$P(\delta)$	probability of occurrence of a failure scenario δ
$\Psi(\delta)$	random variable referring to the percentage ψ of flows delivered in scenario δ
$p_{\Psi}(\psi)$	probability density function of percentage ψ of flows surviving the region failure, defined by Eq. 9.4

$$p_{\Psi}(\psi) = \sum_{\delta: \Psi(\delta)=\psi} P(\delta) \quad (9.4)$$

We introduce three measures of WMN survivability for a regional failure scenario, i.e.:

1. Region failure survivability function (RFS) of the percentage ψ of flows successfully transmitted after regional failures:

$$RFS(\psi) = \sum_{\delta: \Psi(\delta) \geq \psi} P(\delta) = 1 - \sum_{\delta: \Psi(\delta) < \psi} P(\delta) = 1 - cdf(\Psi) \quad (9.5)$$

As given in Eq. 9.5, $RFS(\psi)$ is defined for any value of ψ as the cumulative probability of all region failure scenarios δ (i.e., for differentiated radiuses \hat{r} of failure regions) for which at least ψ percent of flows survived the failure. It can be thus expressed as the reverse cumulative distribution function of Ψ . Although Eq. 9.5 shows some similarities with the respective one from [36] for wired networks, the calculation of $P(\delta)$ values is entirely different.

2. p -fractile region survivability (PFRS):

$$PFRS(p) = \inf \left\{ \psi : \sum_{\delta: \Psi(\delta) < \psi} P(\delta) = p \right\} \quad (9.6)$$

Following formula (9.6), the value of p -fractile region survivability refers to the minimum percentage ψ of flows delivered after a regional failure, for which the probability of not exceeding this value is equal to p . PFRS thus returns useful information about probability p that the total flow is reduced to at most ψ percent after the failure.

Since RFS and PFRS measures do not depend directly on the radius \hat{r} (i.e., they allow radius \hat{r} to take any value from $(0, \hat{r}_{\max})$ interval), they are designed to give general information on network vulnerability to regional failures. These measures are thus appropriate if the objective is to analyze the performance of WMNs independent of the failure region size \hat{r} . However, the information they provide is of different types.

For instance, if for a given WMN, at least ψ percent of traffic should be delivered (e.g., because such a portion of traffic is considered to be critical based on the Service Level Agreement), then RFS is the appropriate one to provide

information about probability p of fulfilling this requirement under regional failures independent of size \hat{r} of the failure region. Naturally, the greater the value of p , the better performance of a network can be achieved.

PFRS is, in turn, a suitable measure for a network operator to determine, given the respective probability p , what is the upper bound on the fraction ψ of flow surviving a regional failure. It is, therefore, useful to give information on the probability that not all of the ψ percent of flows (e.g., referred to as the critical flow) will survive the regional failure, i.e., in statements like: “with probability 0.7, the total flow will be reduced to at most 80% of the traffic served before the regional failure.”

The following EPFD function is introduced to obtain detailed characteristics of a WMN performance related to particular radiuses \hat{r} of failure regions.

3. Expected percentage of total flow delivered after a failure (EPFD) as a function of region radius \hat{r} :

$$EPFD(\hat{r}) = \sum_{\psi} \psi \cdot p_{\psi}(\psi, \hat{r}) \tag{9.7}$$

where:

\hat{r} is the radius of a failure region;
 $p_{\psi}(\psi, \hat{r})$ is the probability density function of Ψ defined for failure region of radius \hat{r} .

$$p_{\psi}(\psi, \hat{r}) = \sum_{\delta: \Psi(\delta)=\psi; \hat{r}} P(\delta) \tag{9.8}$$

$EPFD(\hat{r})$ is defined in Eq. 9.7 as the expected value of the percentage of flows to survive failures of nodes bounded in circular regions, i.e., derived using the probability density function $p_{\psi}(\psi, \hat{r})$ obtained for failure regions of a given radius \hat{r} (see formula (9.8)).

Concerning scenarios of EPFD measure utilization, it can be helpful in any performance analysis/comparison of WMNs under regional failures being the result of, e.g., natural disasters (like floods or volcano eruptions), for which the failure region is commonly expected to have a circular shape defined by a given radius \hat{r} . Another application of EPFD measure would be, e.g., when expecting failures confined to a certain region characterized by radius \hat{r} (e.g., incoming flood), to predict its impact on WMN performance, being helpful to take preventive actions.

The later part of this section provides information on how to utilize the three introduced measures to evaluate the vulnerability of WMNs to regional failures and how to use them to compare performance characteristics of different topologies.

9.1.3 Method of a WMN Survivability Evaluation

This section explains our methodology of WMN survivability characteristics evaluation under regional failures. In particular, we focus on determining the introduced RFS, PFRS, and EPFD characteristics, for example, WMNs.

Proposed measures are derived from the auxiliary function $F[\psi]$ providing information on the frequency a given percentage of flows ($\psi \in \{0, 1, \dots, 100\}$) is successfully delivered after regional failures. $F[\psi]$ values can be collected for a given WMN based on network performance observations after consecutive disruptive events implying failures of WMN nodes confined to given regions. However, deriving any characteristics based on real-life experiments is time-consuming and practically impossible due to rather long inter-failure time intervals (typically measured in months/years).

In this section, an iterative procedure is presented to simulate consecutive region failures in a way that eliminates the inter-failure time. In this way, it is possible to analyze the performance of existing networks and predict the survivability characteristics of planned (i.e., non-deployed) WMNs using information related to the abstract WMN topology and estimated demand volumes.

The 13-step procedure to determine $F[\psi]$ values for a single set of demands is given in Fig. 9.4. The most essential input information is related to the following:

- Topology of existing/planned WMN defined by a graph G with sets N and A of nodes and directed arcs, representing network nodes and links, respectively
- Location of network nodes defined by coordinates (\bar{x}_n, \bar{y}_n)
- Demands d_r , given by the requested throughput c_r , and source and destination nodes s_r / t_r

After initialization, Steps 1–2, the purpose of each iteration given by Steps 3–13 is to obtain the percentage ψ of flows delivered after failures of WMN nodes occurring in a given failure region. The coordinates of each failure epicenter and the radius \hat{r} of a failure region are defined as random values by the continuous uniform distribution function (following [35]).

In particular, it implies that in each iteration of the analyzed procedure:

- The location of a failure epicenter is chosen at random within the smallest rectangular area containing the WMN topology, using the continuous uniform distribution function.
- Radius \hat{r} of a failure circular region is uniformly distributed over $(0, \hat{r}_{\max})$, with \hat{r}_{\max} equal to half of the largest Euclidean distance between any two nodes in the network.

After the iteration initialization, Steps 3–5, Step 6 is to identify the set of failed nodes (based on formula (9.3)). To evaluate the percentage ψ of flows delivered in a given regional failure scenario, for each flow with both end nodes being non-faulty, our method tries to find an alternate path of capacity c_r (Steps 7–9). If the new path is found, but, due to link capacity limitations, it cannot be assigned the demanded

INPUT

- WMN topology given by graph $G = (N, A)$, where N and A are the sets of nodes and directed arcs, accordingly,
- location of network nodes determined by coordinates (\bar{x}_n, \bar{y}_n) ,
- node-to-node incidence matrix A_{nm} ,
- capacities c_h of arcs $a_h = (i, j) \in A$,
- matrix D_{mn} of aggregate capacities c_r required for demands d_r between end nodes s_r and t_r ,
- total load c (the aggregate value of all transported flows before the occurrence of a region failure),
- total number FR of analyzed failure regions

OUTPUT $F[\psi]$ function**VARIABLES**

- \hat{f} the aggregate flow restored after a region failure,
- \bar{c}_h free (residual) capacity at arc a_h ,
- ic iteration counter,
- c_r capacity to be reserved for demand d_r along links traversed by the respective paths in G

-
- Step 1 For each $\psi \in \{0, 1, \dots, 100\}$, set $F[\psi] = 0$.
- Step 2 Set $ic := 0$.
- Step 3 Create the temporal incidence matrix \bar{A}_{nm} by assigning $\bar{A}_{nm} := A_{nm}$.
- Step 4 Set $\hat{f} := 0$.
- Step 5 Use the uniform distribution function to determine the coordinates of the next failure epicentre, as well as the radius \hat{r} of a failure region taken from range $(0; \hat{r}_{\max})$.
- Step 6 Use the node failure probability function (Eq. 9.3) to determine the set of failed nodes.
- Step 7 In \bar{A}_{nm} , set 0 to all elements representing failed links after failures in a given region.
- Step 8 For each arc a_h , set the initial residual capacity $\bar{c}_h = c_h$ (i.e., to the value of the total link capacity available at a_h).
- Step 9 For each demand d_r with both end nodes s_r and t_r not affected by the failure:
- 9.1 Set the value c_r denoting capacity not assigned to demand d_r to the initial value: $\bar{c}_r := c_r$.
 - 9.2 Find the shortest path π using the distance metric and the incidence matrix \bar{A}_{nm} ,
 - 9.3 Determine the capacity $\rho_r := \min_{a_h \in \pi} \bar{c}_h$ of π , where \bar{c}_h is the current residual capacity at arc a_h . If $\bar{c}_r \leq \rho_r$, then increase \hat{f} by $\mu := \bar{c}_r$, else increase \hat{f} by $\mu := \rho_r$.
 - 9.4 Decrease \bar{c}_r by μ .
 - 9.5 For each arc a_h traversed by path π , calculate new residual capacity $\bar{c}_h := \bar{c}_h - \mu$.
- Step 10 For all affected flows already not fully served (i.e., for which $\bar{c}_r > 0$), try to find the next shortest path. If such a path exists, assign a new portion of capacity to it along the respective links, increase by, decrease by, and calculate the respective new residual capacities of arcs a_h traversed by this path. Repeat these actions for each demand r until $\bar{c}_r = 0$, or no new path can be found.
- Step 11 Calculate the percentage of flows \hat{f}/f restored after failures occurring in a given region (where f is the total traffic served before the failure), and increment the value of the element in F determined by index $\lfloor 100 \cdot \hat{f} / f \rfloor$.
- Step 12 Increment the value of ic .
- Step 13 If $ic < FR$, then go to Step 3.
-

Fig. 9.4 Method of determining $F(\psi)$ values

capacity c_r , multipath routing is then applied to increase as much as possible the capacity assigned to demand d_r after a regional failure (Step 10).

The percentage ψ of flows successfully delivered after a failure is calculated in Step 11 based on the ratio of the aggregate flow \hat{f} restored after the failure to the total flow f being transported before the failure (i.e., after finding the alternate paths for all demands in a given region failure scenario). Following Steps 12–13, the analysis is repeated until the number FR of failure regions is evaluated.

All three introduced functions (RFS, PFRS, and EPFD) are next derived based on $F[\psi]$ values. In particular:

- RFS(ψ) is calculated based on empirical probabilities of restoring ψ percent of flows after failures (each such probability is obtained by dividing the respective value of $F[\psi]$ by FR , i.e., by the total number of analyzed failure regions). According to formula (9.5), RFS(ψ) is determined as the reverse cumulative distribution function of Ψ .
- PFRS(p) is obtained based on the cumulative distribution function of Ψ (formula (9.6)).
- EPFD(\hat{r}) is calculated based on probability density functions $p_\psi(\psi, \hat{r})$ found separately for each radius \hat{r} of a failure region using Eq. 9.7.

To find the optimal solution to the problem of determining a new set of paths in a capacity-constrained network after failures to maximize the amount of restored flows, the respective linear programming formulation of the problem (LP) is necessary [40]. However, due to its \mathcal{NP} -completeness (see, e.g., [43]), the optimal solution can be found in a reasonable time using offline approaches only for small problem instances (e.g., for networks up to 12–15 nodes). Therefore, in the proposed method, calculating the alternate paths (Steps 9.2 and 10 in Fig. 9.4) is done using the heuristic approach based on Dijkstra's algorithm [15] that is proved to have the polynomial computational complexity bounded in above by $O(|N|^2)$, where $|N|$ is the number of WMN nodes.

9.1.4 Analysis of Modeling Results and Conclusions

In this section, we evaluate the vulnerability of five example WMNs to region failures (i.e., N29, N29_2, N29_3, N44, and N59 networks from Fig. 9.5), utilizing the proposed survivability measures. The first three networks (presented in Fig. 9.5a–c) are formed by 29 nodes located in $4000 \times 10,000 \text{ m}^2$, $6000 \times 6000 \text{ m}^2$, and $8000 \times 8000 \text{ m}^2$ fields, respectively, connected by 68, 68, and 57 wireless links, respectively. The other two networks shown in Fig. 9.5d–e consist of 44 and 59 nodes (located in fields of $10,000 \times 10,000 \text{ m}^2$), respectively, connected by 97 and 150 wireless links, respectively.

It is worth noting that for the N29 network, due to visible differences between horizontal and vertical sizes of the rectangular area (4000 m and 10,000 m, respectively), this network is likely to obtain the worst results concerning the portion of

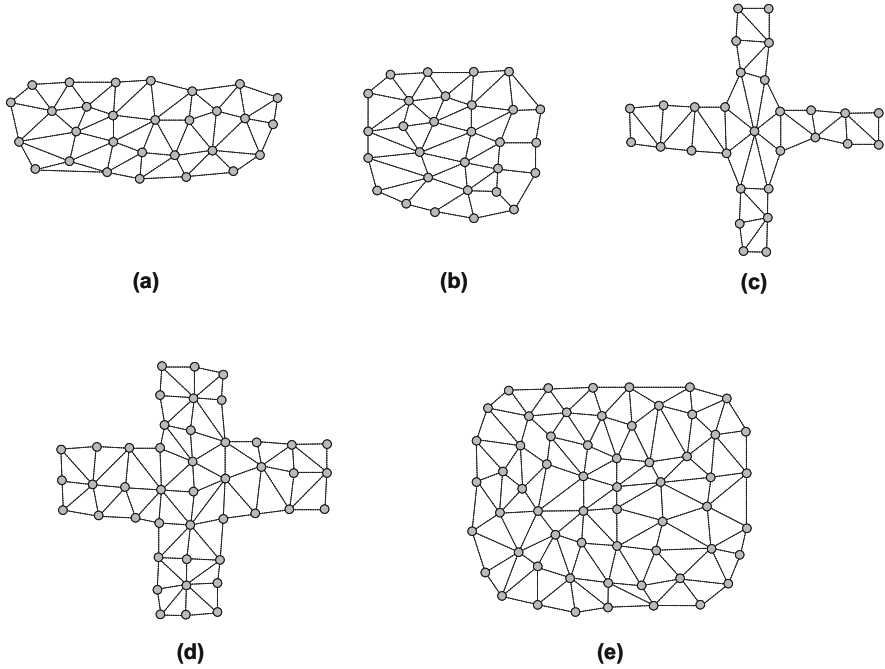


Fig. 9.5 Evaluated topologies of: N29 (a); N29_2 (b); N29_3 (c); N44 (d); and N59 (e) networks

flows surviving the regional failures (since for each network, the analyzed radiuses \hat{r} of failure regions were up to half of the largest Euclidean distance between any two nodes in the network).

When assessing the vulnerability of network flows to region disruptions, all transmission paths (both before and after failures) were calculated as the cheapest ones using the standard metric of distance [34, 41]. After failures, a reactive approach was utilized to redirect flows with survived end nodes. To provide the appropriate statistical analysis related to RFS, PFRS, and EPFD functions, the original values of $F[\psi]$ were obtained as the aggregate ones, including all 100 investigated demand sets of a certain size. For each set of demands, failures related to $FR = 9000$ random regions were simulated.

Three simulation scenarios were considered. The first two, referred to as Scenarios A and B, were prepared to use the proposed measures to evaluate the characteristics of different WMNs under a similar network load. To achieve this, the sets of unicast transmission demands included 25% of randomly chosen node pairs. Scenario A was to verify characteristics of WMNs of the same size in terms of the number of nodes (i.e., N29, N29_2, and N29_3 networks consisting of 29 nodes), while Scenario B was aimed at evaluating networks covering a similar area (i.e., not necessarily comparable in terms of the number of nodes). Therefore, topologies analyzed in Scenario B included N29, N44, and N59.

Additional Scenario C was to verify the properties of our measures under differentiated loads of the N59 network. In particular, four sizes of demand sets (i.e., consisting of randomly chosen 25%, 50%, 75%, and 100% node pairs) were examined. The capacity c_r of each unicast demand d_r was assumed to be unitary.

Each network link offered 160 units of unitary capacity in each direction. Considering failure scenarios, radiuses \hat{r} of failure regions were uniformly distributed in range $(0, \hat{r}_{\max})$, where \hat{r}_{\max} was equal to half of the largest Euclidean distance between any two network nodes. Statistical analysis of results was based on 95% confidence intervals. However, since the sizes of obtained intervals did not exceed 1% of the original values due to low visibility, they are not shown in Figs. 9.6–9.12.

Region Failure Survivability (RFS)

Evaluation of the vulnerability of WMN topologies to regional failures using the RFS measure under the assumptions of Scenario A is presented in Figs. 9.6 and 9.7. Recall that the RFS measure, defined in Eq. 9.5, was introduced to evaluate the probability that at least ψ percent of flows survive after a regional failure.

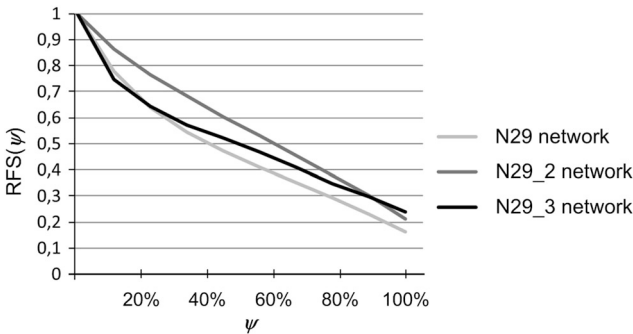


Fig. 9.6 RFS(ψ) function (Scenario A)

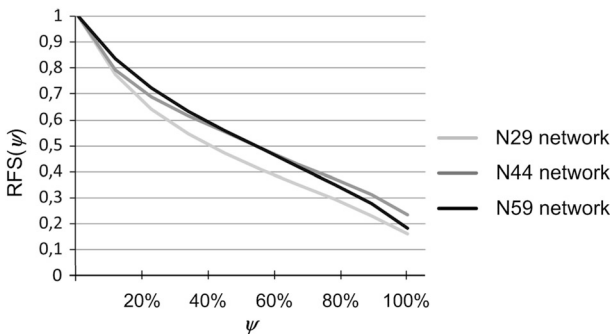


Fig. 9.7 RFS(ψ) function (Scenario B)

As presented in Figs. 9.6 and 9.7, with the increase of ψ , RFS starts decaying from the value of 1 (since, independent of the network topology, the probability of reducing the total flow to at least 0% is equal to 1). When comparing RFS characteristics for any two network topologies, greater values of RFS for any value of ψ imply a better network performance after a failure (since they reflect a greater chance of total flow reduction to at least ψ percent after a failure).

The general conclusion that follows from Figs. 9.6 and 9.7 is that better results concerning network survivability characteristics under regional failures are attributed to WMN networks with RFS functions driven by a slower decay with the increase of ψ (i.e., for which independent of ψ parameter, RFS values are higher). For instance, as shown in Fig. 9.6, the N29 network (for which its horizontal and vertical sizes are remarkably different) is outperformed by the N29_2 and N29_3 networks (located inside a square area) in Scenario A. In the same way, the N44 and N59 networks turned out to outperform the N29 network in Scenario B (Fig. 9.7).

***p*-Fractile Region Survivability (PFRS)**

Figures 9.8 and 9.9 show evaluation of WMN survivability characteristics using the *p*-fractile region survivability (PFRS) measure for Scenarios A and B. Recall that PFRS (Eq. 9.6) is to provide information on probability *p* that the fraction of total flow delivered after regional failures will not exceed ψ (Y axis on Figs. 9.8 and 9.9).

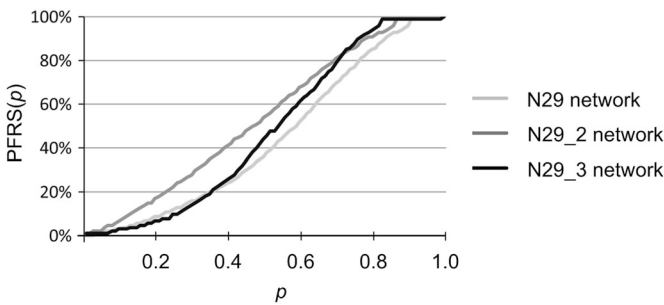


Fig. 9.8 PFRS(*p*) function (Scenario A)

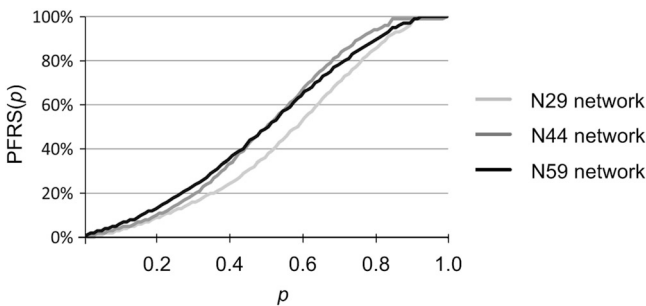


Fig. 9.9 PFRS(*p*) function (Scenario B)

For any WMN, it is thus better if, for any value of p , the upper bound on the portion ψ of flow surviving the failure is higher. As shown in Figs. 9.8 and 9.9, independent of the network topology, PFRS values are consistently positively correlated with p . Generally, the lower the values of PFRS, the more vulnerable the network is to regional failures. Similar to results for the RFS measure, PFRS also showed that the N29 network has the worst properties among all analyzed WMNs in Scenarios A–B.

EPFD Function

Figures 9.10 and 9.11 show values of EPFD function obtained in Scenarios A and B. Recall that EPFD function is defined by formula (9.7) as the expected percentage of the total flow delivered after failures occurring in circular areas of a certain radius \hat{r} . For any radius \hat{r} , greater values of the EPFD function imply more network flows surviving the failures. As shown in Figs. 9.10 and 9.11, the N29 network obtained the worst characteristics also concerning EPFD measure (which is compliant with the respective RFS and PFRS characteristics from Figs. 9.6–9.9, respectively).

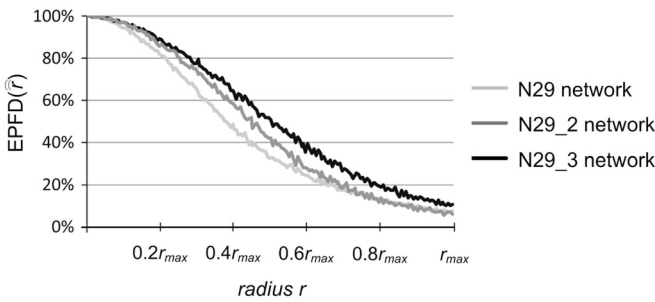


Fig. 9.10 EPFD(\hat{r}) function (Scenario A)

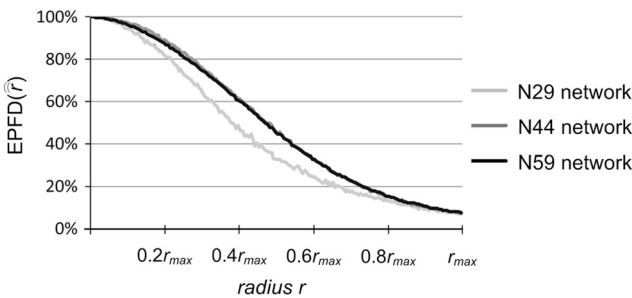


Fig. 9.11 EPFD(\hat{r}) function (Scenario B)

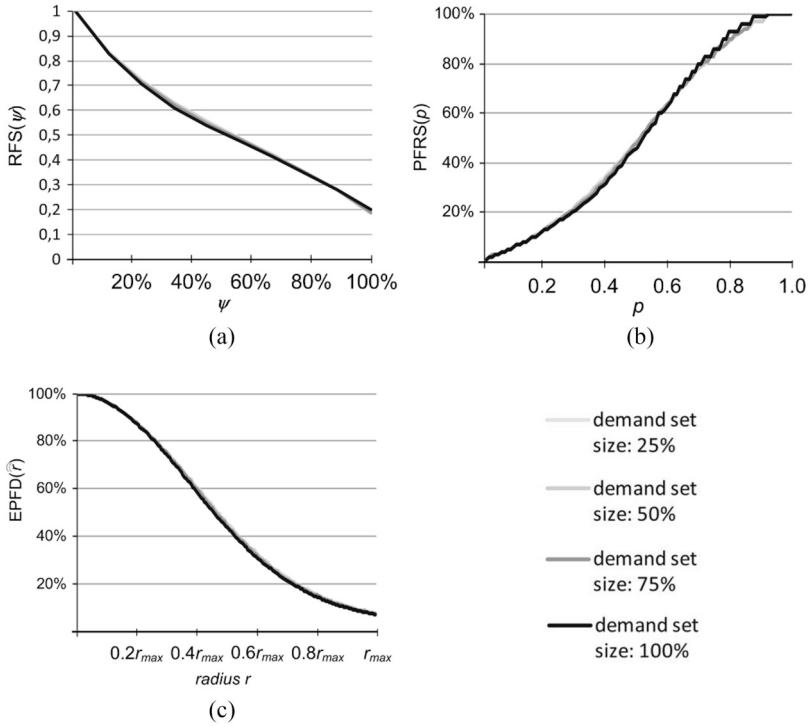


Fig. 9.12 Characteristics of (a) RFS(ψ), (b) PFRS(p), and (c) EPFD(\hat{r}) functions for Scenario C (N59 network)

It is worth mentioning that none of the three measures depends on the network load (as shown in Fig. 9.12 for Scenario C). Therefore, they can be used to compare the characteristics of different WMN topologies.

In this section, we focused on evaluating the vulnerability of WMNs to region failures occurring in circular areas and introduced three measures for evaluating WMN survivability. The first two measures, i.e., region failure survivability function—RFS and p -fractile region survivability function—PFRS, were proposed to assess WMN vulnerability to regional failures independent of the radius \hat{r} of the failure region. The third measure—the expected percentage of total flow delivered after a region failure as a function of region radius \hat{r} (EPFD)—was, in turn, designed to evaluate WMN performance depending on the radius \hat{r} of a circular failure region.

Proposed measures were later utilized to evaluate the properties of three example topologies of WMNs. Simulation analysis confirmed that these measures provide adequate and consistent information on the vulnerability of WMN networks to regional failures. Since for all introduced measures, achieved characteristics did not depend on the network load, they can thus be utilized in comparisons of different WMNs.

9.2 A New Approach to the Design of Weather Disruption-Tolerant Wireless Mesh Networks

As discussed in the former part of this chapter, failures of WMN nodes/links may imply severe data losses. In this section, we focus on link failures and present the respective approach to survivable routing to improve the WMN performance under link failures. As stated in [57], WMN links are susceptible to weather disruptions, particularly precipitation. Heavy rain storms may cause high signal attenuation, remarkably reducing the available link capacity or implying a link failure, leading to instability problems of routing (i.e., route flapping).

The issue of survivable routing is well-researched concerning wired networks (see, e.g., [47, 55, 58, 61, 67]), in particular concerning the protection of WDM network flows ([47, 55, 56, 60]). Among a few proposals on routing resilience in wireless networks, we can mention reference [10] addressing shared medium problems and node mobility issues. However, these solutions cannot be directly applied to WMNs due to remarkably different characteristics. In particular, WMNs are commonly nonmobile and do not encounter contention problems (if equipped with directional antennas). Therefore, except for link stability issues, WMNs seem to share the most important characteristics with wired networks [27].

To protect flows against weather-based disruptions of WMN links, it seems reasonable to use information related to expected incoming rain storms (e.g., achieved from radar echo measurements) to predict the real shapes of signal attenuation regions. Based on this idea, two approaches were introduced in [27], namely XL-OSPF and P-WARP, to modify the link-state OSPF routing based on weather predictions. Both techniques utilize formulas (9.9)–(9.10) from [14], defining the dependency of signal attenuation on the rain rate.

$$\Omega(R_p, \Theta) = \alpha R_p^\beta \left[\frac{e^{u\beta\vartheta} - 1}{u\beta} - \frac{b^\beta e^{i\beta\vartheta}}{i\beta} + \frac{b^\beta e^{i\beta\Theta}}{i\beta} \right], \quad \vartheta \leq \Theta \leq 22.5 \text{ km} \quad (9.9)$$

$$\Omega(R_p, \Theta) = \alpha R_p^\beta \left[\frac{e^{u\beta\Theta} - 1}{u\beta} \right], \quad 0 \leq \Theta \leq \vartheta \quad (9.10)$$

where:

- Ω is the signal attenuation in dB;
- Θ is the length of the path over which the rain is observed;
- R_p is the rain rate in mm/h;
- α, β are the numerical constants from [14].

$$u = \frac{\ln(b e^{i\beta\vartheta})}{\beta}, \quad b = 2.3 R_p^{-0.17}.$$

$$i = 0.026 - 0.03 \ln R_p, \quad \vartheta = 3.8 - 0.6 \ln R_p.$$

In particular, XL-OSPF utilizes a special metric of link cost being proportional to the observed bit error rate (BER) of the link (which is justifiable due to the clear

impact of signal attenuation on the effective BER, as well as on packet error rate—PER). This metric is utilized reactively to update the OSPF routing characteristics. However, such an approach is not straightforward to deploy since, in the Media Access Control (MAC) layer, there is no information on the actual BER between network nodes (it can be estimated using signal-to-noise ratio—SNR).

P-WARP, in turn, estimates the costs of WMN links using weather-based predictions of future conditions of links. This can be done at either one dedicated node or a subset of nodes capable of collecting the weather-related radar data.

In this section, we focus on reducing the signal attenuation level along millimeter-wave links in the presence of rain storms. In particular, in Sect. 9.2.1, we present in detail our method from [44] to perform in advance the periodic updates of a WMN topology following forecasts of heavy rain storms, using the functionality of a dynamic antenna alignment offered by several equipment vendors (see, e.g., [54]). Next, in Sect. 9.2.2, we describe the ILP model we proposed to obtain the optimal routing solution per the forecasted levels of signal attenuation at WMN links (that also returns the proper assignment of non-interfering channels to intersecting links). After that, in Sect. 9.2.3, we present the analysis of the problem's computational complexity, followed by an evaluation of our approach characteristics (Sect. 9.2.4).

To the best of our knowledge, the protection of WMN links against weather-based regional failures has not been sufficiently researched so far. In particular, there is no other proactive approach that is based on periodic updates of a WMN topology.

9.2.1 Proposed Approach

The technique to protect WMN links against weather-based disruptions described here does not impose any modifications to the routing algorithm. Therefore, it can be used with practically any routing scheme, making our solution easily deployable. In particular, transmission paths are established based on conventional metric of link costs (e.g., the number of hops).

The main idea of our approach is to prepare the network for changing weather conditions by applying the periodic updates of WMN topology to improve the throughput during rain storms. We propose to perform consecutive updates of a WMN topology by employing dynamic antenna alignment features (offered by several equipment vendors) utilizing predictions related to future conditions of WMN links based on rain storm forecasts obtained from real echo rain maps. This, in turn, implies periodic creation (or deletion) of WMN links if low (or high) values of signal attenuation are expected for them, respectively.

The network is modeled in this section by graph $G=(N, A)$, similar to Sect. 9.1.1. In particular, any link between two neighboring nodes, i and j , is represented by two directed arcs $a_h=(i, j)$ and $a_{h'}=(j, i)$, respectively, and is assigned a given transmission channel from the set of available transmission channels. To focus on time-varying characteristics of WMN links, the definition of graph G is extended by:

- \check{T} denoting the lifetime of a network.
- $\vartheta(\check{T}): A \times \check{T} \rightarrow \{0, 1\}$ function determining the existence of links at time $t \in \check{T}$.
- $\gamma(\check{T}): A \times \check{T} \rightarrow \mathcal{R}$ link cost function based on signal attenuation ratio at time $t \in \check{T}$ (formulas (9.9)–(9.10)).

We assume the existence of a dedicated core node responsible for the alignment of antennas of all network nodes that has access to:

- The set of active network nodes and their locations
- Radar echo rain measurements (received periodically)
- Demands to provide transmission between WMN end nodes

The role of this core node is also to execute the procedure shown in Fig. 9.13. In particular, in Step 1 of this scheme, the estimated signal attenuation ω_h at each potential arc $a_h=(i, j)$ is determined using formulas (9.9)–(9.10). The action of Step 2 is to return a new configuration of WMN links. In particular, in the proposed scheme, ω_h values are used as link costs to obtain the set of the cheapest (in terms of signal attenuation) potential paths. If, in Step 2, a given link is not used by any path, it will not be present in the updated WMN topology.

In the method from Fig. 9.13, we propose to utilize the heuristic approach to proceed with Step 2, since the problem to determine the optimal alignment of WMN antennas with the objective to minimize the aggregate signal attenuation over all transmission paths, defined in Sect. 9.2.2, is \mathcal{NP} -complete (as proved in Sect. 9.2.3). New alignment of antennas (Step 3) is expected every τ time units (as defined in Step 4).

INPUT	
–	set of network nodes N , each node i characterized by its coordinates (\bar{x}_i, \bar{y}_i) ,
–	initial set of WMN links extended by possible links between each pair of neighboring nodes,
–	frequency of antenna alignment updates defined by interval τ ,
–	current radar echo rain measurements,
–	aggregate demand volumes for each pair of nodes s_r and t_r of r -th demand
OUTPUT	
	Updated alignment of antennas corresponding to the forecasted level of signal attenuation based on rain storm predictions
Step 1	For each pair of neighboring nodes i and j , determine signal attenuation ω_h of arc $a_h=(i, j)$ to be potentially installed between nodes i and j based on the forecasted radar rain information.
Step 2	Determine a new configuration of links based on estimated values of signal attenuation from Step 1. For this purpose, for each demand r to provide transmission between nodes s_r and t_r , find the cheapest transmission path in terms of costs ω_h calculated in Step 1.
Step 3	Distribute the results of Step 2 to all network nodes to set the alignment of WMN antennas.
Step 4	Wait τ units of time and go to Step 1.

Fig. 9.13 Proposed methodology of periodic updates of alignment of WMN antennas

It is worth recalling that metric ω_h is used in our approach only to update the alignment of antennas at WMN nodes. Routing is, in turn, performed using a conventional protocol with all its characteristics unchanged. This implies that the original metric of link costs (i.e., the one normally used by the routing algorithm) is utilized instead of ω_h values to obtain the real transmission paths.

9.2.2 ILP Formulation of Weather-Resistant Links Formation Problem (WRLFP)

The problem of determining the optimal alignment of WMN antennas (Step 2 from Fig. 9.13) to minimize the aggregate signal attenuation over all transmission paths at time t can be solved by determining the solution to the following ILP model.

Symbols

$G(N,A)$	Graph representing a directed network.
N	Set of network nodes; $ N $ is the number of network nodes.
A	Set of directed arcs; $ A $ is the number of arcs.
h	Arc index; $h = 1, 2, \dots, A $.
D	Set of demands; $ D $ is the number of demands.
r	Demand index; $r = 1, 2, \dots, D $.
L_h	Set of transmission channels available at arc $a_h = (i, j)$.
$1 \dots \Lambda_h$	Indices of transmission channels at arc $a_h = (i, j)$; $\forall_h \Lambda_h = \Lambda$.

Constants

$s_r(t_r)$	Source (destination) node of r -th demand.
c_r	Capacity of r -th demand.
$c_h(t)$	Estimated total capacity of arc $a_h = (i, j)$ at time t .
$\omega_h(t)$	Estimated signal attenuation due to rain falls for arc $a_h = (i, j)$ at time t .

Variables

$x_{r,h}^l$	Equals 1, if l -th channel is assigned for r -th demand path at arc $a_h = (i, j)$; 0 otherwise.
-------------	---

Objective

It is to find the end-to-end transmission paths for all demands, minimizing the cost defined by formula (9.11).

$$\min \varphi(x, t) = \sum_{r \in D} \sum_{l \in L_h} \sum_{h \in A} \omega_h(t) \cdot x_{r,h}^l \quad (9.11)$$

where $\omega_h(t)$ is the cost of arc $a_h = (i, j)$ based on signal attenuation ratio at time t .

Constraints

1. Flow conservation rules (based on Kirchhoff's law) for end-to-end paths:

$$\sum_{l \in L_h} \sum_{\substack{h \in \{h: a_h = (n, j) \in A; \\ j = 1, 2, \dots, |N|; j \neq n\}}} x_{r,h}^l - \sum_{l \in L_h} \sum_{\substack{h \in \{h: a_h = (i, n) \in A; \\ i = 1, 2, \dots, |N|; i \neq n\}}} x_{r,h}^l = \begin{cases} 1, & \text{if } n = s_r \\ -1, & \text{if } n = t_r \\ 0, & \text{otherwise} \end{cases} \quad (9.12)$$

where:

$a_h = (n, j)$ denotes an arc incident out of node n ; $a_h = (i, n)$ refers to an arc incident into node n ; $r = 1, 2, \dots, |D|$; $n = 1, 2, \dots, |N|$.

2. On the finite capacity of arcs a_h (i.e., to assure that the total flow assigned to arc a_h will not exceed the maximum available capacity):

$$\sum_{l \in L_h} \sum_{r \in D} x_{r,h}^l \cdot c_r \leq c_h(t); \quad h \in A \quad (9.13)$$

3. On the selection of different channels to interfering links (at most one link from the set of interfering links can be assigned a given channel l):

$$\sum_{r \in D} x_{r,h}^l + \sum_{r \in D} x_{r,h'}^l \leq 1 \quad (9.14)$$

for each pair of intersecting arcs a_h and $a_{h'}$; $l \in L_h$.

9.2.3 Computational Complexity of WRLFP Problem

This section discusses the complexity of the considered optimization problem (9.11)–(9.14). In particular, by proving that it belongs to the class of \mathcal{NP} -complete problems (by showing that one of its subproblems being the channel assignment problem, referred to as WR_CAP , is \mathcal{NP} -complete), we explain that there is no efficient algorithm proposed so far to find the optimal solution in polynomial time.

Since the assignment of channels to links is confined to the set of Λ available channels (where Λ can be any arbitrarily chosen small integer value), the optimization version of the WR_CAP channel allocation subproblem can be defined as follows.

WR_CAPopt(A'):

Given the set of network arcs A' utilized by paths in Step 2 from Fig. 9.13, find the optimal assignment of transmission channels to arcs a_h minimizing the number

of used channels, providing that none of the intersecting arcs receives the same channel.

To show the \mathcal{NP} -completeness of WR_CAP, it is sufficient to analyze its recognition version (i.e., a problem with a “yes/no” answer) [28] shown below.

WR_CAPrec(A', k):

Given a set of arcs A' utilized by paths in Step 2 from Fig. 9.13, is it possible to find the optimal assignment of channels to arcs a_h in the network that requires k different channels, providing that none of the intersecting arcs receives the same channel?

If the recognition version of the problem is \mathcal{NP} -complete, so is its optimization version [2].

Theorem: WR_CAP problem is \mathcal{NP} -complete.

Proof Following [2], when proving the \mathcal{NP} -completeness of the WR_CAP problem, it is sufficient to show that:

- (a) WR_CAPrec(A', k) belongs to the class of \mathcal{NP} problems.
- (b) A known \mathcal{NP} -complete problem polynomially reduces to WR_CAPrec(A', k).

Regarding (a): WR_CAP problem belongs to complexity class \mathcal{NP} since it can be determined in polynomial time whether a given assignment of transmission channels to arcs a_h is valid (i.e., whether it requires exactly k channels from the set $\{1, \dots, |\Lambda|\}$). In particular, checking the assignment of channels can be done in at most $O(|A'|) = O(|n^2|)$ operations, while verifying whether different channels are assigned to intersecting links requires at most $O(|n^2|)$ steps.

Regarding (b): To provide the second part of the proof, we will show that the known \mathcal{NP} -complete problem of determining the optimal vertex-coloring of a graph of conflicts Γ [28], here referred to as VCGC, can be transformed in polynomial time into WR_CAP problem. As shown in [28], the recognition version of the VCGC problem can be defined in the following way.

VCGCrec(Γ, k):

Given a graph of conflicts $\Gamma = (V, E)$, where V is the set of vertices, and E is the set of edges $e_h = (i, j)$ representing conflicts between the respective vertices i and j , is it possible to find the optimal assignment of colors to vertices from V requiring exactly k colors in a way that any two conflicting vertices i and j (i.e., connected by an edge in Γ) receive different colors?

Assume that:

- $\{\Gamma = (V, E), k\}$ is the input to the VCGC recognition instance of the problem.
- Γ also represents the graph of conflicts for links to be installed in the network after executing Step 2 of the method from Fig. 9.13. In this graph:
 - * Vertices from V represent links to be installed in the network.
 - * There exists edge $e_h = (j, k)$ in Γ if the respective network arcs a_j and a_k in G intersect with each other, i.e., if they have to be assigned different channels.

- (\rightarrow) Let us assume that it is feasible to color vertices from Γ using k different colors. In this case, any valid coloring of Γ by k different colors in $\text{VCGCrec}(\Gamma, k)$ automatically returns a proper assignment of k different channels to interfering links in $\text{WR_CAPrec}(A', k)$.
- (\leftarrow) Assume that k channels are sufficient to solve the $\text{WR_CAPrec}(A', k)$ problem. Then, after creating the respective graph of conflicts Γ for interfering WMN links, we automatically have a valid coloring of Γ vertices that requires k different colors. ■

If we relax the problem by disregarding the requirement for allocation of different channels to intersecting links, the simplified problem remains \mathcal{NP} -complete as a basic task to determine transmission paths between $|D|$ pairs of nodes in capacity-constrained networks (classified as \mathcal{NP} -complete in [43]). Therefore, to perform Step 2 from Fig. 9.13, the heuristic Dijkstra's algorithm from [15] is used.

Example Execution Steps of the Proposed Method

Results of a single iteration of the proposed method execution are presented in Fig. 9.14. The initial alignment of antennas is shown in Fig. 9.14a. Based on actual information related to the predicted rain intensity from Fig. 9.14b, a single iteration of our procedure is to update the network topology necessary to prepare the network for the forthcoming rain.

For this purpose, the WMN topology is first extended by the respective core node (responsible for determining the updates of a network topology) to include all possible links between neighboring nodes (see Fig. 9.14b). A new alignment of antennas is next determined based on the forecasted attenuation of a signal along each potential link (see Fig. 9.14c). As a result, the updated topology from Fig. 9.14c

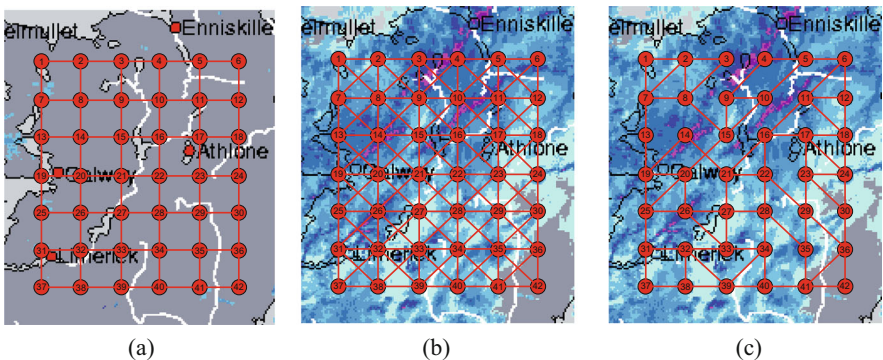


Fig. 9.14 Example execution steps of our procedure to modify the network topology (here the artificial Irish Network) according to the current rain storm forecasts including: (a) initial topology of a network; (b) extended topology including all possible links; (c) results of the algorithm execution

does not include links located within heavy rain storm areas (e.g., links (3, 4), (10, 11), (14, 15), and (15, 16)).

9.2.4 Analysis of Modeling Results and Conclusions

Simulations were performed to verify the characteristics of our approach for two examples of artificial WMN topologies from Fig. 9.15, located in the area of Southern England and Ireland, respectively. The topology of each network included 42 nodes and formed a grid structure with link lengths equal to 15 km.

Characteristics of our technique (here referred to as “with protection”) were compared with the common one, implying no changes in the alignment of antennas (further referred to as the “no protection” case).

In the proposed technique, the initial set of WMN links included the ones marked with solid red lines in Fig. 9.15. Dashed blue lines are, in turn, used in Fig. 9.15 to indicate the extension of the set of links for possible utilization by the proposed technique. In the reference “no protection” approach, the set of links did not change over time (i.e., it was determined only by red lines from Fig. 9.15). In each network, nodes 1 and 42 were configured as gateways connecting the other nodes to the Internet. Traffic outgoing the network via one of these gateways was assumed to be generated by each WMN node at a rate of 3 Mb/s.

Simulations were focused on measuring the average signal attenuation ratio due to rain storms along transmission paths, as well as the average path hop count for three real scenarios of rain storms that occurred in November 2011:

- Scenario A: Southern England, Nov. 25, 2011, from 3:00 AM till 10:00 AM
- Scenario B: Ireland, November 26–27, 2011, from 8:00 PM till PM 7:00 AM
- Scenario C: Ireland, November 24, 2011, from 10:00 AM till 12:00 PM

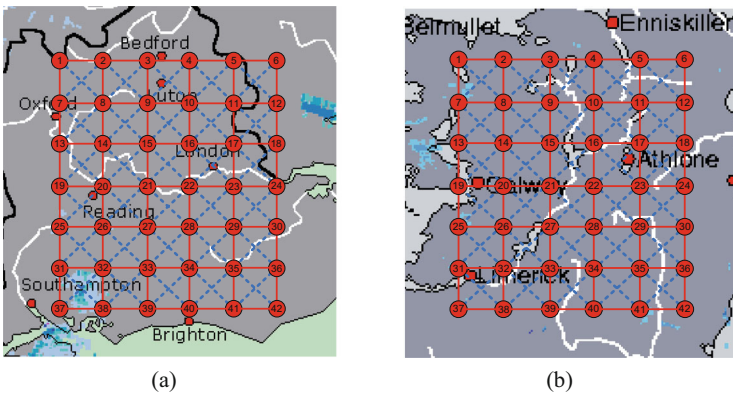


Fig. 9.15 Example topologies of WMNs used in simulations (a) Southern English Network; (b) Irish Network

Radar rain maps utilized in simulations were recorded every 15 minutes. The duration of the analyzed rain storms varied from 7 to 14 hours. A limited set of investigated rain maps (one map per hour) is shown in the Appendix (Sect. 9.2.5).

Signal Attenuation

As shown in Fig. 9.16, the signal attenuation level increased remarkably during heavy rain intervals. However, due to periodic updates of antenna alignment according to the forecasted signal attenuation ratio, our approach was able to prepare the WMN topology in advance for the forthcoming rain and, as a result, to significantly decrease the signal attenuation ratio (up to 90%, as shown in Fig. 9.16). A general conclusion is that the most significant improvement was observed for periods of heavy rain (which is a very desirable feature). On the contrary, in the case of light rains, updating the alignment of antennas implied only a slight reduction of the analyzed signal attenuation ratio.

Number of Path Links

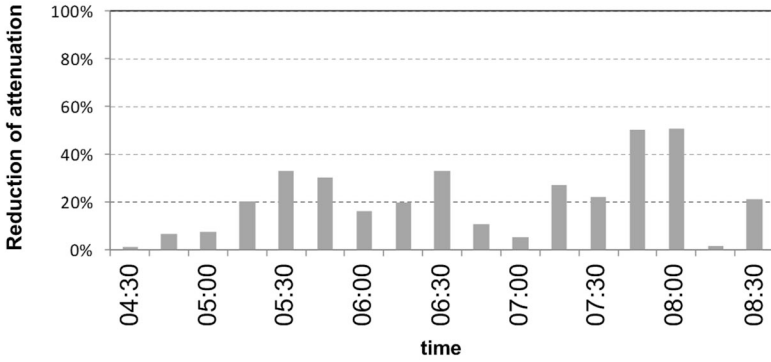
Considering the average hop count of end-to-end transmission paths, for the standard “no protection” method (for which the costs of links were independent of signal attenuation ratio), the average number of path links was equal to 5.6.

Due to the operations of WMN link creation/deletion being the implications of changing attenuation conditions, our technique resulted in establishing WMN links more elastically. In particular, this often implied forming diagonal links (e.g., between nodes 1 and 8), which, in general, resulted in shorter paths. As presented in Fig. 9.17, the average end-to-end hop count for our technique was often visibly lower than that for the reference approach. However, during heavy rain periods (Scenario B, 10:00 PM–1:00 AM; Scenario C, 4:00 PM–10:00 PM), the average hop count for our approach was higher due to the need to provide detours over heavy rain areas.

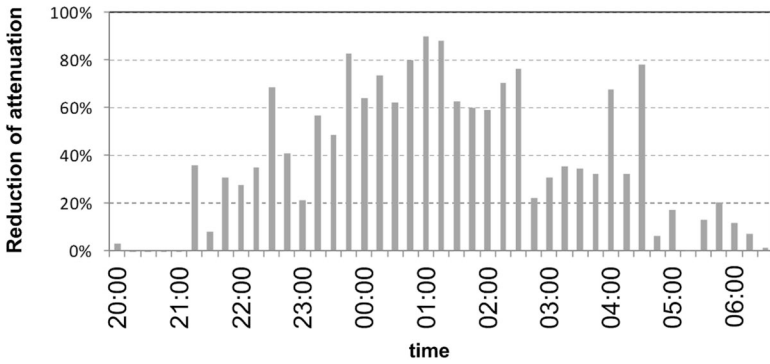
This section addressed the signal attenuation problem in WMNs due to heavy rain storms. To improve the network’s performance during rainy intervals, we presented a method to apply the periodic updates of a WMN topology that utilizes information from radar echo rain measurements in advance. Our approach can be easily implemented in practice, as dynamic antenna alignment functionality is available in several commercial products. Another advantage is that our approach does not imply any changes in a routing algorithm.

It was verified by simulations performed for real radar rain maps that the proposed technique can bring about a significant decrease (up to 90%) of signal attenuation, compared to the results of the reference “no protection” approach of not applying any changes to WMN topology. This improvement was observed for heavy rain periods (which is indeed a very desired feature).

Scenario A



Scenario B



Scenario C

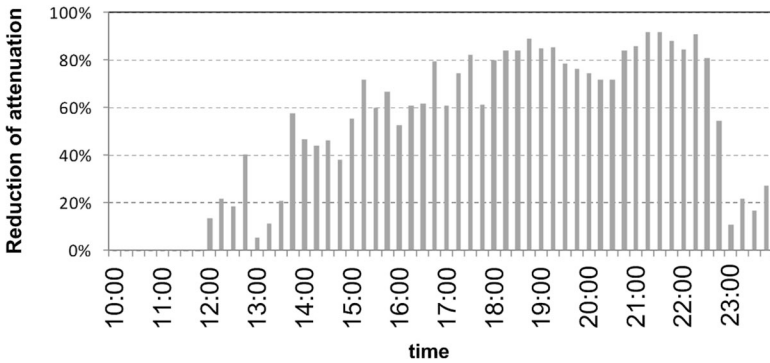
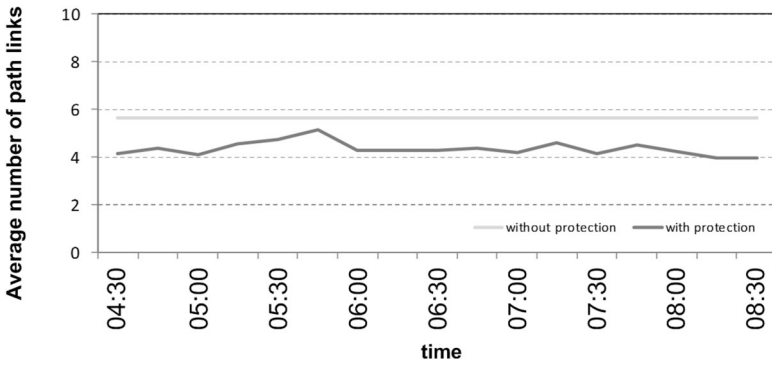
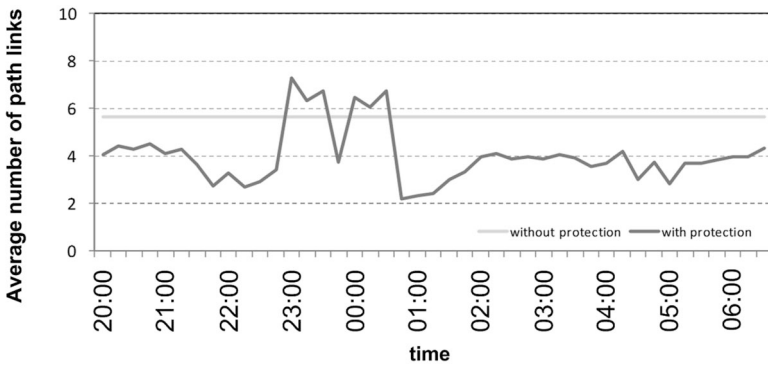


Fig. 9.16 Obtained results concerning reduction of signal attenuation

Scenario A



Scenario B



Scenario C

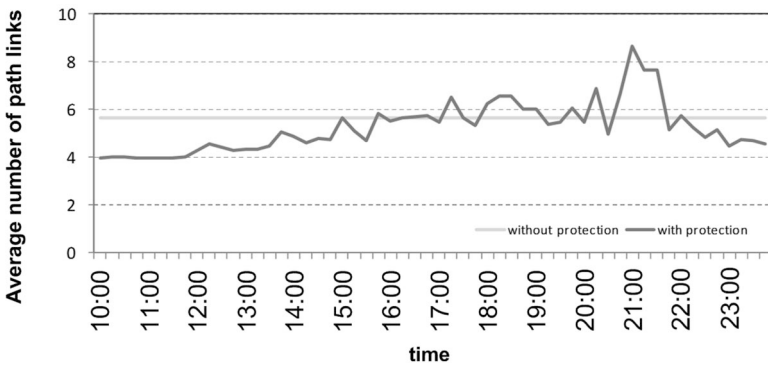
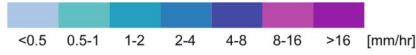


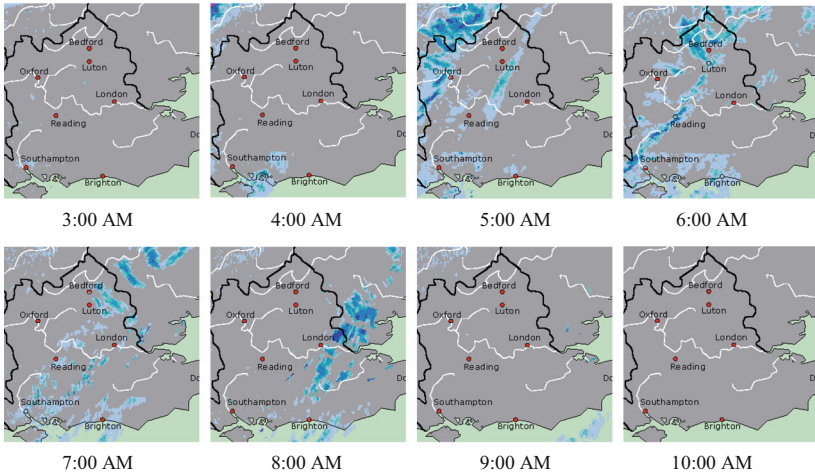
Fig. 9.17 Obtained results concerning the average hop count

9.2.5 Appendix—Rain Radar Maps Used in Simulations

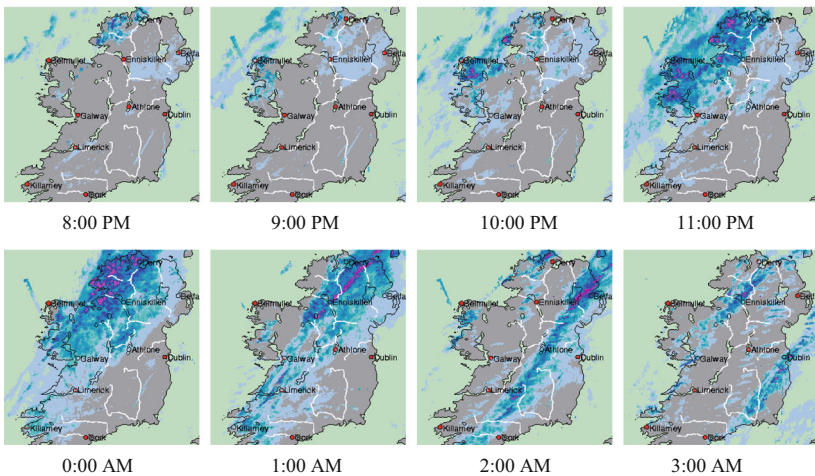
Radar rain maps used in Sect. 9.2 are presented in this Appendix in one-hour intervals (during simulations, rain maps were, however, collected every 15 min). Each map presented here provides information about the rain intensity following the intensity scale provided by www.weatheronline.com service.



Scenario A: Southern England, November 25, 2011



Scenario B: Ireland, November 26-27, 2011



9.3 Summary

As shown in this chapter, the resilience of WMNs is a challenging issue. In terms of resilient routing, WMNs seem to exhibit most characteristics commonly attributed to wired networks (e.g., stationary nodes, high capacity, or no limits on energy consumption), however, with a clear exception referring to the time-varying link stability. Due to high-frequency communications, the vulnerability of WMN links to weather-based disruptions is even more challenging than in conventional 802.11 architectures. That is why the direct application of resilience mechanisms originally designed for pure wired or ad hoc (wireless) networks is improper.

As shown in this chapter, the number of proposals addressing the resilient routing issue in WMNs is limited. They include, e.g., routing metrics updates to keep changing the communication paths reactively as a response to time-varying characteristics of WMN links. However, a general observation (following from research results on wired networks resilience) is that considering the extent of losses after failures, better results would be achieved when applying the proactive approach (implying preparation of an alternate transmission solution in advance—before the occurrence of a failure). Additionally, no survivability measures have been proposed so far to evaluate the WMN performance for a common scenario of regional failures (implied, e.g., by weather-based region disruptions).

To address these issues, the respective survivability measures have been proposed in this chapter to allow for evaluation of a WMN performance under region failures leading to massive failures of WMN nodes/links. The unique characteristics of WMN links also made us propose a transmission scheme that can prepare the network in advance for the forthcoming heavy rain using automatic antenna alignment features. As a result, due to information from radar echo rain maps, settings of WMN antennas could be proactively updated to create links omitting areas of predicted heavy rain (which reduced the signal attenuation ratio up to 90%).

It seems that other resilience approaches proposed for wired networks, e.g., based on multiple alternate paths, could also be applied to WMNs after adapting them to the characteristics of WMN links. This is a vast area for future research.

References

1. Agarwal, P.K., Efrat, A., Ganjugunte, S.K., Hay, D., Sankararaman, S., Zussman, G.: Network vulnerability to single, multiple and probabilistic physical attacks. In: Proceedings of the Military Communications Conference (MILCOM'10), pp. 1824–1829 (2010)
2. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows: Theory, Algorithms, and Applications. Prentice Hall, Englewood Cliffs (1993)
3. Akyildiz, I.F., Wang, X., Wang, W.: Wireless Mesh Networks: a survey. *Comput. Networks* **47**(7), 445–487 (2005)
4. Aruba Networks: <http://www.arubanetworks.com/>. Accessed on 24 Nov 2014

5. Avallone, S., Akyildiz, I.F., Giorgio, V.: A channel and rate assignment algorithm and a layer-2.5 forwarding paradigm for multi-radio wireless mesh networks. *IEEE/ACM Trans. Networking* **17**(1), 267–280 (2009)
6. Balbuena, M.C., Carmona, A., Fiol, M.A.: Distance connectivity in graphs and digraphs. *J. Graph Theory* **22**(4), 281–292 (1998)
7. Beineke, L.W., Oellermann, O.R., Pipperta, R.E.: The average connectivity of a graph. *Discrete Math.* **252**(1–3), 31–45 (2002)
8. Benyamina, D., Hafid, A., Gendreau, M.: Wireless Mesh Networks design – a survey. *IEEE Commun. Surv. Tutorials* **14**(2), 299–310 (2012)
9. Biswas, S., Morris, R.: ExOR: opportunistic multi-hop routing for wireless networks. *SIGCOMM Comput. Commun. Rev.* **35**(4), 133–144 (2005)
10. Campista, M.E.M., Esposito, P.M., Moraes, I.M., Costa, L.H.M.K., Duarte, O.C.M.B., Passos, D.G., de Albuquerque, C.V.N., Saade, D.C.M., Rubinstein, M.G.: Routing metrics and protocols for wireless mesh networks. *IEEE Network* **22**(1), 6–12 (2008)
11. Capone, A., Carello, G., Filippini, I., Gualandi, S., Malucelli, F.: Routing, scheduling and channel assignment in Wireless Mesh Networks: optimization models and algorithms. *Ad Hoc Networks* **8**(6), 545–563 (2010)
12. Couto, D.S.J.D., Aguayo, D., Bicket, J., Morris, R.: A high throughput path metric for multi-hop wireless routing. In: *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom'03)*, pp. 134–146 (2003)
13. Couto, D.S.J.D., Aguayo, D., Chambers, A., Morris, R.: Performance of multihop wireless networks: shortest path is not enough. *SIGCOMM Comput. Commun. Rev.* **33**(1), 83–88 (2003)
14. Crane, R.: Prediction of attenuation by rain. *IEEE Trans. Commun.* **28**(9), 1717–1733 (1980)
15. Dijkstra, E.: A note on two problems in connexion with graphs. *Numer. Math.* **1**, 269–271 (1959)
16. Draves, R., Padhye, J., Zill, B.: Routing in multi-radio, multi-hop wireless mesh network. In: *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom'04)*, pp. 114–128 (2004)
17. Efstathiou, E.C., Frangoudis, P.A., Polyzos, G.C.: Stimulating participation in wireless community networks. In: *Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM'06)*, pp. 1–13 (2006)
18. Gabale, V., Raman, B., Dutta, P., Kalyanraman, S.: A classification framework for scheduling algorithms in Wireless Mesh Networks. *IEEE Commun. Surv. Tutorials* **15**(1), 199–222 (2013)
19. Ganjali, Y., Keshavarzian, A.: Load balancing in ad hoc networks: single-path routing vs. multi-path routing. In: *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM'04)*, vol. 2, pp. 1120–1125 (2004)
20. Gass, R., Diot, C.: Measurements of in-motion 802.11 networking. In: *Proceedings of the 7th IEEE Workshop on Mobile Computing Systems and Applications (HotMobile'06)*, pp. 69–74 (2006)
21. Ghazisaidi, N., Scheutzw, M., Maier, M.: Survivability analysis of Next-Generation Passive Optical Networks and fiber-wireless access networks. *IEEE Trans. Reliab.* **60**(2), 479–492 (2011)
22. Gore, D.A., Karandikar, A.: Link scheduling algorithms for Wireless Mesh Networks. *IEEE Commun. Surv. Tutorials* **13**(2), 258–273 (2011)
23. Guo, Y.: Path connectivity in local tournaments. *Discrete Math.* **167**(168), 353–372 (1997)
24. Henderson, T., Kotz, D., Abyzov, I.: The changing usage of a mature campus-wide wireless network. In: *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom'04)*, pp. 187–201 (2004)
25. Huang, S., Dutta, R.: Design of Wireless Mesh Networks under the additive interference model. In: *Proceedings of the IEEE International Conference on Computer Communications and Networks (ICCCN'06)*, pp. 253–260 (2006)
26. IEEE standards: <http://standards.ieee.org/findstds/standard/802.11s-2011.html>. Accessed on 11 Jan 2015

27. Jabbar, A., Rohrer, J.P., Oberthaler, A., Cetinkaya, E.K., Frost, V., Sterbenz, J.P.G.: Performance comparison of weather disruption-tolerant cross-layer routing algorithms. In: Proceedings of the 28th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM'09), pp. 1143–1151 (2009)
28. Karp, R.M.: Reducibility among combinatorial problems. *Complexity Comput. Comput.*, 85–103 (1972)
29. Khan, J.A., Alnuweiri, H.M.: Traffic engineering with distributed dynamic channel allocation in BFWA mesh networks at millimeter wave band. In: Proceedings of the 14th IEEE Workshop on Local and Metropolitan Area Networks (IEEE LANMAN'05), pp. 1–6 (2005)
30. Kim, K., Venkatasabramanian, N.: Assessing the impact of geographically correlated failures on overlay-based data dissemination. In: Proceedings of the IEEE Global Communications Conference (IEEE Globecom'10), pp. 1–5 (2010)
31. Kodialam, M., Nandagopal, T.: Characterizing the capacity region in multi-radio multi-channel Wireless Mesh Network. In: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking (MIBICOM'05), pp. 73–87 (2005)
32. Kyasnanur, P., Vaidya, N.H.: Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks. *SIGMOBILE Mob. Comput. Commun. Rev.* **10**(1), 31–43 (2006)
33. Lee, S., Bhattacharjee, B., Banerjee, S.: Efficient geographic routing in multihop wireless networks. In: Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'05), pp. 230–241 (2005)
34. Li, H., Cheng, Y., Zhou, Ch., Zhuang, W.: Routing metrics for minimizing end-to-end delay in multiradio multichannel wireless networks. *IEEE Trans. Parall. Distrib. Syst.* **24**(11), 2293–2303 (2013)
35. Liu, J., Jiang, X., Nishiyama, H., Kato, N.: Reliability assessment for wireless mesh networks under probabilistic region failure model. *IEEE Trans. Veh. Technol.* **60**(5), 2253–2264 (2011)
36. Molisz, W.: Survivability function – a measure of disaster-based routing performance. *IEEE J. Sel. Areas Commun.* **22**(9), 1876–1883 (2004)
37. Motorola: <http://wirelessnetworks-asia.motorola.com/>. Accessed on 11 Jan 2015
38. Neumayer, S., Modiano, E.: Network reliability with geographically correlated failures. In: Proceedings of the 29th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM'10), pp. 1–9 (2010)
39. Ohata, K., Maruhashi, K., Ito, M., Nishiumi, T.: Millimeter-wave broadband transceivers. *NEC J. Adv. Technol.* **2**(3), 211–216 (2005)
40. Papadimitriou, Ch.: *Computational Complexity*. Addison-Wesley, Boston (1994)
41. Paris, S., Nita-Rotaru, C., Martignon, F., Capone, A.: Cross-layer metrics for reliable routing in wireless mesh networks. *IEEE/ACM Trans. Networking* **21**(3), 1003–1016 (2013)
42. Pathak, P.H., Dutta, R.: A survey of network design problems and joint design approaches in Wireless Mesh Networks. *IEEE Commun. Surv. Tutorials* **13**(3), 396–428 (2011)
43. Pioro, M., Medhi, D.: *Routing, Flow and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann Publishers, Burlington (2004)
44. Rak, J.: A new approach to design of weather disruption-tolerant Wireless Mesh Networks. *Telecommun. Syst.* **61**, 311–323 (2015)
45. Rak, J.: Measures of region failure survivability for wireless mesh networks. *Wireless Netw.* **21**(2), 673–684 (2015)
46. Ramachandran, K., Buddhikot, M., Chandranmenon, G., Miller, S., Belding-Royer, E., Almeroth, K.: On the design and implementation of infrastructure mesh networks. In: Proceedings of the IEEE Workshop on Wireless Mesh Networks (WiMesh'05), pp. 1–12 (2005)
47. Ramamurthy, S., Sahasrabudde, L., Mukherjee, B.: Survivable WDM mesh networks. *IEEE/OSA J. Lightwave Technol.* **21**(4), 870–883 (2003)
48. Ramanathan, R., Steenstrup, M.: Hierarchically-organized multihop mobile wireless networks for quality-of-service support. *Mobile Networks Appl.* **3**(1), 101–119 (1998)
49. Robinson, J., Swaminathan, R., Knightly, E.W.: Assessment of urban-scale wireless network with a small number of measurements. In: Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MobiCom'08), pp. 187–198 (2008)

50. Sen, A., Shen, B.H., Zhou, L., Hao, B.: Fault-tolerance in sensor networks: A new evaluation metric. In: Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM'06), pp. 1–12 (2006)
51. Sen, A., Banerjee, S., Ghosh, P., Shirazipourazad, S.: Impact of region based faults on the connectivity of wireless networks: In: Proceedings of the 47th Allerton Conference on Communication, Control and Computing, pp. 1430–1437 (2009)
52. Sen, A., Murthy, S., Banerjee, S.: Region-based connectivity – a new paradigm for design of fault-tolerant networks. In: Proceedings of the 15th International Conference on High Performance Switching and Routing (HPSR'09), pp. 1–7 (2009)
53. Shengli, Y., Wang, B.: Highly available path routing in mesh networks under multiple link failures. *IEEE Trans. Reliab.* **60**(4), 823–832 (2011)
54. SkyPilot: http://skypilot.trilliantinc.com/pdf/broch_sp_products.pdf. Accessed on 09 Jan 2015
55. Somani, A.: *Survivability and Traffic Grooming in WDM Optical Networks*. Cambridge University Press, Cambridge (2006)
56. Soproni, P., Cinkler, T., Rak, J.: Methods for physical impairment constrained routing with selected protection in all-optical networks. *Telecommun. Syst.* **56**(1), 177–188 (2014)
57. Sterbenz, J.P.G., Cetinkaya, E.K., Hameed, M.A., Jabbar, A., Shi, Q., Rohrer, J.P.: Evaluation of network resilience, survivability, and disruption tolerance: analysis, topology generation, simulation and experimentation. *Telecommun. Syst.* **52**(2), 705–736 (2013)
58. Sterbenz, J.P.G., Hutchison, D., Cetinkaya, E.K., Jabbar, A., Rohrer, J.P., Schoeller, M., Smith, P.: Redundancy, diversity, and connectivity to achieve multilevel network resilience, survivability, and disruption tolerance. *Telecommun. Syst.* **56**(1), 17–31 (2014)
59. Strix Systems: http://www.strixsystems.com/Service_Providers.aspx. Accessed on 11 Jan 2015
60. Tapolcai, J.: Survey on out-of-band failure localization in all-optical mesh networks. *Telecommun. Syst.* **56**(1), 169–176 (2014)
61. Tapolcai, J., Ho, P.-H., Verchere, D., Cinkler, T., Haque, A.: A new shared segment protection method for survivable networks with guaranteed recovery time. *IEEE Trans. Reliab.* **57**(2), 272–282 (2008)
62. TerraNet AB: <http://www.terranet.se>. Accessed on 12 Jan 2015
63. Todd, B., Doucette, J.: Multi-flow optimization model for design of a shared backup path protected network. In: Proceedings of the IEEE International Conference on Communications (IEEE ICC'08), pp. 131–138 (2008)
64. Torkildson, E., Ananthasubramaniam, B., Madhow, U., Rodwell, M.: Millimeter-wave MIMO: wireless links at optical speeds. In: Proceedings of the 44th Allerton Conference on Communication, Control and Computing, pp. 1–9 (2006)
65. Tropos: <http://www.tropos.com/index1.php>. Accessed on 11 Jan 2015
66. TU-R F.1704. Characteristics of multipoint-to-multipoint fixed wireless systems with mesh network topology operating in frequency bands above about 17 GHz, ITU-R Recommendation F.1704 (2005)
67. Vasseur, J.-P., Pickavet, M., Demeester, P.: *Network Recovery*. Morgan Kaufmann, Burlington (2004)
68. Vural, S., Wei, D., Moessner, K.: Survey of experimental evaluation studies for wireless mesh network deployments in urban areas towards ubiquitous Internet. *IEEE Commun. Surv. Tutorials* **15**(1), 223–239 (2013)
69. XIOCOM: http://www.xiocom.com/serv_ind.html. Accessed on 09 Mar 2015
70. Yang, Y., Wang, J., Kravets, R.: Interference-aware load balancing for multihop wireless networks. Technical Report, University of Illinois at Urbana-Champaign (2005)
71. Zhang, J., Wu, H., Zhang, Q., Li, B.: Joint routing and scheduling in multi-radio multi-channel multi-hop wireless networks. In: Proceedings of the IEEE International Conference on Broadband Networks (BroadNETS'05), pp. 631–640 (2005)
72. Zhao, L., Gao, L., Zhao, X., Ou, Sh.: Power and bandwidth efficiency of wireless mesh networks. *IET Netw.* **2**(3), 131–140 (2013)

Chapter 10

Disruption-Tolerant Routing in Vehicular Ad Hoc Networks



Owing to a significant increase in the number of vehicles on roads, raising the possibility of accidents, we have been recently observing a growing interest in *inter-vehicular communications (IVC)* [55] enabled by the deployment of vehicular wireless communication systems. Following [23, 31, 55], *Vehicular Ad hoc Networks (VANETs)* are now considered by car manufacturers as an emerging solution to provide communications for a wide range of applications designed to solve a number of mutually nonindependent problems in particular related to:

- *Public safety* aspects. Road safety can be improved by messages exchanged by vehicles, e.g., in the case of accidents/collisions, bad weather conditions (ice/water on the road) [67], and unexpected events (e.g., low bridges, oil on the road), or to assist the drivers in lane change/overtaking operations [5, 6, 61]
- *Traffic coordination* issues. VANETs can be utilized to provide traffic monitoring/shaping (including traffic light management), i.e., aimed at adjusting the scheduling of traffic lights to help the drivers move in the green phase, thus also contributing to the *reduction of environmental pollution* [1]
- *Infotainment* providing the travelers with on-board information and entertainment services such as Internet access or music download [27, 36, 64]

Based on the ability to forward information at transit nodes, IVC networks can be next classified as either (1) single- or (2) multi-hop IVCs (shortly, SIVCs and MIVCs, respectively) [55], as shown in Fig. 10.1. Single-hop systems are often used by applications requiring short-range communications (e.g., automatic cruise control, lane merging). The latter group (i.e., multi-hop vehicular ad hoc networks used, e.g., by traffic monitoring applications) is investigated in detail later in this chapter. A detailed overview of vehicular networking issues is presented in [31].

It is worth noting that many VANET applications (e.g., related to collision warning or traffic coordination issues) require reliable real-time communications to work efficiently since information arriving too late is often no longer useful.

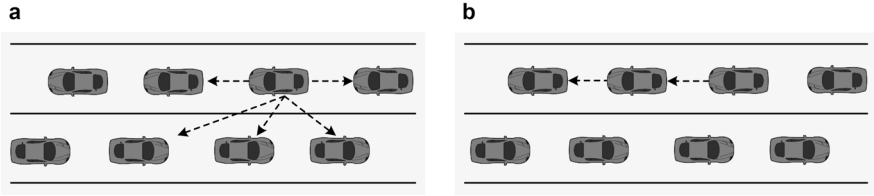


Fig. 10.1 Examples of (a) single- and (b) multi-hop inter-vehicular communications

Another important fact is that the usability of a VANET system frequently strongly depends on the *penetration rate*, defined as the ratio of vehicles equipped with VANET solutions. Any vehicle equipped with a system with p percent penetration rate will benefit in only p percent of all situations [55].

As proposed by the U.S. Federal Communications Commission (FCC) and defined in the IEEE 802.11 2012 specification (formerly the 802.11p standard [27]), vehicles equipped with wireless devices can form in an ad-hoc manner a VANET network utilizing seven 10 MHz channels in the 5.880–5.925 GHz band (often referred to as *Dedicated Short Range Communications (DSRC)* [64, 67]. According to this specification, effective channel capacity is in the range of 3–54 Mbps, while the typical link length is limited to about 300 m [2, 55, 64].

As shown in Fig. 10.2, the set of channels consists of one *control channel*—*CCH* (also denoted as CH 178)—and six 10 MHz *service channels (SCH)*, namely CH 172, 174, 176, 180, 182, and 184. Additional channel CH 170 (with 5 MHz bandwidth) is reserved for future use. High-power channel CH 184 is used for the distribution of safety messages only. Except for CH 184, all other service channels can be utilized by non-safety applications. To obtain higher data rates, channels CH 174 and 176 can be combined into a single 20 MHz channel CH 175. The same can be done for channels CH 180 and 182 to form channel CH 181.

In DSRC, each device, when tuned to the control channel (CCH) for half of the frame time (i.e., 50 ms), can distribute beacon messages containing information related to vehicle speed, location (coordinates), etc. [1]. Such beacons can be exchanged periodically with frequency in range (1, 10) Hz, i.e., every 100 ms–1 s. In the same interval, emergency messages can also be generated. As shown in Fig. 10.3,

Channel number	CH 170	CH 172	CH 174	CH 176	CH 178	CH 180	CH 182	CH 184
			CH 175			CH 181		
Channel use	For future use	SCH	SCH	SCH	CCH	SCH	SCH	SCH
Bandwidth [MHz]	5	10	10	10	10	10	10	10
			20			20		
Bit rate [Mbps]	–	3-27	3-27	3-27	3-27	3-27	3-27	3-27
			6-54			6-54		

Fig. 10.2 Utilization of VANET channels based on [2]

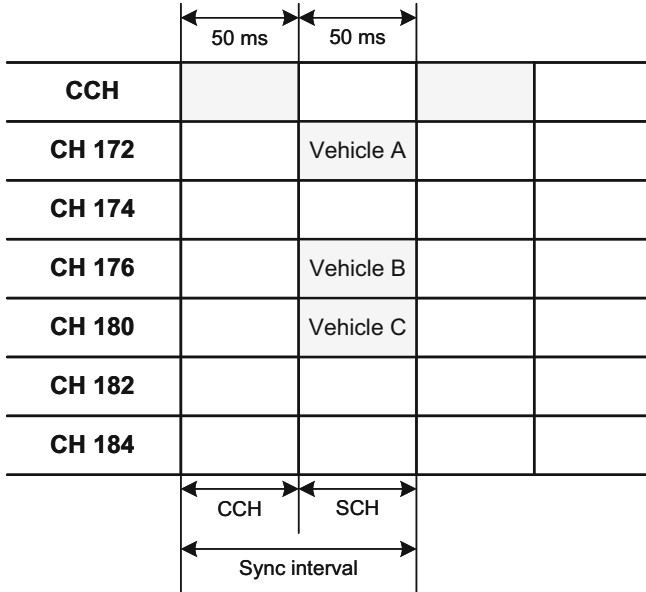


Fig. 10.3 Example of multichannel operation of DSRC

at the end of the CCH interval, devices can switch to one of the six service channels for the next 50 ms (SCH interval) to perform tasks related to the distribution of applications data (preceded by a negotiation phase at the end of CCH interval). IVC networks implement the major assumptions of the 802.11 standards family since IEEE 802.11a is used by both ETSI in Europe and IEEE in the USA as a basis for vehicular communications.

Vehicular communications can be provided either without or with the support of a roadside infrastructure, also referred to as *vehicle-to-vehicle (V2V)* and *vehicle-to-infrastructure (V2I)* wireless networking, respectively [31, 65]. V2V communications is infrastructure-free and provided only by *On-Board Units (OBUs)*—the appropriate in-vehicle equipment. In the case of V2I class, message forwarding always takes place between OBUs installed inside vehicles and the respective roadside infrastructure (including *Road-Side Units—RSUs*) [55].

Following [55], V2I systems can be further decomposed into *sparse* and *ubiquitous* systems offering services at selected points (e.g., hot-spots, road intersections, or in the entire network area, respectively). Examples of sparse V2I systems applications include parking availability, parking payment, collection of tolls for roads/bridges/tunnels, busy intersection scheduling, or gas station advertisement.

An example of a ubiquitous V2I system would be related, e.g., to vehicle to land-based communications offering high-speed Internet access providing onboard entertainment using an entire range of applications, from e-mail and media streaming to web browsing and voice over IP, independent of vehicle location, but, therefore, prohibitively expensive.

Following [55], each OBU, to work properly, should be equipped with:

- Central Processing Unit (CPU) implementing applications and communication protocols
- Wireless transceiver (to transmit/receive data to/from the neighboring vehicles or a roadside infrastructure)
- Global Positioning System (GPS) receiver providing positioning and time synchronization information
- Sensors measuring differentiated parameters
- Interface allowing for human-system interaction

VANETs can be considered a special case of *Mobile Ad hoc NETWORKS (MANETs)* due to their self-organization, self-management, short transmission range, and relatively low bandwidth. Following [28, 38, 63], individual characteristics of VANETs include:

- Highly dynamic topology with frequent topology changes resulting in common *path unavailability* or even causing network disconnections/partitioning (lifetime of a VANET link is typically measured in terms of seconds)
- Sufficient level of energy and storage (since vehicles, contrary to sensors, are not small devices). The only clear exception to this rule refers to nodes formed by *Vulnerable Road Users (VRUs)*, e.g., pedestrians [3].
- Utilization of geographic-based message distribution providing fast dissemination of time-critical information to other vehicles
- Stringent requirements on message propagation delay (e.g., for safety applications)

Due to these characteristics, providing reliable transmission is a challenging issue. The definition of communications reliability for VANETs also significantly differs from the generic one originally proposed for communication systems.

Following [40], the *reliability of inter-vehicular communications* can be defined as the ability to deliver messages to destination vehicles within the specified operation duration.

Although in the last decade, several tutorials have been published in the literature (covering, e.g., vehicular networking issues [23, 31], mobility models [22], information distribution [38, 49], characteristics of VANET applications [58], or green communications [1]), reliability of end-to-end vehicular communications is a relatively new issue with few proposals/results available.

In VANETs, data distribution (commonly referred to as *data dissemination*) is defined as the transportation of data to the intended recipients while satisfying certain requirements, such as, e.g., delay [58]. These requirements are obviously differentiated depending on the characteristics of applications and disseminated data.

To focus on the reliability of inter-vehicular communications, it is necessary to identify first the respective QoS requirements of VANET applications in terms of reliability attributes, in particular, including message delivery latency, as presented in Sect. 10.1. Such differentiation of application types implies differentiated QoS requirements. In particular, apart from services requiring real-time message delivery (e.g., safety applications), there also exist delay-tolerant applications (e.g., providing infotainment services).

Since one-hop broadcasting, also referred to as *single-hop message dissemination* [57], is the basic networking technique for many VANET applications (especially for safety-related services), there is a group of recent papers focusing on reliability issues of inter-vehicle links. For instance, the authors of [4, 8, 64] investigated the inter-vehicle distance distribution characteristics and vehicle movement patterns as the main factors of the limited lifetime of links. Indeed, for any two neighboring vehicles moving in opposite directions at a speed of 96 km/h, the average link lifetime is at most 10 s [38]. A detailed overview of single- and multi-hop message dissemination protocols can be found in [49].

Many VANET applications require multi-hop communications to deliver information to distant end nodes. Examples include V2V communications providing dissemination of safety-related messages to vehicles separated by several transit nodes. For them, communications reliability should be analyzed on the path level [17, 69], or in terms of end-to-end communications, e.g., in the case of multi-hop multipath routing/broadcasting [25, 54].

Issues of multi-hop data delivery are investigated in detail in Sect. 10.2. Sections 10.3 and 10.4, in turn, include our two original proposals to improve the reliability of end-to-end V2V communications. Section 10.5 presents the concluding remarks.

10.1 Reliability Requirements of VANET Applications

Following [23], VANET applications can be classified into safety, transport efficiency, and infotainment. In each case, transmission reliability is an integral part of QoS requirements due to its obvious relation with message delivery latency. In general, due to short-range communications, the reliability of inter-vehicular communications depends on the number of vehicles equipped with VANET solutions. However, differentiated characteristics of applications imply differentiated ways to achieve this goal. Categories of safety applications, as identified by the Vehicle Safety Communications Consortium (VSCC) in [59], are presented in Fig. 10.4 with information from [31] related to the upper bound on message delivery latency (i.e., the *critical latency*).

Safety applications require real-time communications since the validity of exchanged information (e.g., post-crash warnings) expires quickly, and any such delayed information shortly becomes useless for neighboring vehicles. Therefore,

Category	Application scenario	Minimum frequency	Critical latency
<i>Intersection Collision Avoidance</i>	<ul style="list-style-type: none"> - Blind Merge Warning - Intersection Collision Warning - Left Turn Assistant - Pedestrian Crossing Information Warning - Stop Sign Movement Assistant - Stop Sign Violation Warning - Traffic Signal Violation Warning 	10 Hz	< 100 ms
<i>Public Safety</i>	<ul style="list-style-type: none"> - Approaching Emergency Vehicle Warning - Emergency Vehicle Signal Preemption - Post-Crash Warning - SOS Services 	10 Hz	< 100 ms
<i>Sign Extension</i>	<ul style="list-style-type: none"> - Curve Speed Warning - In-Vehicle Amber Alert Warning - Low Bridge Warning - Low Parking Structure Warning - Work Zone Warning - Wrong Way Driver Warning 	10 Hz	< 100 ms
<i>Vehicle Diagnostics and Maintenance</i>	<ul style="list-style-type: none"> - Just-in-Time Repair Notification - Safety Recall Notice 	10 Hz	< 100 ms
<i>Information from Other Vehicles</i>	<ul style="list-style-type: none"> - Adaptive Headlamp Aiming - Automation System (Platoon) - Blind Spot Warning - Cooperative Adaptive Cruise Control - Cooperative Collision Warning - Cooperative Forward Collision Warning - Cooperative Glare Reduction - Cooperative Vehicle-Highway - Emergency Electronic Brake Lights - Highway Merge Assistant - Highway/Railroad Collision Warning - Lane Change Warning - Pre-Crash Sensing - Road Condition Warning - Vehicle-to-Vehicle Road Feature Notification - Visibility Enhancer 	10 Hz	< 100 ms

Fig. 10.4 Classification of safety applications based on [31, 59]

following [1, 31], 100 ms is considered the maximum latency of safety message delivery, while 10 Hz is the minimum frequency of such exchange of messages.

Safety-related notifications can be either *event-driven* or *periodic* [23]. Event-driven messages are disseminated after the identification of an event. Periodic notifications are, in turn, utilized to provide proactive distribution of messages related to vehicle status/location (e.g., in the case of forward collision warnings).

Safety applications, commonly using one-hop broadcasting, have stringent requirements on the minimum scope of message dissemination. According to [23], sending safety-related messages over a distance of at least 150 m should be feasible by one-hop broadcast communications. Regarding the multi-hop distribution of safety messages (each hop realized by one-hop broadcasting), the total coverage distance of safety applications is between 300 m and 20 km [19, 42].

Category	Application scenario	Minimum frequency	Critical latency
<i>Traffic Efficiency</i>	- Enhanced Route Guidance and Navigation	10 Hz	< 100 ms
	- Green Light Optimal Speed Advisory	10 Hz	< 100 ms
	- V2V Merging Assistance	10 Hz	< 100 ms
<i>Infotainment and Others</i>	- Internet Access in Vehicle	1 Hz	< 500 ms
	- Point of Interest Notification	1 Hz	< 500 ms
	- Remote Diagnostics	1 Hz	< 500 ms

Fig. 10.5 Classification of non-safety applications based on [42]

Figure 10.5 presents a classification of non-safety applications, as identified by the Car-to-Car Communications Consortium (C2C-CC), consisting of Audi, BMW, DaimlerChrysler, Fiat, Renault, and Volkswagen from [42], including information related to traffic efficiency and infotainment applications.

The former class (traffic efficiency) comprises applications utilizing either vehicle-to-infrastructure communications (e.g., traffic light scheduling to help the driver move in the green phase by keeping the green light optimal speed) or vehicle-to-vehicle communications (e.g., V2V merging assistance). Analogously, for the latter class of infotainment applications, either V2I communications (e.g., for point of interest (POI) notifications) or V2V communications (e.g., multi-hop Internet wireless access) can be utilized. However, there is generally no standardized consensus about the requirements concerning reliable communications characteristics and metrics to measure them.

A significant part of non-safety applications (especially related to infotainment issues) belongs to the class of delay-tolerant services, for which real-time data delivery is not required. For them, the maximum end-to-end latency can thus be higher (e.g., 500 ms, as shown in Fig. 10.5). Another characteristic is that, contrary to safety applications, non-safety services, e.g., Internet access apart from operating in a V2I environment, often use multi-hop V2V communications.

For non-safety applications without stringent requirements on real-time data delivery, it is frequently sufficient to use the best-effort scheme, e.g., by incorporating the store-carry-forward technique [47, 63]. This solution allows the messages to be stored at a given transit node until the next forwarding node becomes available (i.e., if it appears available in the communications range of the transit node).

Figure 10.6 summarizes functionalities related to communications type, addressing, efficiency, and real-time requirements of selected applications.

10.2 Network Layer Addressing and Routing Issues

Concerning addressing issues, two schemes can be distinguished, namely fixed and geographical addressing [55]. In *fixed addressing*, a node is assigned a specified

Application	Communications type				Addressing scheme	Efficiency dependent on OBU density	Real-time requirements
	Single-hop V2V	Multi-hop V2V	Sparse V2I	Ubiquitous V2I			
Car-to-land communications				+	<i>Fixed</i>		
Collision warning (highway)		+		+	<i>Geo</i>	+	+
Collision warning (intersection)	+	+	+	+	<i>Geo</i>	+	+
Targeted vehicular communications	+	+		+	<i>Fixed</i>	+	
Traffic coordination	+	+		+	<i>Geo</i>	+	+
Traffic light scheduling			+	+	<i>Geo</i>	+	
Traffic monitoring		+		+	<i>Geo</i>	+	
Traveller information support			+	+	<i>Geo</i>		

Fig. 10.6 Summary of representative application requirements based on [55]

address once it joins the network, which remains unchanged until it leaves the network.

In *geographical addressing*, where each node is characterized based on its geographical coordinates, the address assigned to a given node based on its location changes as the vehicle moves (i.e., not necessarily leaving the network). Apart from geographical information, packet forwarding often depends on additional attributes, e.g., direction of vehicle movement, road ID, vehicle type, height, weight, maximum speed, or even driver characteristics (i.e., beginner, professional) [55].

In the later part of this chapter, we focus on multi-hop V2V communications where data is forwarded via multiple hops from a sender to one/multiple receivers. At this point, it is necessary to emphasize that in some papers, e.g., in [38], multi-hop broadcasting is improperly considered as one of the VANET routing schemes since, in practice, it does not involve any Layer-3 processing (apart from Layer-2 broadcasting). Therefore, multi-hop routing and multi-hop broadcasting should be analyzed separately, as considered in Sects. 10.3 and 10.4, respectively.

Multi-hop V2V networking practically has no physical boundaries. As a result, the capacity of such an unbounded system does not scale. To find a scalable solution, it is usually assumed that data can be forwarded to vehicles located within a specific area [55]. Depending on application requirements, the following routing approaches can be distinguished [55]:

- Unicast routing with fixed addressing (e.g., for entertainment applications like file transfer)
- Unicast routing with geographical addressing utilized to improve routing efficiency (compared to fixed addressing)
- Multicast routing with fixed addressing—possible in theory, but practically not used due to a huge overhead related to multicast groups maintenance
- Multicast routing with geographical addressing—used by most applications (including, e.g., emergency warning or traffic monitoring applications)

However, multicast routing with geographical addressing is often replaced by broadcast multi-hop transmission addressed in Sect. 10.4.

10.2.1 Unicast Routing with Fixed Addressing

Targeted (i.e., unicast) multi-hop forwarding allows for *localized* communications between two vehicles. Example applications include voice/video transmission or instant messaging between vehicles traveling together for long distances [55]. For this combination of routing and addressing, two sorts of routing protocols can be distinguished: AODV-based and cluster-based protocols.

Ad hoc On-Demand Distance Vector (shortly, *AODV*) routing, being an example of a standard reactive routing protocol for ad hoc networks [51], has also been investigated as a base routing protocol for vehicular networking [46]. AODV does not maintain any route that is not needed [50, 58]. Before a packet is sent, route discovery is initiated by the source node by broadcasting the Route Request (RREQ) message toward the destination node. Upon receiving the RREQ message by any transit node, its routing table is updated, and the RREQ is rebroadcast. After RREQ is received by the destination node, the Route Response (RREP) unicast message is sent back toward the source node along the reverse path. The path is finally set up after RREP is received by the source node. The route is used as long as it is active.

The respective changes to conventional AODV protocol are needed to adapt it to VANET networks due to the lack of apparent boundaries of VANET systems topologies, e.g., as in [48] where RREQs are forwarded only in certain zones, or as in [62], where RREQs are broadcast up to the maximum number of hops.

End-to-end unicast communications can also be performed hierarchically, e.g., using one of the cluster-based routing protocols, where vehicles are organized into virtual clusters coordinated by cluster heads (see Fig. 10.7) [12]. Inter-cluster communication is done via cluster heads, while intra-cluster message dissemination is feasible via direct links.

10.2.2 Unicast Routing with Geographical Addressing

Concerning geographical addressing, a large number of proposals are based on the Greedy Perimeter Stateless Routing (GPSR) protocol from [32], i.e., the location-based unicast routing protocol, assuming that VANET nodes maintain only information about neighboring vehicles related, e.g., to their location.

Under geographical addressing, two major methods of packet forwarding can be distinguished: *greedy forwarding* and *perimeter forwarding* [28]. The former assumes that the packet is forwarded to the neighboring node located geographically closest to the destination node. If such a neighbor does not exist (e.g., due to the respective gap region with no nodes between the current node and the destination

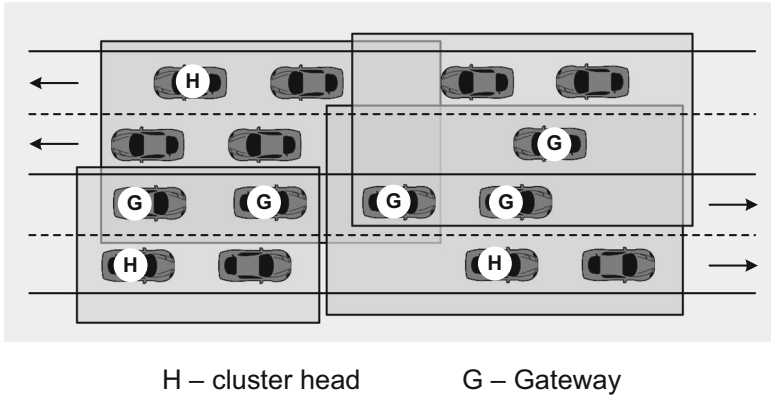


Fig. 10.7 Example scenario of cluster-based routing

node), the latter (i.e., perimeter) forwarding can be used to forward the packet around the perimeter of this gap region to the counterclockwise neighbor with respect to the current node.

10.2.3 Multicast Routing with Geographical Addressing

Since geographical addressing is better suited for multicasting, such a combination (often referred to as *geocasting* [41]) is frequently in use for VANET communications to forward the message to vehicles located in a specified geographical region (commonly of a rectangular/circular shape). This forwarding is typically provided by flooding the packets [21] within a forwarding zone. For instance, information referring to road accidents or traffic lights (see Fig. 10.8) would typically affect only vehicles coming from behind [1, 9].

A summary of the fundamental characteristics of VANET routing protocols is presented in Fig. 10.9.

10.2.4 Broadcast Multi-hop Message Dissemination

Multi-hop broadcasting [38] is a frequently used method of multi-hop dissemination of messages such as, e.g., weather-, road condition-, or accident-related announcements, including, e.g., a detour route, an accident alert, or a construction warning [31]. It is also often used in the initial phase of unicast route discovery (for instance, as in AODV routing). Finally, broadcasting is a good scheme if a message needs to be disseminated in a broadcast way to multiple nodes, but the transmission range exceeds a single-hop distance.

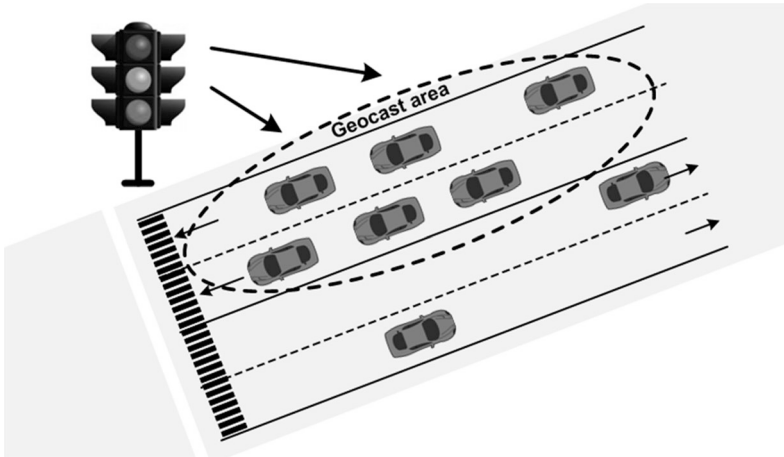


Fig. 10.8 Example geocast area used in geographical routing (comprising vehicles based on their geographical location)

Protocol	Addressing	Uni/Multicast	Path state	Neighbor state	Hierarchical
AODV	Fixed	Unicast	Yes	Yes	No
Cluster	Fixed	Unicast	Yes	Yes	Yes
GPSR	Geographical	Unicast	No	Yes	No
Geocasting	Geographical	Multicast	No	No	No

Fig. 10.9 Characteristics of example VANET routing protocols

Since the main focus of this book is on the resilience of routing schemes, in the later part of this chapter, we will focus on the resilience of multi-hop end-to-end V2V communications. In particular, we will address an important problem of VANET communications availability for end-to-end unicast routing, which, to our knowledge, has been only marginally considered in the literature. The problem is essential due to the existence of numerous applications making use of end-to-end unicast communications, including voice/video, instant messaging between vehicles traveling together, or multi-hop Internet access, to mention a few.

In VANETs, end-to-end path availability is challenging due to the frequent faults of VANET links [65]. Therefore, we will focus on providing a high level of disruption tolerance by searching for “stable links” able to increase the lifetime of communication paths and present two original algorithms of multipath and anypath communications in Sects. 10.3 and 10.4, respectively.

10.3 Improving the Resilience of End-to-End V2V Communications by Multipath Routing

This section focuses on the resilience of multi-hop unicast V2V communications. In particular, we present our technique of multi-hop multipath end-to-end V2V routing enhanced with functionalities to select stable VANET links in path computations and being able to provide differentiated levels of service availability to respond to differentiated requirements of applications from [52]. To the best of our knowledge, such a solution jointly taking into consideration the issues of (1) communications paths stability, (2) multipath routing, and (3) provisioning of multiple levels of service availability for differentiated application classes has not been proposed before.

In the literature, there is currently no consensus concerning the inter-vehicle distance distribution having a direct influence on the lifetime of VANET links, as well as on end-to-end communication paths. For instance, in [56], the respective analysis of link lifetime was presented for a codirectional vehicles scenario (i.e., for vehicles moving in the same direction) under the assumption of equal spaces between vehicles and normally distributed velocities. For such a scenario, the log-normal distribution was shown in [13] to be proper for modeling the headway distance and next utilized in [64] to present an improved analysis of link lifetime, including differentiated velocities and accelerations of codirectional vehicles.

However, even though quite realistic for a highway scenario, such a simplified case of codirectional vehicles seems less important, e.g., in urban environments, where differentiated directions of vehicle movements play a major role. Besides, as shown in other papers, inter-vehicle distances can be modeled by gamma [64], exponential [24, 67], or Poisson distribution [30] as well.

To mitigate the problem of the short lifetime of V2V multi-hop paths, several approaches to multi-hop routing have been proposed in the literature aimed at improving path characteristics related to the stability of traversed links. Among them, we can distinguish single-path algorithms (e.g., [45]) utilizing mobility-related information (direction and velocity of vehicles) to find transmission paths traversing links with a low probability of being broken in the near future. However, following [7], even if the link stability criterion is incorporated into the path computation scheme, owing to a high level of node mobility, the lifetime of a multi-hop path is commonly shorter than the time needed to install the path.

Another solution to improve the reliability of end-to-end transmission is to utilize multipath routing, which can transmit information via multiple (frequently disjoint) paths. Additionally, multipath routing is also characterized by improved network throughput, load balancing, and packet delivery ratio [25].

Among end-to-end multipath routing algorithms available in the literature, two extensions of the AODV routing scheme are worth distinguishing, namely, Ad hoc On-demand Multipath Distance Vector (AOMDV) introduced in [43] and Ad hoc On-demand Distance Vector Multipath (AODVM) from [66] establishing multiple link-/node-disjoint paths, respectively. However, the multipath concept itself may

not be sufficient since the high mobility of vehicles is often responsible for failures of all alternate paths between a certain pair of end nodes in a short time [63].

Despite the clear advantages of both approaches, i.e., link stability-oriented path selection and multipath routing, there is practically no approach available combining both features apart from our one from [52] presented in this section as follows. Section 10.3.1 includes (1) analysis of the probability of end-to-end transmission availability for multi-hop multipath V2V communications in the presence of link failures under the assumption of the exponential distribution of inter-vehicle distances and (2) numerical results necessary to determine the number of end-to-end disjoint paths sufficient to improve the multipath transmission availability visibly.

These results are next utilized in Sect. 10.3.2 to propose the concept of multipath link-disjoint end-to-end routing aimed at establishing paths characterized by increased lifetime. This approach is also designed to provide differentiated service availability levels to respond to differentiated requirements of applications.

Simulation results and conclusions are presented in Sect. 10.3.3.

10.3.1 Probability of V2V Transmission Availability

A V2V network model considered in this section is focused on analyzing the inter-vehicle connectivity. Therefore, it disregards other issue like transmission errors, delay, or contention. Since link faults are responsible for most VANET failures, here we focus on protection against link faults. Any two vehicles i and j are said to be connected by a direct link $a_h = (i, j)$ if the distance $r_{i,j}$ between them is not greater than the maximum range r_{\max} [20]. Therefore, the probability ρ_h that two vehicles are connected by link a_h at any time t can be calculated based on a probability density function of inter-vehicle distance $p(r_{i,j})$, as given in Eq. 10.1.

$$\rho_h = P(r_{i,j} < r_{\max}) = \int_0^{r_{\max}} p(s)ds \tag{10.1}$$

In the case of a single-path routing (Fig. 10.10a), if path η consists of k_n links, then assuming mutual independence of link lengths (as being commonly investigated [44, 67]), the probability $\tilde{\pi}_n$ of path existence can be expressed by Eq. 10.2.

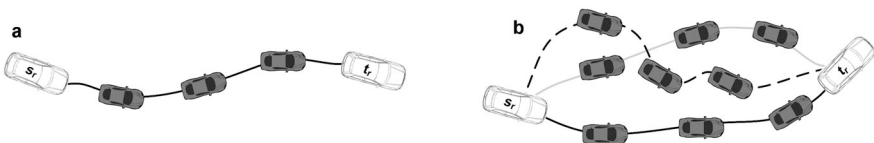


Fig. 10.10 Examples of (a) single-path and (b) multipath link-disjoint routing

$$\tilde{\pi}_n = \prod_{h:a_h \in \eta} \rho_h = \left(\int_0^{r_{\max}} p(s) ds \right)^{k_n} \quad (10.2)$$

In a multipath transmission scenario including m end-to-end link-disjoint paths (see Fig. 10.10b), the destination node can be reached if at least one of all m end-to-end link-disjoint paths is operational. In such a scheme, the probability of multipath transmission availability $\tilde{\Psi}_m$ can be determined for any time t , as given in Eq. 10.3.

$$\tilde{\Psi}_m = 1 - \prod_{n=1}^m (1 - \tilde{\pi}_n) = 1 - \prod_{n=1}^m \left(1 - \left(\int_0^{r_{\max}} p(s) ds \right)^{k_n} \right) \quad (10.3)$$

Assuming the exponential distribution of inter-vehicle distances (following, e.g., [24, 67], probabilities of link existence (ρ_h) and multipath transmission availability ($\tilde{\Psi}_m$) at any time t are given by Eqs. 10.4 and 10.5, respectively.

$$\rho_h = \int_0^{r_{\max}} \lambda e^{-\lambda s} ds = 1 - e^{-\lambda \cdot r_{\max}} \quad (10.4)$$

$$\tilde{\Psi}_m = 1 - \prod_{n=1}^m \left(1 - \left(1 - e^{-\lambda \cdot r_{\max}} \right)^{k_n} \right) \quad (10.5)$$

Example values of $\tilde{\Psi}_m$ as a function of end-to-end path count m , assuming that $r_{\max} = 300$ m and $\lambda = 0.01$, are presented in Fig. 10.11. These results show that multipath routing can provide a suitable means to improve the probability of transmission availability. However, increasing m above 2–3 does not provide any significant improvement. In path computations, it is thus reasonable to limit the number of end-to-end disjoint paths to the value sufficient to meet the requirements of particular applications.

To analyze the time-dependent probability of multi-hop path availability for any time t_0 , we need to derive first the formula determining the existence of a single link a_h after Δt time. We assume that for each vehicle i , $\Phi_i(t_0) = [x_i(t_0), y_i(t_0)]^T$ is its position vector at initial time t_0 . The initial distance between vehicles i and j (see Fig. 10.12) can be defined by Eq. 10.6.

	$m = 1$ path	$m = 2$ paths	$m = 3$ paths	$m = 4$ paths	$m = 5$ paths
$\tilde{\Psi}_m^*$	0.8257	0.9622	0.9905	0.9973	0.9991

* for the average hop counts of 1st-5th path equal to 3.75, 4.79, 5.69, 6.63, and 7.61, respectively.

Fig. 10.11 Example values of $\tilde{\Psi}_m$ as a function of end-to-end link-disjoint path count m for exponential inter-vehicle distance distribution

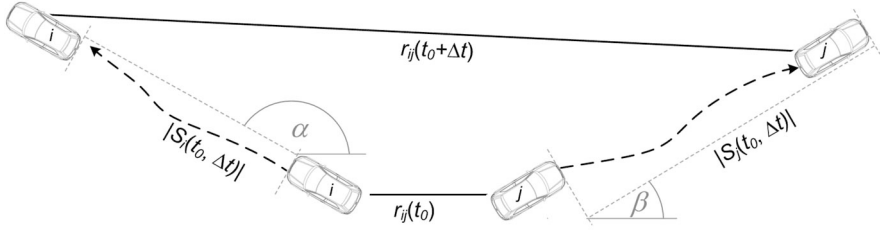


Fig. 10.12 Example scenario of vehicle movements

$$r_{i,j}(t_0) = |\Phi_i(t_0) - \Phi_j(t_0)| = \sqrt{(x_i(t_0) - x_j(t_0))^2 + (y_i(t_0) - y_j(t_0))^2} \quad (10.6)$$

After Δt time units, information on vehicle i displacement can be represented by the movement vector $S_i(t_0, \Delta t) = [s_i^x(t_0, \Delta t), s_i^y(t_0, \Delta t)]^T$ with consecutive elements referring to movement information along X and Y axis, respectively, depending on vehicle i velocity function $v_i(t) = [v_i^x(t), v_i^y(t)]^T$ in $(t_0, t_0 + \Delta t)$ interval. For each vehicle i , S_i thus also includes information on direction. At time $t_0 + \Delta t$, a new position vector $\Phi_i(t_0 + \Delta t)$ of vehicle i is given by Eq. 10.7.

$$\begin{aligned} \Phi_i(t_0 + \Delta t) &= \Phi_i(t_0) + S_i(t_0, \Delta t) \\ &= \begin{bmatrix} x_i(t_0) \\ y_i(t_0) \end{bmatrix} + \begin{bmatrix} s_i^x(t_0, \Delta t) \\ s_i^y(t_0, \Delta t) \end{bmatrix} = \begin{bmatrix} x_i(t_0) \\ y_i(t_0) \end{bmatrix} + \begin{bmatrix} \int_{t_0}^{t_0 + \Delta t} v_i^x(s) ds \\ \int_{t_0}^{t_0 + \Delta t} v_i^y(s) ds \end{bmatrix} \end{aligned} \quad (10.7)$$

Vehicles i and j thus remain connected at $t_0 + \Delta t$, if

$$r_{i,j}(t_0 + \Delta t) = |\Phi_i(t_0 + \Delta t) - \Phi_j(t_0 + \Delta t)| \leq r_{\max} \quad (10.8)$$

The left part of the formula (10.8) can be extended based on Eqs. 10.6–10.7, as in Eq. 10.9.

$$\begin{aligned} r_{i,j}(t_0 + \Delta t) &= \sqrt{(x_i(t_0 + \Delta t) - x_j(t_0 + \Delta t))^2 + (y_i(t_0 + \Delta t) - y_j(t_0 + \Delta t))^2} = \\ &= \sqrt{\left(x_i(t_0) + \int_{t_0}^{t_0 + \Delta t} v_i^x(s) ds - \left(x_j(t_0) + \int_{t_0}^{t_0 + \Delta t} v_j^x(s) ds \right) \right)^2 + \\ &\quad + \left(y_i(t_0) + \int_{t_0}^{t_0 + \Delta t} v_i^y(s) ds - \left(y_j(t_0) + \int_{t_0}^{t_0 + \Delta t} v_j^y(s) ds \right) \right)^2} \end{aligned} \quad (10.9)$$

The respective probabilities of single link existence (ρ_h), single-path transmission availability ($\tilde{\pi}_n$), and multipath transmission availability ($\tilde{\Psi}_m$) can be defined by Eqs. 10.10–10.12.

$$\rho_h(t_0 + \Delta t) = P(r_{i,j}(t_0 + \Delta t) < r_{\max}) \quad (10.10)$$

$$\tilde{\pi}_n(t_0 + \Delta t) = \prod_{h:a_h \in \eta} \rho_h(t_0 + \Delta t) = (P(r_{i,j}(t_0 + \Delta t) < r_{\max}))^{k_n} \quad (10.11)$$

$$\begin{aligned} \tilde{\Psi}_m(t_0 + \Delta t) &= 1 - \prod_{n=1}^m (1 - \tilde{\pi}_n(t_0 + \Delta t)) \\ &= 1 - \prod_{n=1}^m \left(1 - \left(P(r_{i,j}(t_0 + \Delta t) < r_{\max}) \right)^{k_n} \right) \end{aligned} \quad (10.12)$$

Further analysis of $\tilde{\Psi}_m(t_0 + \Delta t)$ requires information related to specific traffic patterns, as well as its impact on $\rho_h(t_0 + \Delta t)$ values. Since our main interest is to improve multi-hop transmission availability in the presence of link failures, the above formulas will be helpful to introduce a routing technique that establishes paths traversing “stable links,” i.e., links with a high probability of existence after Δt time.

10.3.2 Provisioning of Multiple Availability Classes

VANET applications, like any others designed for various network architectures, are characterized by differentiated requirements related to *service availability* (i.e., probability of being in an “up” state [14]). In order not to overprovision a remarkable set of low-priority applications by offering only a single class of service, there is a reasonable need to propose an elastic approach able to meet these differentiated characteristics. Otherwise, most applications would be offered a better level of service than necessary at the price of the increased network load. Therefore, in this section, we introduce the respective class-based approach and define the three following availability classes shown in Fig. 10.13.

Differentiated guarantees on path availability are achieved here by the routing scheme establishing multiple end-to-end link-disjoint paths as follows:

- *Bronze class*: a single multi-hop path
- *Silver class*: $m = 2$ link-disjoint multi-hop paths
- *Gold class*: $m = 3$ link-disjoint multi-hop paths

Based on topological constraints of VANETs often limiting the number of disjoint end-to-end paths nearly equal to the average node’s degree [15], the

Class	Example applications
Bronze	Delay-tolerant services (e.g., Internet access; infotainment)
Silver	E.g. traffic coordination
Gold	Real-time services (e.g. emergency warnings)

In the literature, there are also other approaches related to service differentiation. However, in the case of VANETs, they refer, e.g. to the differentiation of transmission opportunity time (like the EDCA approach from [10], [26]). However, unlike in EDCA, the objective of our approach is to provide differentiation in terms of levels of protection against link failures.

Fig. 10.13 Proposed classes of path availability

maximum number of link-disjoint paths is assumed here to be equal to 3 (which also complies with the results from Fig. 10.11 showing an only marginal increase of probability of transmission availability for the number of disjoint paths m over 3).

Proposed Metric of Link Costs

To establish multi-hop paths with a low probability of being broken in a short time, we need to introduce a metric of link costs aimed at selecting links with an estimated long lifetime. This can be obtained, e.g., by selecting links between neighboring vehicles having similar velocity vectors. In the ideal case, if for any t in the $(t_0, t_0 + \Delta t)$ interval, conditions $|v_i(t)| = |v_j(t)|$ and $\alpha = \beta$ are satisfied (see Fig. 10.12), then inter-vehicle distance will be unchanged after Δt units of time.

Figure 10.14 presents example changes of inter-vehicle distance $r_{i,j}$ as a function of Δt analyzed for various relations of angles α and β from Fig. 10.12, for two scenarios of constant linear velocities of vehicles i and j and the initial inter-vehicle distance at t_0 equal to 100 m.

Figure 10.14 thus shows that to increase the lifetime of any VANET multi-hop path, it is essential to select links between neighboring vehicles characterized by similar movement vectors. However, at any time t_0 , precise information related to movement vectors is, for obvious reasons, available only concerning the past $(t_0 - \Delta t, t_0)$ interval. Therefore, in our approach, we propose to estimate the future inter-vehicle distance at time $t_0 + \Delta t$ based on the respective information on vehicle movement from the past $(t_0 - \Delta t, t_0)$ interval.

In particular, we propose to use formula (10.13) to evaluate the cost ξ_h of any link a_h using the information on neighboring vehicles movement in the interval $(t_0 - \Delta t, t_0)$. According to (10.13), the minimum cost ($\xi_h = \varepsilon$) is assigned to links between neighboring vehicles i and j characterized by equal movement vectors in $(t_0 - \Delta t, t_0)$ interval. Contrary to existing approaches (e.g., [13, 56, 64]), links with estimated long lifetimes are thus preferred in our scheme.

$$\xi_h = \sqrt{(s_i^x(t_0 - \Delta t, t_0) - s_j^x(t_0 - \Delta t, t_0))^2 + (s_i^y(t_0 - \Delta t, t_0) - s_j^y(t_0 - \Delta t, t_0))^2} \quad (10.13)$$

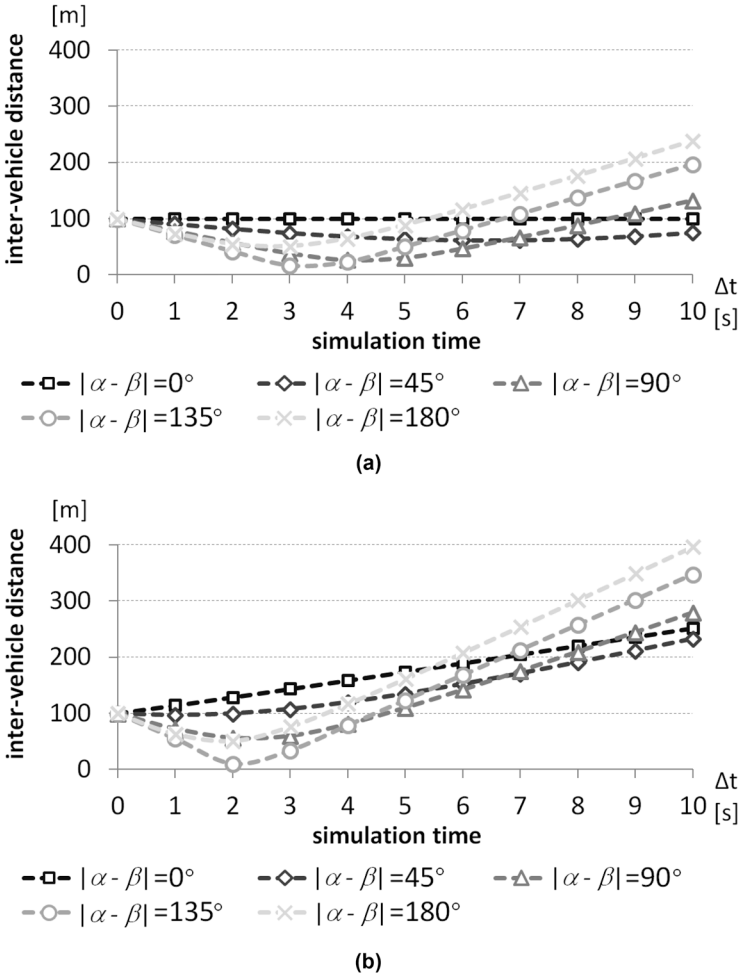


Fig. 10.14 Examples of inter-vehicle distance $r_{i,j}$ as a function of Δt for various values of $|\alpha - \beta|$ difference from Fig. 10.12 and constant linear velocity values: (a) $|v_i| = |v_j| = 16$ m/s and (b) $|v_i| = 16$ m/s, $|v_j| = 32$ m/s

Remarks on Routing Algorithm Extensions

The proposed approach can be applied to any V2V routing algorithm. Due to the popularity of AODV routing, as well as the availability of its multipath link-disjoint AOMDV version in the literature, here we evaluate our method by introducing the Class-Based Multipath link-disjoint V2V routing scheme based on the AODV algorithm (CBM-AODV), as given in Fig. 10.15. AOMDV is also used in Sect. 10.3.3 as a reference approach in all performance comparisons.

INPUT
<ul style="list-style-type: none"> – set V of vehicles; A – set of arcs a_h representing links between neighboring vehicles i and j – set D of transmission demands, each demand d_i represented by the end nodes s_i and t_i, and class of a demand
OUTPUT a set of m end-to-end link-disjoint paths
For each demand:
Step 1 Determine the number m of necessary end-to-end link-disjoint paths.
Step 2 Send m copies of RREQ broadcast message from source node s_r towards destination node t_r . In order to provide link disjointedness of established end-to-end paths, when forwarding the RREQ messages by each transit node i received from a given preceding node j :
2.1 Update the current cost of a path from s_r to i based on the cost of link (j, i) using Eq. 10.13.
2.2 Forward the RREQ message towards t_r , if the incoming RREQ message has not been sent by preceding node j to node i before (to be determined based on structures from Fig. 10.16).
Step 3 Upon receiving RREQ messages by the destination node t_r , send the respective RREP messages towards the source node s_r with respect to m RREQ messages having the lowest total path cost according to Eq. 10.13.

Fig. 10.15 CBM-AODV procedure to establish end-to-end link-disjoint paths

application ID	source node s_r	destination node t_r	preceding node j
----------------	-------------------	------------------------	--------------------

Fig. 10.16 Structures used in CBM-AODV to establish link-disjoint paths

Several updates to the conventional AODV routing algorithm are necessary to implement our solution. In order to obtain m end-to-end link-disjoint paths, source node s_r has to send toward destination node t_r multiple (i.e., m) Route Request messages—RReqs (see Steps 1 and 2 from Fig. 10.15). RReqs are followed by receiving m Route Reply messages (RReps), characterized by m lowest total path costs (compared to sending one RRep message only in the original AODV scheme)¹—Step 3 from Fig. 10.15. To make it feasible, information on the number of required RReps should be included in the RReq message.

Another important modification refers to the desired link disjointedness of multiple end-to-end paths, which can be provided by structures shown in Fig. 10.16 to be stored at each transit node i . This is to assure that the next copy of the RReq message originally sent from source node s_r toward destination node t_r via a given preceding node j is not sent by node i toward t_r again, as long as the respective paths remain operational (Steps 2.1 and 2.2).

In a typical scenario of broadcasting the RReq messages followed by sending back the RRep messages, established paths are commonly the cheapest ones in terms of the message propagation delay (which does not guarantee establishing paths that traverse links with estimated long duration time, referred to as “stable links” in this section). This problem is overcome in our scheme by implementing the cost metric

¹ In our multipath algorithm, the number of exchanged control messages is the same as for the reference AOMDV technique.

from formula (10.13) by extending the RReq message broadcasted by vehicle j to include additional fields for storing the values $s_j^x(t_0 - \Delta t, t_0)$ and $s_j^y(t_0 - \Delta t, t_0)$ and the total path cost.

To increase the level of service availability, the procedure of finding a new path is launched immediately after detecting the failure of any path (i.e., not waiting for detection of failures of all m alternate paths).

For each demand, the proposed scheme is characterized by the polynomial complexity bounded above by $O(|N|)$, where $|N|$ is the number of network nodes since its primary determinant is related to the task of establishing a single end-to-end path by broadcasting the RRep messages (of $O(|N|)$ complexity).

10.3.3 Analysis of Modeling Results and Conclusions

Evaluation of characteristics of our approach was performed using simulations for the realistic scenario of a 53-node VANET network from Fig. 10.17. The aim of the simulations was to evaluate the average values of path length, hop count, and the forecasted path lifetime. As proposed in the former subsection, the movement vector S_i of any vehicle i in the interval $(t_0, t_0 + \Delta t)$ was estimated based on the respective information from the past interval $(t_0 - \Delta t, t_0)$, with $\Delta t = 1$ s. Following [30], since message transmission delay can be considered negligible, network topology (i.e., locations of vehicles and their velocities) was assumed to be “frozen” in all path computations (i.e., it did not change).

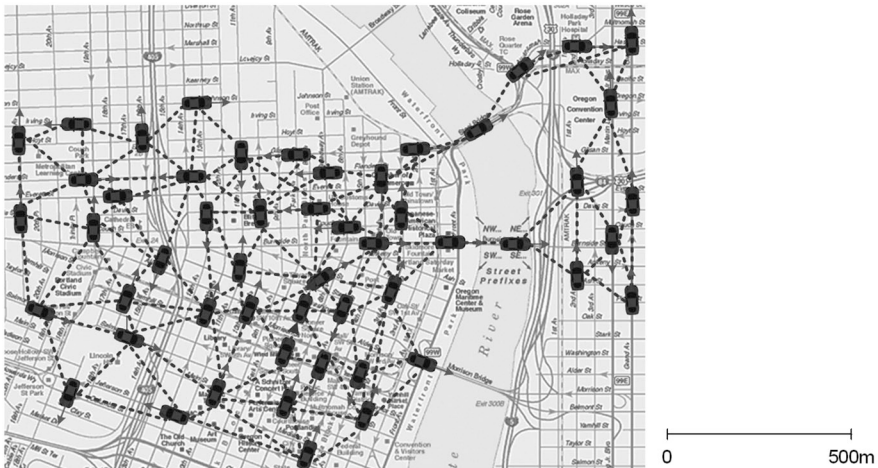


Fig. 10.17 Example VANET network (Portland area, USA) used in simulations

Experiments included two scenarios. In Scenario A, all vehicles were assumed to have equal average linear velocity (i.e., 10 m/s) in the time interval $(t_0 - \Delta t, t_0)$. Scenario B was, in turn, designed to simulate the uniform distribution of the average linear velocity in the range of 0–16 m/s (complying with common speed limitations in urban areas). In each scenario, the set of transmission demands comprised all pairs of vehicles equally divided into three proposed classes (i.e., bronze, silver, and gold).

Characteristics of the proposed approach were compared with the respective reference ones of the AOMDV link-disjoint multipath routing algorithm from [43] using a common transmission delay metric. Evaluation results are presented in Figs. 10.18 and 10.19 for 1st, 2nd, and 3rd link-disjoint paths (denoted as P1, P2, and P3, respectively). The respective lengths of 95% confidence intervals of the average values of analyzed parameters are not presented due to negligibly small sizes.

In general, the average length of paths calculated by the proposed CBM-AODV algorithm was about 17% greater in relation to the characteristics of the reference approach. However, this implied an increase of the end-to-end transmission delay only by about 1 ms for the analyzed network from Fig. 10.17, which was almost negligible.

The average cost of established paths calculated based on Eq. 10.13 for our CBM-AODV approach was up to 33% lower (Scenario A). This difference was insignificantly lower for Scenario B with differentiated linear velocities of vehicles. These results show that our approach can establish end-to-end paths with remarkably improved lifetime. A detailed analysis of path lifetime presented in the right part of Figs. 10.18 and 10.19 also indicated up to 45% (22.64% on average) better results for the CBM-AODV approach compared to the reference scheme.

Service class	Algorithm	Hop count			Path cost			Path lifetime [s]		
		P1	P2	P3	P1	P2	P3	P1	P2	P3
Bronze	CBM-AODV	4.97	–	–	12.73	–	–	84.78	–	–
	Reference approach	4.20	–	–	18.85	–	–	70.12	–	–
Silver	CBM-AODV	4.87	6.50	–	12.29	21.25	–	86.40	57.52	–
	Reference approach	4.29	5.54	–	16.62	28.01	–	77.77	39.44	–
Gold	CBM-AODV	3.82	4.40	5.54	10.00	15.55	23.06	107.36	76.40	38.93
	Reference approach	3.40	3.86	4.74	13.08	19.08	28.32	99.88	66.24	31.92

Fig. 10.18 Path characteristics for Scenario A

Service class	Algorithm	Hop count			Path cost			Path lifetime [s]		
		P1	P2	P3	P1	P2	P3	P1	P2	P3
Bronze	CBM-AODV	5.04	–	–	26.77	–	–	28.58	–	–
	Reference approach	4.20	–	–	35.65	–	–	22.21	–	–
Silver	CBM-AODV	5.04	6.59	–	26.76	45.10	–	28.58	21.20	–
	Reference approach	4.37	5.46	–	31.47	55.15	–	24.79	15.20	–
Gold	CBM-AODV	3.73	4.79	5.75	19.82	28.82	42.88	35.06	26.59	18.34
	Reference approach	3.29	4.01	4.71	23.35	35.92	51.55	31.10	21.02	14.51

Fig. 10.19 Path characteristics for Scenario B

For each analyzed algorithm, path lifetime decreased with the increase of the number of path links since it was determined by the minimum value of the individual lifetime of path links. It is worth noting that, in each scenario, the maximum time needed to establish the alternate path after a link failure was at most 15 ms, which in turn allowed any vehicle to change its location by at most 0.48 m (if we consider the maximum velocity of 32 m/s, e.g., as commonly assumed for highways).

Therefore, especially for the introduced silver and gold availability classes, it is unlikely that the remaining working paths will fail while re-establishing one of the failed paths. This, in turn, confirms the efficiency of our solution in assuring transmission continuity in the presence of VANET link failures.

10.4 A New Approach to Anypath Forwarding Providing Long Path Lifetime

In this section, we focus on *anypath forwarding* being another means to improve the reliability of multi-hop communications. The general difference between end-to-end multipath and anypath forwarding is that in the former scheme (considered in Sect. 10.3), a packet is sent in parallel along multiple paths (see Fig. 10.20a), while in the latter (i.e., anypath) forwarding, at each stage it is received as a broadcast message by several neighboring nodes but is later forwarded by only one of them (Fig. 10.20b).

In anypath scheme, also called (*opportunistic*) routing [16], the set of neighboring nodes, a packet is sent to, is called the *forwarding set*—Fig. 10.21. This set is selected in advance in the route planning phase for each transit node forwarding the packets toward a given destination node [35].

Under anypath forwarding, nodes from the forwarding set act cooperatively to forward the packet toward the destination node.² However, based on relay priorities assigned to neighboring nodes by a reliable anycast scheme [35], only one of these

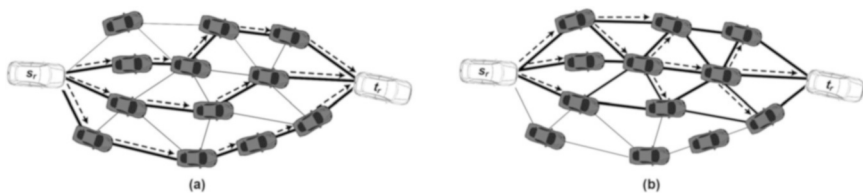
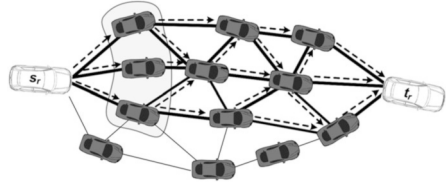


Fig. 10.20 Difference between (a) multipath and (b) anypath forwarding

² It is essential to note that different forwarding sets are generally used for different destination nodes.

Fig. 10.21 Example anypath between vehicles s_r and t_r marked with bold arrows. The respective forwarding set (shown for source node s_r toward node t_r) is marked with a gray area



neighboring nodes will next forward the packet toward the destination. This feature is to avoid unnecessary duplicate forwarding at each transit node.

A general rule is to assign higher priorities to relay nodes characterized by lower costs of paths toward the destination node. A packet will be forwarded by a certain lower-priority node only if it is not forwarded by all the respective higher-priority neighboring nodes (e.g., because they fail to receive the packet), determined by lack of MAC acknowledgment (i.e., ACK message) sent in a given timeslot by a higher-priority node upon receiving the packet [68]. The packet is lost only in case none of the nodes belonging to the forwarding set receive it [35].

In general, following [16], the cost of an anypath toward the destination node decreases with the increase of a number of forwarding relays. However, this may also imply increased transmission delay (too many nodes in the forwarding set may result in longer paths or even create loops). Therefore, the size of any forwarding set should be a trade-off between these two characteristics. Since under anypath communications, for each transit node, the probability of forwarding a packet successfully to at least one neighboring node is greater than the probability of delivering it to a specified forwarding node only [16, 68],³ the reliability of anypath communications is obviously greater than that of the unicast scheme.

However, each packet may traverse a multitude of possible paths (forming the anypath) to reach the destination since the rule for selecting the next hop is non-deterministic (Fig. 10.21). Therefore, a possible disadvantage of this opportunistic forwarding scheme can be route flapping due to choosing a particular route on a per-packet basis by the respective link- and network-layer protocol mechanisms.

In VANETs, anypath flapping can also be increased by frequent inter-vehicle link failures. Therefore, in this section, we focus on link stability as an essential factor to prevent route flapping in anypath communications. This problem is of significant importance, especially for several real-time safety services with stringent QoS requirements (e.g., safe driving assistance including real-time video transmission or emergency warnings [61]).

The concept of anypath communications in VANETs is relatively new, and the number of relevant proposals (e.g., [11, 29, 34, 37]) is limited. In particular, apart from our proposal from [53] presented in detail in the latter part of this section, there is practically no other approach available focusing on the reliability of anypath

³ Other benefits of anypath scheme utilization include reduced cost of retransmissions, improved throughput, and better energy efficiency.

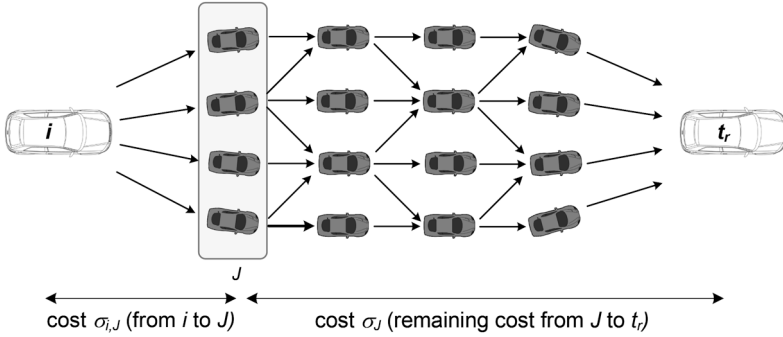


Fig. 10.22 Calculation of anypath total cost based on division into two costs

communications and, in particular, aimed at improving the stability of established anypaths.

In particular, Sect. 10.4.1 is to present (1) a definition of a scheme of long-lifetime anypath (LLA) routing that utilizes a new metric of link costs based on the introduced link stability index and (2) details of LLA solution deployment. Evaluation of algorithm characteristics is, in turn, presented in Sect. 10.4.2.

10.4.1 Long-Lifetime Anypath (LLA) Concept

When modeling point-to-multipoint link characteristic to anypath forwarding (see Fig. 10.22), the network is commonly represented by a hypergraph $\Gamma = (V, A)$, where V denotes the set of network vehicles, and A represents the set of hyperlinks, each hyperlink defined by an ordered pair (i, J) used to describe a given vehicle i connected with the forwarding set J of neighboring vehicles.⁴

The anypath cost between vehicles i and t_r can be defined by the Bellman equation given by formula (10.14), i.e., formed by the cost $\sigma_{i,J}$ of a hyperlink (i, J) from vehicle i to J and the remaining anypath cost σ_J from J to vehicle t_r [34].

$$\sigma_{i,t} = \sigma_{i,J} + \sigma_J \tag{10.14}$$

Following [16, 35], in the case of independent packet losses, the hyperlink cost $\sigma_{i,J}$ can be, in turn, defined as given in formula (10.15).

$$\sigma_{i,J} = \frac{1}{P_{i,J}} = \frac{1}{1 - \prod_{j \in J} (1 - p_{i,j})} \tag{10.15}$$

⁴ Under anypath routing, for each forwarding set J , indices $\{1, 2, \dots, n\}$ are assigned to nodes ascending the remaining path costs σ_j to destination node t_r (i.e., $\sigma_1 \leq \sigma_2 \leq \dots \leq \sigma_n$).

where $p_{i,J}$ denotes the probability of delivering the packet from node i to at least one node from J calculated based on individual probabilities of packet delivery $p_{i,j}$ obtained from Layer 2.

Another important meaning of $\sigma_{i,J}$ is that it represents the expected number of anypath transmissions required to successfully deliver the packet sent by node i to any node from J (see, e.g., definition of EATX metric from [16]).

The remaining cost σ_J of an anypath from J to t_r can be defined as the weighted average of costs of all paths from J to t_r as given in Eq. 10.16.

$$\sigma_J = \sum_{j \in J} w_{i,j} \sigma_j \quad (10.16)$$

where weight $w_{i,j}$ reflects the probability of node j being the forwarding node of a packet received from vehicle i , while σ_j represents the cost of a path between vehicle j from J and destination vehicle t_r [16].

Under the common simplified assumption of independent packet losses, $w_{i,j}$ values can be defined based on probabilities $p_{i,j}$ as given in Eq. 10.17.

$$w_{i,j} = \frac{p_{i,j} \prod_{k=1}^{j-1} (1 - p_{i,k})}{1 - \prod_{j \in J} (1 - p_{i,j})}, \quad \sum_i w_{i,j} = 1 \quad (10.17)$$

Mobility characteristics of vehicles play a crucial role in determining $p_{i,j}$ values. Therefore, for any time t_0 , future values of $p_{i,j}$ depend on the time-varying movement information of vehicles. For any time t_0 , any two connected vehicles i and j remain connected after Δt time units if distance $r_{i,j}$ between them at $t_0 + \Delta t$ remains in communications range $<0, r_{\max}>$, i.e.,

$$r_{i,j}(t_0 + \Delta t) = |\Phi_i(t_0 + \Delta t) - \Phi_j(t_0 + \Delta t)| \leq r_{\max} \quad (10.18)$$

where $\Phi_i(t_0 + \Delta t)$ is a position vector of vehicle i at $t_0 + \Delta t$ defined by Eq. 10.19.

$$\Phi_i(t_0 + \Delta t) = \Phi_i(t_0) + S_i(t_0, \Delta t) = \begin{bmatrix} x_i(t_0) \\ y_i(t_0) \end{bmatrix} + \begin{bmatrix} s_i^x(t_0, \Delta t) \\ s_i^y(t_0, \Delta t) \end{bmatrix} \quad (10.19)$$

$S_i(t_0, \Delta t) = [s_i^x(t_0, \Delta t), s_i^y(t_0, \Delta t)]^T$ is the movement vector of vehicle i .

Therefore, to reduce the effect of route flapping in anypath communications for consecutive packets, similar to Sect. 10.3, “stable links” (i.e., links between vehicles moving in similar directions with similar speeds) need to be identified and selected by the anypath calculation algorithm. We define the *stability index* $s_{i,j}$ of link (i, j) at any time t_0 as a value in range $<0; 1>$, as given in Eq. 10.20, i.e., as based on the

normalized increase of distance between vehicles i and j in the past $(t_0 - \Delta t, t_0)$ interval.

$$s_{i,j} = 1 - \frac{\min \left(\sqrt{(s_i^x(t_0 - \Delta t, t_0) - s_j^x(t_0 - \Delta t, t_0))^2 + (s_i^y(t_0 - \Delta t, t_0) - s_j^y(t_0 - \Delta t, t_0))^2}; r_{\text{upper}} \right)}{r_{\text{upper}}} \quad (10.20)$$

According to Eq. 10.20, the best value of stability index ($s_{i,j} = 1$) is assigned to links between nodes i and j characterized by equal movement vectors in the past $(t_0 - \Delta t, t_0)$ interval (i.e., implying no change in inter-vehicle distance). The worst value of $s_{i,j}=0$ is assigned to links that changed their length by more than the maximum assumed value r_{upper} in Δt time (based on the maximum allowed speed).

The probability of packet delivery at link destination nodes j in the near future (i.e., in $(t_0, t_0 + \Delta t)$ interval) is much influenced by link stability indices since probability $p_{i,j}$ of packet delivery between a pair of neighboring vehicles i and j is negatively correlated with link lengths [61]. Therefore, to limit the possibility of anypath route flapping, in this section, we propose to determine the cost of a link between any pair of neighboring vehicles i and j as given in Eq. 10.21, i.e., to include the value of stability index $s_{i,j}$.⁵

$$\xi_{i,j} = \frac{1}{p_{i,j} \cdot s_{i,j}} \quad (10.21)$$

Based on Eq. 10.21, the lowest cost $\xi_{i,j}$ (with a lower bound equal to 1.0) is assigned to links characterized by high values of stability index $s_{i,j}$ (i.e., links with estimated long lifetime), as well as high values of packet delivery ratio $p_{i,j}$ (specific for short links). Analogously, the respective costs $\sigma_{i,j}$ and weights $w_{i,j}$ are defined in our scheme as given in Eqs. 10.22–10.23.

$$\sigma_{i,j} = \frac{1}{1 - \prod_{j \in J} (1 - p_{i,j} s_{i,j})} \quad (10.22)$$

$$w_{i,j} = \frac{p_{i,j} s_{i,j} \prod_{k=1}^{j-1} (1 - p_{i,k} s_{i,k})}{1 - \prod_{j \in J} (1 - p_{i,j} s_{i,j})} \quad (10.23)$$

⁵ Similar to *end-to-end path reliability* being a product of delivery ratios $p_{i,j}$ of path links [33, 52], end-to-end transmission stability $S_{s,t}^r$ for demand d_r between source and destination vehicles s_r and t_r can be defined as a product of stability indices of all links of path η : $S_{s,t}^r = \prod_{(i,j) \in \eta} (s_{i,j})$.

Vehicle ID	X axis movement	Y axis movement
------------	-----------------	-----------------

Fig. 10.23 MOVEMENT messages of neighboring vehicles j stored at vehicles i

Details of LLA Approach Deployment

To enhance the anypath forwarding scheme with the proposed LLA functionality, it is necessary to implement a procedure to evaluate link stability indices ($s_{i,j}$). Necessary extensions are related to the periodic calculation of these values at each transit node i utilizing the MOVEMENT structures from Fig. 10.23, which should be stored at node i for each neighboring vehicle j . The respective X and Y axes movement values of neighboring vehicle j , stored in these structures, should be calculated at node i every Δt time units based on the default content of Cooperative Awareness Messages (CAMs) [18].

CAMs are commonly broadcast every 0.1–1 s by vehicles j via the Control Channel [34, 60]. In particular, CAMs include, by default, information on the vehicle's current location (X and Y coordinates) obtained from the Global Positioning System. To derive the individual link delivery ratios $p_{i,j}$ for $(t_0 - \Delta t, t_0)$ interval, a standard procedure of broadcasting the common Hello messages from each vehicle i via CCH followed by receiving the ACK messages from vehicles j [39] can be utilized.

Our algorithm of Long-Lifetime Anypath establishment (LLA), presented in Fig. 10.24, is based on the Shortest Anypath First (SAF) approach from [35]. In particular, to implement the LLA characteristics into the SAF approach, we need to replace the costs and weights from Eqs. 10.15 and 10.17 by the respective Eqs. 10.22–10.23. Due to the lack of other approaches similar to LLA in the literature, SAF is used as a reference technique in all comparisons presented in this section.

After performing the initialization Steps 1–2, each i -th iteration (Step 3 of LLA procedure) is to determine the final cost of anypath with respect to one transit vehicle j from N having the minimum value of σ_j .

Numerical Example

We are interested in finding the anypath between vehicles 1 and 7, as shown in Fig. 10.25a including example values of packet delivery probability $p_{i,j}$ and instant stability indices $s_{i,j}$ for the past $(t_0 - \Delta t, t_0)$ interval in the form of the ordered pairs $(p_{i,j}, s_{i,j})$. When executing the LLA algorithm, all costs σ_j are initially set to infinity. The only exception is for the cost σ_7 (referring to a destination vehicle 7), which is set to 0.

As shown in Figs. 10.25 and 10.26, the set of candidate next hops (relays) is formed in a way to minimize the cost σ_j . After establishing the anypath, general rules of anypath forwarding are utilized to deliver the packets to the destination node. In particular, relay priorities of vehicles j are determined for each forwarding set J , based on costs σ_j evaluated using our formulas (10.22) and (10.23).

INPUT	
–	set V of vehicles
–	a demand to establish the anypath between vehicles s_r and t_r
OUTPUT Anypath between vehicles s_r and t_r	
INDICES	
D	the set of nodes having the anypath to node t_r already defined
J	forwarding set
J_i	forwarding set for vehicle i to reach t_r
N	the queue of nodes that do not have the shortest anypath to t_r yet calculated (ordered ascending the σ_i values)
σ_j	the upper bound on the cost of the shortest anypath from j to t_r
Step 1 for each node i from V , set: $\sigma_i := \infty$; $J_i := \emptyset$	
Step 2 Set $\sigma_{t_r} := 0$; $D := \{t_r\}$; $N := V$	
Step 3 while $N \neq \emptyset$:	
	$j := \min_{k: \text{node } k \in N} \sigma_k$
	$D := D \cup \{j\}$
	for each incoming arc (i, j)
	$J := J_i \cup \{j\}$
	if $\sigma_i > \sigma_j$
	$\sigma_i := \sigma_{i,j} + \sigma_j$ (using Eqs. 10.21-10.23)
	$J_i := J$

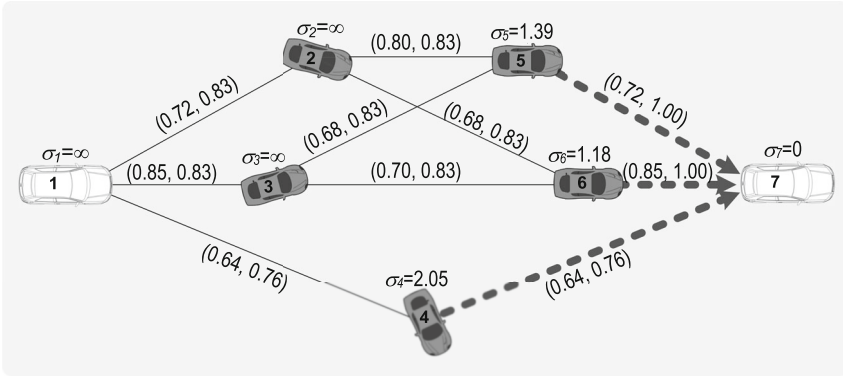
Fig. 10.24 LLA procedure

Execution of the LLA algorithm is terminated after $|V|$ iterations, i.e., after setting the final anypath cost σ_j to all nodes in the network. Assuming that the selection of a vehicle with the current minimum cost σ_j can be made in $O(\log|V|)$ steps (e.g., using the binary search), our LLA approach is characterized by the overall complexity bounded in above by $O(|V| \cdot \log|V|)$.

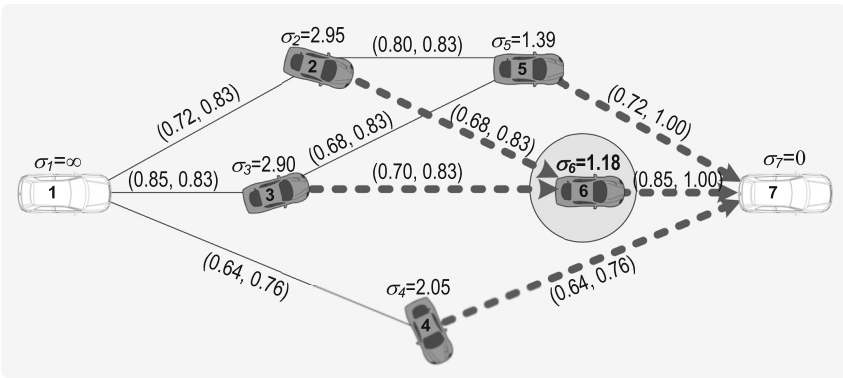
10.4.2 Analysis of Modeling Results and Conclusions

This section presents the results of the LLA approach evaluation, in particular, including the average values of path cost, hop count, message transmission delay, minimum and average path link stability, and end-to-end transmission stability. For each anypath, these characteristics are shown concerning its primary path (i.e., path of the lowest cost). The evaluation was performed for a realistic case of a 53-node network from Fig. 10.17 (i.e., the same one as in Sect. 10.3). In each of the 50 conducted experiments, the following assumptions were considered:

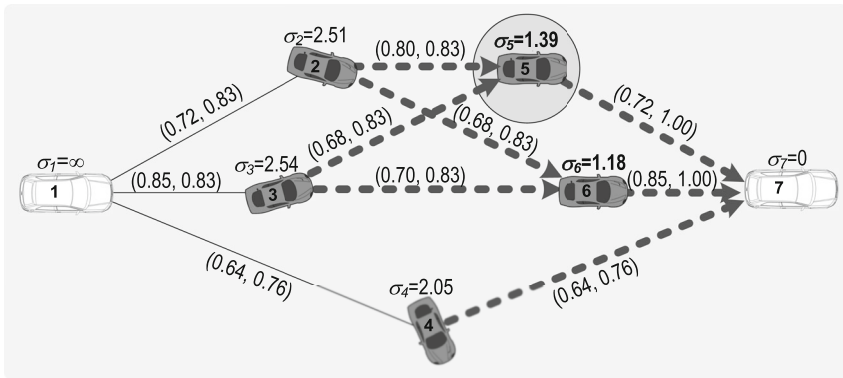
- The set of transmission demands comprised all pairs of vehicles.
- At time t_0 , cars were moving in directions compliant with the roadmap from Fig. 10.17.



(a)

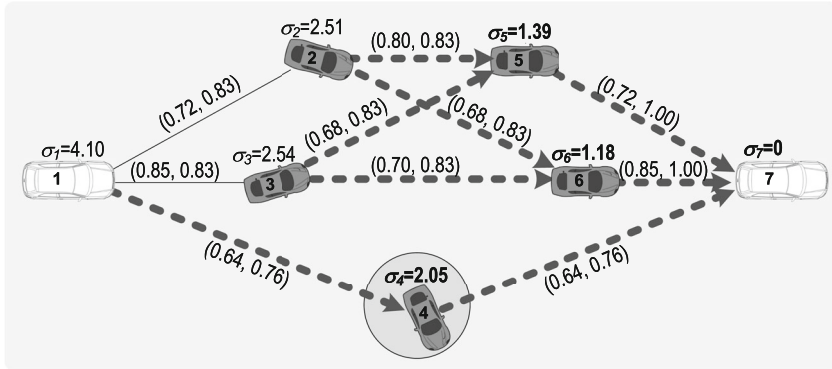


(b)

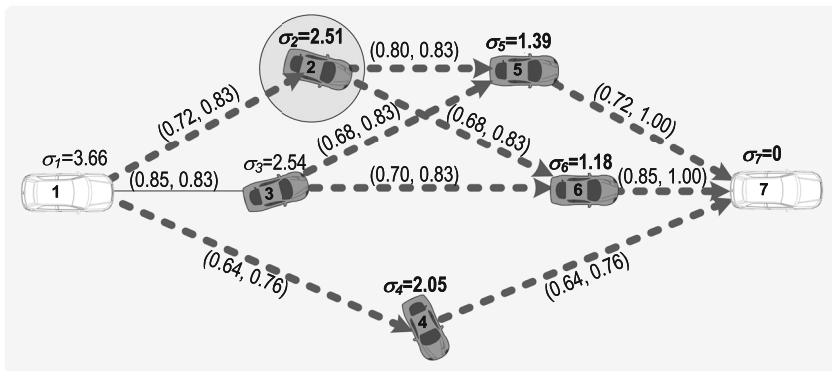


(c)

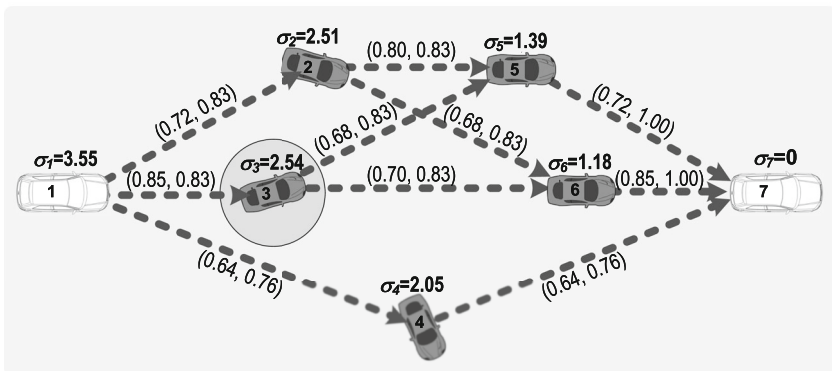
Fig. 10.25 Example execution steps of the LLA algorithm to determine the anypath between vehicles 1 and 7, including (a) initial graph with link stability indices $s_{i,j}$ and one-hop anypaths and (b)–(c) results of the first two successive iterations of LLA algorithm execution



(d)



(e)



(f)

Fig. 10.26 Example execution steps of the LLA algorithm to determine the anypath between vehicles 1 and 7, including the next three successive iterations of LLA algorithm execution

Algorithm		Path cost	Minimum link stability	Average link stability	End-to-end stability ($S_{e,l}$)	Hop count	Transmission delay [ms]
LLA	Average value	36.60	0.25	0.55	0.09	5.61	15.36
	Length of 95% confidence intervals	6.56	0.02	0.03	0.01	0.28	0.08
SAF	Average value	150.20	0.11	0.33	0.06	4.38	12.01
	Length of 95% confidence intervals	23.55	0.02	0.03	0.01	0.26	0.06

Fig. 10.27 Average values of obtained characteristics enhanced with 95% confidence interval analysis

- Linear velocities at time t_0 were uniformly distributed in the range of 0–16 m/s (based on common speed limitations in urban areas), with the maximum change r_{upper} of inter-vehicle distance in $\Delta t = 1$ s interval set to $r_{\text{upper}} = 16$ m.

Similar to Sect. 10.3, estimated movement vectors S_i of vehicles in the future $(t_0, t_0 + \Delta t)$ interval were calculated based on the respective ones referring to the past $(t_0 - \Delta t, t_0)$ interval, where $\Delta t = 1$ s. Due to negligibly small values of transmission delay times [30], network topology (including the location of vehicles and their speeds) was assumed to be “frozen,” i.e., it did not change during path computations. Results obtained for the LLA algorithm were next compared to the ones of the reference SAF algorithm from [35]. Following [61], formula (10.24) was used to estimate link delivery ratios $p_{i,j}$, while the calculation of path costs was realized according to introduced formulas (10.21)–(10.23) and based on metric from Eq. 10.14.

$$p_{i,j} = \begin{cases} 0.999, & \text{if } r_{i,j} \leq 400 \\ (-0.4x + 210)/100, & \text{if } 400 \leq r_{i,j} \leq 500 \\ 0.1, & \text{if } 500 \leq r_{i,j} \leq 600 \\ 0, & \text{if } r_{i,j} > 600 \end{cases} \quad (10.24)$$

Obtained average values of analyzed characteristics, together with the lengths of the respective 95% confidence intervals, are presented in Fig. 10.27. Results achieved by our LLA scheme concerning the total path cost were about 76% better than the reference ones’ characteristic for the SAF algorithm (36.60 against 150.20). Additionally, our LLA approach also achieved better ratios of minimum link stability (0.25 against 0.11), average link stability (0.55 against 0.33), and end-to-end stability (50% of improvement) of established anypaths. All these results confirmed the ability of the LLA approach to establish paths characterized by improved stability, as opposed to the common SAF technique. This is additionally shown in Fig. 10.28 presenting histograms of minimum and average values of link stability indices extended by the respective 95% confidence intervals.

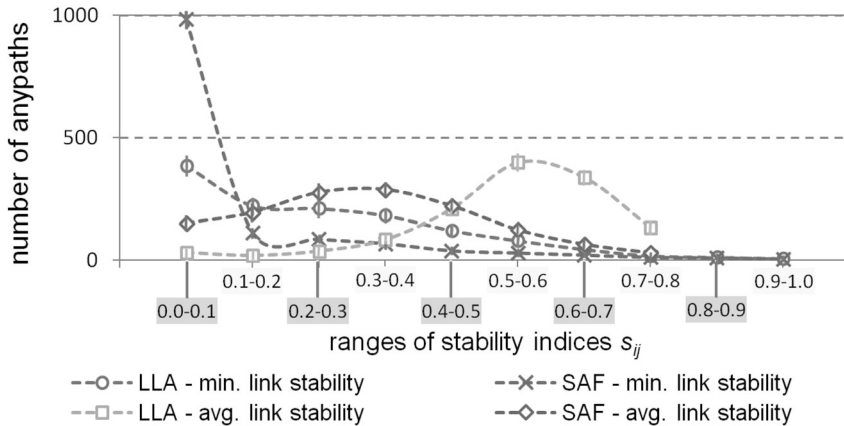


Fig. 10.28 Histogram of link stabilities

LLA advantages came at a price of increased length of transmission paths (they were about 28% longer, on average, compared to the SAF reference scheme), which implied only a small increase of message transmission delay of about 3.3 ms, on average.

10.5 Summary

As discussed in this section, the resilience of end-to-end multi-hop communications in vehicular ad hoc networks is a challenging issue due to the mobility of vehicles and the characteristics of DSRC links. In such a time-dependent environment, broadcasting commonly turns out to be the most efficient way to deliver messages to destination nodes. However, it frequently brings about an extensive load of VANET links, which is often unacceptable due to their relatively low capacity. Therefore, to limit the negative effect of broadcasting on network performance and, at the same time, improve the stability of end-to-end paths, in this chapter, we proposed two mechanisms of multipath and anypath forwarding enhanced with a selection of links based on current information related to link stability. Evaluation of the characteristics of the proposed methods showed that they can significantly improve the stability of end-to-end paths at a price of practically negligible increase in path length.

References

1. Alsabaan, M., Alasmary, W., Albasir, A., Naik, K.: Vehicular networks for a greener environment: a survey. *IEEE Commun. Surv. Tutorials* **15**(3), 1372–1388 (2013)
2. Amendment of the commission's rules regarding dedicated short-range communication services in the 5.850-5.925 GHz band (5.9 GHz band), Federal Communications Commission FCC 03-324 (2004)
3. Anaya, J.J., Merdrignac, P., Shagdar, O., Nashashibi, F., Naranjo, J.E.: Vehicle to pedestrian communications for protection of vulnerable road users. In: *Proceedings of the IEEE Intelligent Vehicles Symposium (IVS'14)*, pp. 1037–1042 (2014)
4. Bauza, R., Gozalvez, J., Sepulcre, M.: Power-aware link quality estimation for vehicular communication networks. *IEEE Commun. Lett.* **17**(4), 649–652 (2013)
5. Belyaev, E., Molchanov, P., Vinel, A., Koucheryavy, Y.: The use of automotive radars in video-based overtaking assistance applications. *IEEE Trans. Intell. Transport. Syst.* **14**(3), 1035–1042 (2013)
6. Belyaev, E., Vinel, A., Egiazarian, K., Koucheryavy, Y.: Power control in see-through overtaking assistance system. *IEEE Commun. Lett.* **17**(3), 612–615 (2013)
7. Blum, J.J., Eskandarian, A., Hoffman, L.: Challenges of intervehicle ad-hoc networks. *IEEE Trans. Intell. Transport. Syst.* **5**(4), 347–351 (2004)
8. Boukerche, A., Rezende, C., Pazzi, R.W.: A link-reliability-based approach to providing QoS support for VANETs. In: *Proceedings of the IEEE International Conference on Communications (IEEE ICC'09)*, pp. 1–5 (2009)
9. Briesemeister, L., Schaefer, L., Hommel, G.: Disseminating messages among highly mobile hosts based on inter-vehicle communication. In: *Proceedings of the IEEE Intelligent Vehicle Symposium (IVS'00)*, pp. 522–527 (2000)
10. Campolo, C., Molinaro, A., Vinel, A., Zhang, Y.: Modeling prioritized broadcasting in multichannel vehicular networks. *IEEE Trans. Veh. Technol.* **61**(2), 687–701 (2012)
11. Chachulski, Sz., Jennings, M., Katti, S., Katabi, D.: Trading structure for randomness in wireless opportunistic routing. In: *Proceedings of the ACM Annual Conference of the Special Interest Group on Data Communication (ACM SIGCOMM'07)*, pp. 169–180 (2007)
12. Chen, W., Cai, S.: Ad hoc peer-to-peer network architecture for vehicle safety communications. *IEEE Commun. Mag.* **43**(4), 100–107 (2005)
13. Chen, X., Li, L., Zhang, Y.: A Markov model for headway/spacing distribution of road traffic. *IEEE Trans. Intell. Transport. Syst.* **11**(4), 773–785, (2010)
14. Cholda, P., Mykkeltveit, A., Helvik, B.E., Wittner, O.J., Jajszczyk, A.: A survey of service resilience differentiation frameworks in communication networks. *IEEE Commun. Surv. Tutorials* **9**(2), 32–55 (2007)
15. Deb, B., Bhatnagar, S., Nath, B.: ReInForM: reliable forwarding using multiple paths in sensor networks. In: *Proceedings of the 28th IEEE Conference on Local Computer Networks (IEEE LCN'03)*, pp. 406–415 (2003)
16. Dubios-Ferriere, H., Grossglauser, M., Vetterli, M.: Valuable detours: least cost anypath routing. *IEEE/ACM Trans. Networking* **19**(2), 333–346 (2011)
17. El-atty, S.M.A., Stamatiou, G.K.: Performance analysis of multihop connectivity in VANET. In: *Proceedings of the 7th International Symposium on Wireless Communication Systems (ISWCS'10)*, pp. 335–339 (2010)
18. ETSI: Intelligent Transport Systems (ITS); Vehicular communications; Basic set of applications; Part 2: Specification of cooperative awareness basic service: http://www.etsi.org/deliver/etsi_ts/102600_102699/10263702/01.02.01_60/ts_10263702v010201p.pdf. Accessed on 25 Nov 2014
19. ETSI/ITS/102638, Intelligent Transport System (ITS); Vehicular communications; Basic set of applications; Definition, ETSI Std. ETSI ITS, Specification TR 102 638 version 1.1.1 (June 2009)

20. Federal Communications Commission: Standard specification for telecommunications and information exchange between roadside and vehicle systems – 5GHz band dedicated short range communications (DSRC) medium access control (MAC) and physical layer (PHY) specifications, ASTM E2213-01 (Sept. 2003)
21. Fukuhara, T., Warabino, T., Ohseki, T., Saito, K., Sugiyama, K., Nishida, T., Eguchi, K.: Broadcast methods for Inter-Vehicle Communication system. In: Proceedings of the IEEE Wireless Communications and Networking Conference (IEEE WCNC'05), vol. 4, pp. 2252–2257 (2005)
22. Harri, J., Filali, F., Bonnet, C.: Mobility models for vehicular ad hoc networks: a survey and taxonomy. *IEEE Commun. Surv. Tutorials* **11**(4), 19–041 (2009)
23. Hartenstein, H., Laberteaux, K.P.: A tutorial survey on vehicular ad hoc networks. *IEEE Commun. Mag.* **46**(6), 164–171 (2008)
24. Hartenstein, H., Bochow, B., Ebner, E., Lott, M., Radimirsch M., Vollmer, D.: Position-aware ad hoc wireless networks for inter-vehicle communications: the Fleetnet project. In: Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MobiHoc'01), pp. 259–261 (2001)
25. Huang, X., Fang, Y.: Performance study of node-disjoint multipath routing in vehicular ad hoc networks. *IEEE Trans. Veh. Technol.* **58**(4), 1942–1950 (2009)
26. Hui, J., Devetsikiotis, M.: A unified model for the performance analysis of IEEE 802.11e EDCA. *IEEE Trans. Commun.* **53**(9), 1498–1510 (2005)
27. IEEE Standards: <http://standards.ieee.org/findstds/standard/802.11p-2010.html> (2010). Accessed 21 Jul 2014
28. Jerbi, M., Senouci, S.-M., Rasheed, T., Ghamri-Doudane, Y.: Towards efficient geographic routing in urban vehicular networks. *IEEE Trans. Veh. Technol.* **58**(9), 5048–5059 (2009)
29. Jie, Z., Huang, Ch., Xu, L., Wang, B., Chen, X., Fan, X.: A trusted opportunistic routing algorithm for VANET. In: Proceedings of the 3rd International Conference on Networking and Distributed Computing Conference (ICNDC'12), pp. 86–90 (2012)
30. Jin, W.-L., Recker, W.W.: An analytical model of multihop connectivity of inter-vehicle communication systems. *IEEE Trans. Wireless Commun.* **9**(1), 106–112 (2010)
31. Karagiannis, G., Altintas, O., Ekici, E., Heijenk, G., Jarupan, B., Lin, K., Weil, T.: Vehicular networking: a survey and tutorial on requirements, architectures, challenges, standards, and solutions. *IEEE Commun. Surv. Tutorials* **13**(4), 584–616 (2011)
32. Karp, B., Kung, H.: GPSR: Greedy Perimeter Stateless Routing for wireless networks. In: Proceedings of the ACM Annual International Conference on Mobile Computing and Networking (MobiCom'00), pp. 243–54 (2000)
33. Khandani, A.E., Abounadi, J., Modiano, E., Zheng, L.: Reliability and route diversity in wireless networks. *IEEE Trans. Wireless Commun.* **7**(12), 4772–4776 (2008)
34. Kim, W., Oh, S.Y., Gerla, M., Lee, K.C.: CoRoute: a new cognitive anypath routing protocol. In: Proceedings of the 7th International Conference on Wireless Communications and Mobile Computing Conference (IWCMC'11), pp. 766–771 (2011)
35. Laufer, R., Dubois-Ferriere, H., Kleinrock, L.: Polynomial-time algorithms for multirate anypath routing in wireless multihop networks. *IEEE/ACM Trans. Networking* **20**(3), 742–755 (2012)
36. Lee, K.C., Gerla, M.: Opportunistic vehicular routing. In: Proceedings of the 16th European Wireless Conference (EW'10), pp. 873–880 (2010)
37. Leontiadis, I., Marfia, G., Mack, D., Pau, G., Mascolo, C., Gerla, M.: On the effectiveness of an opportunistic traffic management system for vehicular networks. *IEEE Trans. Intell. Transport. Syst.* **12**(4), 1537–1548 (2011)
38. Li, F., Wang, Y.: Routing in vehicular ad hoc networks: a survey. *IEEE Veh. Technol. Mag.* **2**(2), 12–22 (2007)
39. Li, T., Leith, D., Qiu, L.: Opportunistic routing for interactive traffic in wireless networks. In: Proceedings of the 30th International Conference on Distributed Computing Systems (ICDCS'10), pp. 458–467 (2010)

40. Ma, X., Yin, X., Trivedi, K.: On the reliability of safety applications in VANETs. *Int. J. Performability Eng.* **8**(2), 115–130 (2012)
41. Maihofer, C.: A survey of geocast routing protocols. *IEEE Commun. Surv. Tutorials* **6**(2), 32–42 (2004)
42. Manifesto of the Car-to-Car Communication Consortium, <http://www.car-to-car.org> (Sept. 2007). Accessed 22 Jul 2014
43. Marina, M.K., Das, S.R.: On-demand multipath distance vector routing in ad hoc networks. In: *Proceedings of the 9th International Conference on Network Protocols (IEEE ICNP'01)*, pp. 14–23 (2001)
44. Nagel, R.: The effect of vehicular distance distributions and mobility on VANET communications. In: *Proceedings of the IEEE Intelligent Vehicles Symposium (IEEE IVS'10)*, pp. 1190–1194 (2010)
45. Naumov, V., Gross, T.: Connectivity-aware routing (CAR) in vehicular ad-hoc networks. In: *Proceedings of the 26th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'07)*, pp. 1919–1927 (2007)
46. Nishida, T., Eguchi, K., Okamoto, Y., Warabino, T., Ohseki, T., Fukuhara, T., Saito, K.: Inter-vehicle P2P communication experimental on-board terminal. In: *Proceedings of the 2nd IEEE Consumer Communications and Networking Conference (IEEE CCNC'05)*, pp. 434–438 (2005)
47. Oka, H., Higaki, H.: Multihop data message transmission with inter-vehicle communication and store-carry-forward in sparse vehicle ad-hoc networks (VANET). In: *Proceedings of the New Technologies, Mobility and Security Conference (NTMS'08)*, pp. 1–5 (2008)
48. Ooi, Ch.-Ch., Faisal, N.: Implementation of geocast-enhanced AODV-Bis routing protocol in MANET. In: *Proceedings of the IEEE TENCON'04*, pp. 660–663 (2004)
49. Panichpapiboon, S., Pattara-Atikom, W.: A review of information dissemination protocols for vehicular ad hoc networks. *IEEE Commun. Surv. Tutorials* **14**(3), 784–798 (2012)
50. Perkins, C.E., Royer, E.M.: Ad-hoc on-demand distance vector routing. In: *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*, pp. 90–100 (1999)
51. Perkins, C., Belding-Royer, E., Das, S.: Ad hoc on-demand distance vector (AODV) routing. *IEFT RFC 3561* (2003)
52. Rak, J.: Providing differentiated levels of service availability in VANET communications. *IEEE Commun. Lett.* **17**(7), 1380–1383 (2013)
53. Rak, J.: LLA: a new anypath routing scheme providing long path lifetime in VANETs. *IEEE Commun. Lett.* **18**(2), 281–281 (2014)
54. Sermpezis, P., Koltsidas, G., Pavlidou, F.-N.: Investigating a junction-based multipath source routing algorithm for VANETs. *IEEE Commun. Lett.* **17**(3), 600–603 (2013)
55. Sitchitiu, M.L., Kihl, M.: Inter-vehicle communication systems: a survey. *IEEE Commun. Surv. Tutorials* **10**(2), 88–105 (2008)
56. Sun, W., Yamaguchi, H., Yukimasa, K., Kusumoto, S.: GVGrid: A QoS routing protocol for vehicular ad hoc networks. In: *Proceedings of the 14th IEEE International Workshop on Quality of Service (IEEE IWQoS'06)*, pp. 130–139 (2006)
57. Suthaputchakun, C., Dianati, M., Sun, Z.: Trinary partitioned black-burst-based broadcast protocol for time-critical emergency message dissemination in VANETs. *IEEE Trans. Veh. Technol.* **63**(6), 2926–2940 (2014)
58. Toor, Y., Muhlethaler, P., Laouiti, A.: Vehicle ad hoc networks: Applications and related technical issues. *IEEE Commun. Surv. Tutorials* **10**(3), 74–88 (2008)
59. Vehicle Safety Communications Project, Final Report, DOT HS 810 591, <http://www.nrd.nhtsa.dot.gov/pdf/surplus/nrd-12/060419-0843/> (April 2006). Accessed 21 Jul 2014
60. Vinel, A., Campolo, C., Petit, J., Koucheryavy, Y.: Trustworthy broadcasting in IEEE 802.11p/WAVE vehicular networks: delay analysis. *IEEE Commun. Lett.* **15**(9), 1010–1012 (2011)

61. Vinel, A., Belyaev, E., Egiazarian, K., Koucheryavy, Y.: An overtaking assistance system based on joint beaconing and real-time video transmission. *IEEE Trans. Veh. Technol.* **61**(5), 2319–2329 (2012)
62. Wakikawa, R., Sahasrabudhe, M.: Gateway management for vehicle to vehicle communication. In: *Proceedings of the 1st International Workshop on Vehicle-to-Vehicle Communications (2005)*
63. Wu, Ch.-Sh., Pang, A.-Ch., Hsu, Ch.-Sh.: Design of fast restoration multipath routing in VANETs. In: *Proceedings of the International Computer Symposium (ICS'10)*, pp. 73–78 (2010)
64. Yan, G., Olariu, S.: A probabilistic analysis of link duration in vehicular ad hoc networks. *IEEE Trans. Intell. Transport. Syst.* **12**(4), 1227–1236 (2011)
65. Yang, Q., Lim, A., Li, Sh., Fang, J.: ACAR: adaptive connectivity aware routing protocol for vehicular ad-hoc networks. In: *Proceedings of the International Conference on Computer Communications and Networks (ICCCN'08)*, pp. 1–6 (2008)
66. Ye, Z., Krishnamurthy, S.V., Tripathi, S.K.: A framework for reliable routing in mobile ad-hoc networks. In: *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM'03)*, pp. 270–280 (2003)
67. Yousefi, S., Altman, E., El-Azouzi, R., Fathy, M.: Analytical model for connectivity in vehicular ad hoc networks. *IEEE Trans. Veh. Technol.* **57**(6), 3341–3356 (2008)
68. Zeng, K., Lou, W., Zhai, H.: On end-to-end throughput of opportunistic routing in multirate and multihop wireless networks. In: *Proceedings of the 27th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM'08)*, pp. 1490–1498 (2012)
69. Zhang, W., Chen, Y., Yang, Y., Wang, X., Zhang, Y., Hong, X., Mao, G.: Multi-hop connectivity probability in infrastructure-based vehicular networks. *IEEE Sel. Areas Commun.* **30**(4), 740–747 (2012)

Conclusions

In this book, we focused on issues of resilient routing in networked systems. The ability of a system to maintain its capability to deliver information to the intended destinations in the presence of failures remains crucial, as it is not possible to eliminate all the factors triggering network element failures. Also, the number, intensity, and scale of failure events (particularly in the context of disaster-induced failures) are only predicted to increase. Therefore, failures, driven forces of nature, unintentional activities of third parties, or malicious attacks will continue to interrupt the normal functioning of any networked system. However, by appropriately applying preventive techniques, the negative impact of failures on network performance can be remarkably limited.

Discussions from Chaps. 1 and 2 conclude that to provide efficient means of prevention against disruptions, one must first correctly identify the challenges and faults potentially leading to failures based on characteristics of the network itself, as well as environmental factors favoring the occurrences of failures. In this context, it is also necessary to deploy a resilience strategy reflecting the target properties of system availability and reliability to make it feasible for a system to maintain an acceptable level of its services in failure scenarios. In this context, one may use, e.g., the D^2R^2+DR resilience strategy, which focuses on the implementation of mechanisms of defense, detection, remediation, and recovery to make the networked system able to survive the consequences of failures, as well as the related diagnosis and refinement phases to take advantage of the past failures in preparing for future failures.

When designing any networked system, it is crucial to make use of the respective element- and system-related metrics to assess the potential impact of failures of network elements on the performance of the entire system. These metrics, discussed in detail in Chap. 3 of this book, can be generally helpful in improving the resilience properties of the system architecture at practically all phases of system design, deployment, and update/evolution. The use of these metrics in the design of resilient

routing schemes may also enhance the properties of resilient routing mechanisms and, in particular, mitigate the impact of failures on routing.

As discussed in the second part of this book, the selection of an adequate strategy for resilient routing depends much on the transmission paradigm (connection-oriented vs. packet-switched) discussed in detail in Chaps. 4 and 5 and on the optimization goal (solution optimality vs. efficiency) covered by Chaps. 6 and 7. In these schemes, the resource efficiency objective by default contradicts the fast recovery objective (i.e., the faster the recovery to be achieved, the more backup resources need to be reserved).

An important observation is that the choice of an effective scheme of resilient routing may also depend much on the network architecture's individual characteristics, often making it particularly vulnerable to certain types of disruptions. For instance, heavy rain falls, despite bringing about a remarkable degradation of link capacities, e.g., in WMNs, in turn, have no impact on the respective links in wired (e.g., optical) networks.

To cover a broad range of issues in the design of resilient networked systems, in this book, we also presented three case studies on network resilience referring to the selected up-to-date network architectures. The first of them is the concept of the Internet of the Future. Due to a visible orientation of routing around content, common schemes to provide resilient routing based on utilizing backup paths were shown to require adaptation to make alternate paths access information often replicated at several network nodes. In Chap. 8, we introduced three routing schemes providing access to content after failures of network nodes. By applying the anycast routing, our methods based on the utilization of backup paths leading to different replica servers also provided protection in the case of a failure of a node hosting the content (which is commonly not possible for classical unicast communications). Proposed variants included scenarios of dedicated and shared protection against random failures and dedicated protection under attacks.

Chapter 9 focused on the continuity of end-to-end transmission under failures affecting high-frequency links in Wireless Mesh Networks. Indeed, due to high-frequency communications, WMN links are very susceptible to rainfalls. As a result, the effective capacity of WMN links can be seriously degraded. To provide the appropriate solutions to improve WMN resilience, we first introduced the measures of WMN survivability necessary to evaluate the vulnerability of WMN topologies to disruptions (e.g., weather-based) occurring in bounded areas leading to multiple correlated failures. These measures were also designed to help design the WMNs with improved resistance to region failures.

A second contribution of Chap. 9 was a networking concept to adapt the structure of a WMN to changing weather conditions by periodic updates of antenna alignment based on the forecasted heavy rainfalls following information from radar echo rain maps. The objective was to avoid creating direct links between WMN nodes over areas with predicted heavy signal attenuation. As verified using simulations for real rain scenarios, average signal attenuation could be significantly reduced compared to the reference scheme, which did not apply any changes to WMN antenna alignment.

Chapter 10 focused on resilience issues in wireless mobile networks organized ad hoc around vehicles (VANETs). In this case, wireless links often encounter availability problems related to the high mobility of vehicles, visibly reducing the link lifetime and the lifetime of end-to-end communication paths. VANETs are expected to improve road safety (e.g., by messages exchanged in the case of accidents or bad weather conditions), traffic coordination (e.g., to help the drivers move in the green phase), and provide travelers with infotainment services. To work effectively, VANETs need reliable schemes of message dissemination, particularly resistance to mobility-based link disconnections.

To address this issue, in Chap. 10, we proposed two schemes of end-to-end routing that focus on establishing end-to-end communication paths with increased lifetime. This was achieved by a dedicated link cost metric that utilizes information on the predicted stability of VANET links (based on actual movement information). Two proposed routing schemes based on multipath and anypath forwarding resulted in a notable increase in the stability of each primary transmission path.

The deployment of resilient routing mechanisms relying on the use of backup paths commonly introduces extra cost, as additional backup paths are needed to maintain the transmission ability in failure scenarios. However, the cost of resilience can be remarkably reduced by applying the related mechanisms of backup path sharing. Other means of cost reduction include, e.g., serving low-priority traffic during normal operation of a system using link capacities reserved originally for backup paths, the use of renewable energy sources, all-optical communications (to avoid energy-inefficient signal conversions between the optical and electrical domains at system nodes), or the use of sleep mode for backup paths.

Also, resilience often happens to come at no extra cost, especially when certain forms of redundancy already exist in networked systems, whether or not increasing their resilience was the main reason for their implementation. Examples include, e.g., the deployment of multiple copies of servers originally aimed at ensuring the scalability of services.

Investing in network resilience mechanisms can clearly provide remarkable and indisputable benefits. In particular, resilience allows to avoid substantial financial losses for both users and service providers in failure scenarios. Mechanisms of resilience can also lower the risk of improper functioning of networked systems and services provided by them in scenarios of malicious human activities, events of natural disasters, and other failures triggered unintentionally.

The proper functioning of networked systems can even save human lives since public communication networks often turn out to be the only means of communication in disaster scenarios, also for authorities disseminating, e.g., the rescue information to citizens. All these benefits make it clear that resilience mechanisms should be considered an integral aspect of the design of any networked system.

Glossary

- 1 + 1 protection** A transmission scheme in which traffic is transmitted in a normal operational network state in parallel over two link-/node-disjoint paths, one of which takes the role of the only valid path if the other one fails
- 1:1 protection** A path protection scheme assuming usage of a backup path only after a failure of a node/link affecting the primary path
- Accidental fault** A fault that was created (or appeared) fortuitously
- Active Path First (APF)** A scheme of establishing the pair (or the set) of end-to-end disjoint paths of a demand assuming that calculation of the primary path is done first and is followed by determination of backup path (or backup paths) over the topology of a residual network—i.e., after excluding the arcs traversed by the primary path (for link disjointness), or arcs incident to transit nodes of the primary path (for nodal disjointness)
- Ad hoc On-demand Distance Vector (AODV)** A reactive routing protocol developed for wireless ad hoc networks to establish transmission paths on demand (using Route Request and Route Response messages) and maintaining them as long as they are necessary
- Ad hoc network** A wireless network of a decentralized type not relying on fixed infrastructure, with data forwarding provided by each network node in a dynamic way subject to instantaneous network connectivity
- Add-Drop Multiplexer (ADM)** A wavelength division multiplexing device used for routing and multiplexing/demultiplexing (i.e., adding/dropping) of different channels of light into or out of a single-mode fiber
- Adjacency matrix** A square matrix with binary elements $\hat{a}_{i,j}$ set to 1 in the case of the existence of a communication link from network node n_i to network node n_j and 0 otherwise
- Ageing-related bugs** Software faults that got accumulated over time
- Alternate path** A backup transmission path used as the only path after a failure of a network element (node/link) affecting the primary transmission path

- Anycast routing** A one-to-one-of-many transmission scheme allowing for accessing the content at one of many potential servers, each one storing a copy (also called a replica) of the original content
- Anypath routing** A transmission scheme utilized, e.g., in VANETs where the set of neighboring nodes (called the forwarding set) acts in a cooperative manner to forward each packet toward the destination node
- Asynchronous Transfer Mode (ATM)** A telecommunications concept defined by ITU-T in the late 1980s for carriage of a diverse set of voice, data, and video signals (i.e., designed to unify telecommunication and computer networks), providing functionality similar to both circuit switching and packet switching network architectures
- Auditability** Assessment of whether the communication system is safeguarding information, maintaining data integrity, as well as operating in a way to achieve the goals/objectives of the organization
- Augmented model** A multilayer network scheme being an extension to the overlay model of cooperation between network layers that makes information about nodes reachability available at the UNIs
- Authenticity** Assurance that the considered principals are exactly who they claim to be
- Authorisability** Assurance that the considered elements of a system are accessed according to granted permissions
- Automatic Protection Switching (APS)** A transmission scheme involving establishing a dedicated/shared protection path of the same capacity as the primary path to be protected
- Availability** The readiness for usage of a given service at time t
- Average content accessibility** A functional metric used to evaluate the possibility of delivering the anycast traffic in scenarios of massive failures implied by disaster events
- Average node degree** A structural metric providing information on the density of the network topology, defined as the average number of links incident to nodes in the considered system
- Average shortest path length** A structural metric providing information on the average distance (or the number of links) along the shortest paths calculated considering all pairs of source and destination nodes in the networked system
- Average two-terminal reliability (ATTR)** A structural metric defined as the total number of pairs of nodes in all system components of the system divided by the total number of node pairs in the system
- Backbone network** The core part of a communication network infrastructure interconnecting other parts of the network and different networks
- Backup path** See “alternate path”
- Benign failure** Failure consequences of which are similar in scale to the benefit from the correct functioning of service
- Best-effort delivery** A network service that does not offer any guarantee on data delivery or that a user is provided with a predefined level of QoS/priority

- Betweenness centrality (BC)** A metric of a network node centrality defined in terms of a number of the shortest paths that traverse the considered node and, therefore, an essential indicator of a node's vulnerability to attacks
- Bidirectional Line Switched Ring (BLSR)** A ring network providing protection against failures by offering two transmission rings (for working and backup paths, respectively)
- Bidirectional network link** A link enabling transmission in both directions and often characterized by the same capacity in both directions
- Binary Linear Programming (BLP)** A paradigm of solving optimization problems, in which the objective function and constraints are linear, while variables are restricted to be binary
- Binary Nonlinear Programming (BNLP)** A paradigm of solving optimization problems in which either the objective function or some of the constraints are nonlinear, while variables are restricted to be binary
- Bit Error Rate (BER)** A number of bit errors per total number of bits transferred
- Bohrbug** A software fault that can be easily detected
- Bottom-up recovery** A recovery scheme in a multilayer network where recovery actions concerning the affected flows are initiated in the lowermost layer and are then continued in the upper layers
- Broadcasting** Transmission of information to every node located within direct reach of a sender
- Car-to-Car Communications Consortium (C2C-CC)** A nonprofit industrial organization driven by European vehicle manufacturers and supported by equipment suppliers and research organizations to increase the safety and efficiency of road traffic using inter-vehicular wireless communications
- Cascading failure** A failure of multiple network elements triggered by the initial failure (e.g., failures of network nodes as a result of power outage implied by an earthquake)
- Catastrophic failure** A failure of a significant part of the system from which recovery is hardly possible
- Central node** A network node switching large amounts of data characterized by one of the highest degrees in the network
- Central Processing Unit (CPU)** An electronic circuitry carrying out arithmetic, logical, control, and input/output (I/O) operations specified by the instructions
- Challenge** A characteristic/condition that may occur as an event affecting the normal operation of a network
- Challenge probability** Probability of a challenge occurrence
- Challenge tolerance** A network resilience category focusing on network design approaches to provide service continuity in the presence of challenges
- Class of Service (CoS)** A parameter utilized to identify the type of a packet payload to provide differentiated transmission services to packets based on assigned priorities
- Clean-slate** A concept of deploying new solutions under the assumption that other parts of the network architecture remain unchanged

- Closeness centrality** A metric defined for a node to reflect its average distance to all the other nodes in the system
- Cloud computing/communications** A computing/communications paradigm based on the utilization of computer resources combining the global-scale resource centers and computation possibilities into the cloud to form a “computing utility” available over the Internet
- Clustering coefficient** A structural metric defined for a system topology as the ratio of the number of closed triplets over the total number of open and closed triplets. It is used to evaluate the scale of cluster formation by nodes in the system topology
- Coexistence (of virtual networks)** Parallel existence of multiple virtual networks over the same resources of one or several infrastructure providers
- Common pool** Technique of sharing the backup resources in a multilayer network in a way that the respective protection (backup) paths from different layers do not share the risk of being activated at the same time
- Complete failure** A failure referring to the entire network element
- Confidentiality** Assurance of not disclosing information without proper authorization
- Consistent failure** An event considered a failure by all users
- Content-Aware Networking (CAN)** A paradigm of network intelligence to identify, based on incoming request to access the content, where to find it and how to deliver it
- Content-Centric Networking (CCN)** See “Content-Aware Networking”
- Content Delivery Network (CDN)** A distributed system of interconnected data centers to provide the end users with content at high availability and performance guarantees
- Content-Oriented Networking (CON)** An opposite solution to the conventional host-to-host information delivery, shifting the issues of item identification from hosts to information (i.e., making information rather than conventional IP addresses the primary search goal); see “Content-Aware Networking”
- Control Channel (CCH)** A communication channel in VANETs used to transmit the control messages
- Cooperative Awareness Message (CAM)** Information broadcasted periodically once every 0.1–1 s by a vehicle in VANETs to inform other vehicles, e.g., about its current location
- Correlated failures** Concurrent failures of multiple network elements being interdependent (as, e.g., in the region failure scenario)
- Critical information infrastructure** An information system that is essential for the functioning of a society and economy
- Critical latency** The upper bound on message delivery latency
- D-geodiversity** A property of a given set of communication paths for a given demand such that each transit element of a given path is located in a geographical distance of at least D from any other transit node of any of these paths
- Data dissemination** The transportation of data to the intended recipients while satisfying certain requirements, such as, e.g., delay

- Dedicated protection** A resilient communication scheme based on the assignment of backup paths exclusively for a given working path
- Dedicated Short-Range Communications (DSRC)** Specification of short range to medium-range wireless communication channels for use in inter-vehicular communications
- Degree centrality** A metric of a single node defined as the degree of that node
- Delay** A QoS attribute defined concerning the transmission of information as an interval between given two-time limits determined in various ways (e.g., concerning the time needed for a message to be transmitted end-to-end over the network)
- Delay-tolerant transmission** A transmission scheme not requiring real-time data delivery
- Demand volume** The amount c_r of link capacity requested by demand d_r for a communication path between a given pair of source node s_r and destination node t_r
- Dense Wavelength Division Multiplexing (DWDM)** An optical transmission scheme originally related to optical signals multiplexed within the 1550 nm band, allowing for the coexistence of many independent transmission channels per link
- Dependability** A discipline used to quantify the level of service reliance of a system, i.e., a property of a system such that reliance can justifiably be placed on the service it delivers
- Dependent failure** A failure that is implied by another failure
- Development fault** A fault that arose during the phases of system design, deployment, and modification and when defining the respective procedures for operating the system
- Diameter** A structural metric defined as the minimum hop count between any pair of nodes in the system
- Directional network link** A link enabling transmission in a given direction
- Disaster-based failure** A failure of the network element(s) implied by the occurrence of a disaster of any kind, including natural disasters, technology-related disasters, and malicious attacks
- Disjoint paths** A set of end-to-end paths having no common links (for link disjointness) or no common transit nodes (for nodal disjointness)
- Disruption tolerance** The ability of communication paths to survive disruptions in connectivity among its components
- Dissemination of data/messages** A Layer-2 transmission scheme utilized, e.g., in VANETs, to deliver messages frequently via multiple hops based on single-hop broadcasting
- Distributed Denial of Service (DDoS)** An attempt performed in a distributed way (e.g., by multiple parties) to make the network node resources unavailable to end users mostly by interrupting the services of a host
- Distributed faults** Faults spread across multiple locations
- Diversity** A networking paradigm aimed to assure that the same flaw does not affect multiple elements of a communication system

Domain Name System (DNS) A hierarchical distributed naming system to associate information such as IP addresses with domain names assigned to the considered network nodes

Dual-cost network A scheme with differentiated costs assigned to network links in computations of two disjoint paths of the same demand

Dynamic hardware redundancy Redundancy at the hardware level involving the use of additional elements of a network system in failure scenarios, e.g., to bypass the failed element, as, e.g., in the case of a failure of a communication network node triggering a recovery procedure to activate the respective detours for the affected network traffic

E.800 ITU-T recommendation “Definitions of terms related to Quality of Service”

E.802 ITU-T recommendation “Framework and methodologies for the determination and application of QoS parameters”

E.820 ITU-T recommendation “Call models for serveability and service integrity performance”

E.850 ITU-T recommendation “Connection retainability objective for the international telephone service”

E.855 ITU-T recommendation “Connection integrity objective for the international telephone service”

E.860 ITU-T recommendation “Framework of a service level agreement”

E.862 ITU-T recommendation “Dependability planning of telecommunication networks”

E.880 ITU-T recommendation “Field data collection and evaluation on the performance of equipment, networks and services”

Edge connectivity A structural metric defined similarly to node connectivity as the smallest number of edges from G whose removal leads to system partitioning

Efficiency A structural metric focusing on the inverse values of the number of links of the shortest paths in the networked system, useful in determining how quickly information can be transmitted between any pair of end nodes in the system

Eigenvector centrality A metric used to evaluate the influence of a given node in the network, defined as the value of the i -th element of the eigenvector referring to the largest eigenvalue λ_1 calculated for the adjacency matrix

Electromagnetic pulse (EMP) attack A malicious activity based on a transient electromagnetic disturbance via a short burst of electromagnetic energy

Element-related metric A metric focusing on the properties of individual network elements (nodes/links) following from their existence in the system topology

End-to-end delay A packet-level metric used to determine the total propagation time for a message to travel via all consecutive links of the transmission path between the source and destination nodes

End-to-end routing Transmission of information from the source node toward the destination node frequently over multiple transit nodes

Error A deviation between the observed value/state and its specified (correct) value/state

European Telecommunications Standards Institute (ETSI) A nonprofit telecommunications standardization organization issuing standards for Information and

Communications Technologies (fixed, mobile, radio, converged, broadcast, and Internet technologies)

Event-driven notifications/messages Information sent after identification of an event

Expected transmission count (EXT) A metric used to evaluate the loss rate of packets between neighboring nodes

External fault A fault originating from interactions with the physical/human environment of a networked system

Failure (of network services) An event occurring when the delivered service deviates from the correct service

Failures in time (FIT) The number of failures per billion device hours

Fast reroute A resilience scheme (characteristic of MPLS and IP networks) for fast recovery of traffic by redirecting it locally over the failed network element; see “local protection”

Fault A flaw being either an accidental design flaw (for instance, a software bug) or an intentional flaw not eliminated, for example, due to the cost constraints of the system

Fault avoidance A set of activities to specify, verify, and derive the fault-free software

Fault detection An activity leading to the determination of fault in real-time either in the physical layer (e.g., due to loss of signal, loss of modulation, or loss of clock) using signal degradation recognition (e.g., increased bit error rate—BER) or Quality of Service degradation (indicated by decreased throughput or increased transmission delay)

Fault localization Network activity aimed at determination of the point of fault occurrence

Fault notification Network activity necessary to start redirection of the affected traffic onto the alternate paths

Fault removal Activities leading to the removal of faults from existing software products

Fault tolerance Ability of a communication system to cope with faults being the result of events other than service failures

Federal Communications Commission (FCC) An agency of the United States government aimed to regulate the US interstate communications by radio, television, wire, satellite, and cable focusing on broadband, competition, spectrum, media, public safety, as well as homeland security issues

Five nines property Guarantee on a communication system availability of at least 99.999%

Fixed addressing (in VANETs) A scheme of assigning the address to a VANET node once it joins the network, which remains unchanged until the node leaves the network

Flow p -cycle A scheme where a single protection cycle protects a certain segment of the working path

Forwarding set (in VANETs) A set of VANET neighboring nodes used in anypath communications to forward the packet toward the destination node

- Free capacity** Capacity of a link not assigned to any communication path
- Free-Space Optical (FSO)** An optical communication technology with light propagation in free space for wireless transmission of data
- Full restoration time** Time required for traffic to be routed onto links that are capable of or have been engineered sufficiently to handle traffic in recovery scenarios
- Functional metric** A metric used to analyze the system quality of service
- Future Internet (FI)** A set of relevant capabilities of the global communications infrastructure not existing in the current Internet architecture
- Future Internet Assembly (FIA)** A European forum organized once/twice a year for a collaboration between members of FI projects to maintain European competitiveness in the global marketplace
- G.911** ITU-T recommendation “Parameters and calculation methodologies for reliability and availability of fiber optic systems”
- Generalized Multiprotocol Label Switching (GMPLS)** An extension to MPLS to manage additional classes of interfaces and switching technologies such as TDM, Layer-2 switching, wavelength switching, or fiber switching
- Geocasting** See “geographical addressing”
- Geographical addressing** A scheme of address assignment based on the location of a mobile node (frequently used, e.g., in VANETs, where an address of a VANET node changes as the vehicle moves—not necessarily leaving the network)
- Global Positioning System (GPS)** A space-based satellite navigation system to offer location and time information anywhere on the Earth (or near the Earth) provided that there is an unobstructed line of sight to at least four GPS satellites
- Global protection** See “path protection”
- Global recovery (protection) scheme** A resilience scheme assuming utilization of a single backup path providing the end-to-end protection concerning a given primary path
- Graph diversity** A structural metric used to analyze the frequency of traversing the same communication links and transit nodes by communication paths between given pairs of end nodes
- Graph of conflicts** A graph with vertices modeling objects of a given kind interconnected by edges representing the conflict states concerning the vertices
- Hamiltonian p -cycle** A protection cycle traversing each node of a network exactly once
- Hardware fault** A fault referring to the hardware elements of a system
- Hardware redundancy** A technique of using additional hardware components to provide fault tolerance
- Heisenbug** A software fault being elusive and whose behavior often alters while being researched
- Heterogeneity** A structural metric representing the level of inhomogeneity of node degrees, defined as the standard deviation of degrees of nodes in the system divided by the average node degree
- High-degree node** A network node connected to many other nodes via direct links

Hold-off timer A recovery mechanism designed for multilayer networks to postpone the recovery actions in the higher layer to give the lower-layer time for recovery of the affected traffic

Host-centric communications Conventional communications scheme assuming that named hosts are the main network entities to be addressed

Human-made fault A fault that is a result of human activities/imperfections

Hypertext Transfer Protocol (HTTP) A common application protocol for hypermedia information systems—the primary protocol for data communications for the World Wide Web

IEEE 802.11 A set of specifications referring to MAC and PHY layers addressing implementation issues of wireless local area network communications developed and maintained by the IEEE LAN/MAN Standards Committee (IEEE 802)

Incidence matrix A matrix providing information on the neighborhood relation of network nodes and links. In this matrix, a given row i (out of n rows) refers to node n_i , while column m is associated with m -th network link. An element in i -th row and m -th column of this matrix is set to 1, if a link m incident to node n_i exists and 0 otherwise

Inconsistent failure An event not considered a failure by all users

Information-Centric Networking (ICN) See “Content-Oriented Networking”

Infotainment A group of VANET applications providing travelers with on-board information and entertainment services such as Internet access or music download

Infrastructure provider (InP) An entity managing the physical infrastructure of networks

Inheritance Characteristics of a virtual network allowing the child virtual networks to inherit the architectural attributes of their parent virtual networks

Integer Linear Programming (ILP) A paradigm of solving optimization problems, in which the objective function and constraints are linear, while variables are restricted to be integers

Integer Nonlinear Programming A paradigm of solving the optimization problems, in which either the objective function or some of the constraints are nonlinear, while variables are integer

Integrated (peer) recovery model A multilayer network resilience scheme allowing for sharing of routing information between network layers

Integrity The absence of improper (unauthorized) system alterations

Intentional fault A fault being the result of deliberate activities being either malicious or non-malicious

Interdependency A relation between systems implying the interrelation of systems necessary for them to be able to operate

Inter-domain recovery A recovery scheme (e.g., based on utilization of alternate paths) that involves resources from multiple network domains

Intermittent fault A temporary internal fault due to malfunctioning of devices following, e.g., from changes of parameter values of hardware components such as their temperature

- Internal fault** A fault referring to parts of the system state leading to errors when invoked
- International Federation for Information Processing (IFIP)** A nonprofit organization working in the field of information technology, focusing on sponsoring and organizing conferences and workshops in the area of Information and Communications Technology
- International Telecommunication Union—Telecommunication Standardization Sector (ITU-T)** One of the units of ITU responsible for coordination of telecommunication standards
- Internet Engineering Task Force (IETF)** The open standards organization without formal membership requirements established to develop Internet standards voluntarily, in particular referring to the TCP/IP protocols family
- Internet of Things (IoT)** A network of physical objects (“things”) commonly embedded with electronics, sensors, and software and therefore provided with the ability to exchange information with other connected devices (or the manufacturer/operator)
- Internet Protocol (IP)** The primary communications protocol in the set of Internet protocols responsible for relaying datagrams across communication networks (i.e., routing)
- Internet Protocol version 6 (IPv6)** The latest version of the Internet Protocol (intended to replace IPv4) developed by IETF, e.g., to solve the problem of IPv4 address exhaustion
- Internet Service Provider (ISP)** Commercial, community-owned, nonprofit, or privately owned entity offering services related to participating in the Internet
- Inter-Vehicular Communications (IVC)** A type of wireless communications between vehicles and roadside units to exchange information (e.g., safety- and traffic-related)
- IP Packet Delay Variation (IPDV)** The end-to-end one-way delay difference between consecutive packets in a flow in an IP network (with any lost packets being disregarded)
- IP Packet Error Ratio (IPER)** The number of packets being incorrectly received in an IP network divided by the total number of received packets
- IP Packet Loss Ratio (IPLR)** The number of lost packets divided by the total number of sent packets
- IP Packet Transfer Delay (IPTD)** The aggregate value of end-to-end store-and-forward delays a packet encounters in each transit node before being received by the destination node (i.e., depending on network congestion and the number of transit routers along a transmission path)
- Jitter** A deviation from the assumed periodicity of packet delivery being a metric of the variation of latency
- Jitter-sensitive transmission** A transmission scheme that does not tolerate jitter concerning consecutive packet delivery
- Label Switched Path (LSP)** A communication path set up by a signaling protocol in an MPLS network

- Large-scale testbed** A communication infrastructure of a large (e.g., national/continental) scale deployed to validate the proposed global communications solutions
- Largest first** A heuristic algorithm for graph coloring used to assign colors (integer numbers) to vertices of a given graph in a way that any two neighboring vertices receive different colors. In this algorithm, colors are assigned to vertices ordered descending their degrees
- Latency** See end-to-end delay
- Lightpath** A multi-hop optical path providing end-to-end connectivity in the optical network
- Line of Sight (LOS) propagation** A characteristic of electromagnetic radiation with emissions of light traveling along a straight line
- Linear Programming (LP)** A paradigm of solving optimization problems, in which the objective function and constraints are linear, and variables are continuous
- Link betweenness centrality** An index of betweenness centrality defined for network links to reflect the importance of that link in shortest path multi-hop communications
- Link downtime** A period of link unavailability
- Link-path formulation** Formulation of an optimization problem with variables referring to a set of precomputed paths traversing the network links
- Link protection** A resilient routing scheme assuming protection of each link of a working path by a separate backup path installed in advance (prior to a failure)
- Link restoration** A resilient routing scheme assuming protection of each link of a working path by a separate backup path established after the occurrence of a failure
- Link-State Advertisement (LSA)** A basic communication methodology of the OSPF routing protocol in which network nodes periodically distribute information related to the current characteristics of incident links
- Link stress** A functional metric used to evaluate the efficiency of overlay networks, as it calculates the number of times packets traverse the same physical link
- Link utilization** A functional metric providing information on the percentage of the total (i.e., nominal) capacity used for data transmission
- Local faults** Faults in a single location of a system architecture
- Local protection** See “link protection”
- Local recovery (protection) scheme** A recovery scheme assuming utilization of a backup path designed to redirect the affected traffic over the failed link/node (i.e., short detours)
- Local-to-egress protection** A resilience scheme in MPLS networks involving a backup LSP configured in the reverse direction from the last-hop working LSP node toward the working LSP source node and next back to the destination node of a working LSP via a path being node-disjoint with the related working LSP
- M.3342** ITU-T recommendation “Guidelines for the definition of SLA representation templates”
- M.60** ITU-T recommendation “Maintenance terminology and definitions”
- Maintainability** Predisposition of a system to updates/evolution

- Malicious attack** Any malicious activity (usually originating from an anonymous source) driven by individuals/organizations aimed at causing significant losses concerning target information systems, infrastructures, computer networks, and/or personal computer devices
- Mandelbug** A software fault characterized by complex underlying causes, chaotic and even nondeterministic behavior
- Mean content accessibility** A functional metric used to evaluate the robustness of the networked system concerning the delivery of anycast traffic by taking into consideration a broad range of disasters
- Mean Downtime (MDT)** The mean time of service inaccessibility
- Mean Opinion Score (MOS)** A subjective metric used to evaluate the quality of experience perceived by users. It is calculated as the arithmetic mean value of individual subjective scores given by users
- Mean Time Between Failures (MTBFs)** The mean time between the beginning of two consecutive failures of a service
- Mean Time Between Maintenance (MTBM)** The mean time between the beginning of two consecutive periods of (scheduled) preventive maintenance activities
- Mean Time to Failure (MTTF)/Mean Time to First Failure (MTFF)** The length of a period between a point when the service was initiated until its failure (for a system element, the period between time t when the element was put into operation until its (first) failure)
- Mean Time to Recovery (MTTR)** The mean value of the length of a period between the occurrence of a failure and the successful completion of a recovery action. The mean time spent purely on repair operations (excluding the time between the occurrence of a failure and the beginning of a repair period) is often called the *Mean Time to Repair*
- Mean Uptime (MUT)** The mean time between a successful restoration of a service and the time of the occurrence of the next service failure
- Media Access Control (MAC)** A sublayer of Layer 2 (data link layer) responsible for proper addressing and efficient channel access control mechanisms to enable multiple network nodes to communicate over a shared medium (e.g., in an Ethernet network)
- Metric** A function designed to measure the individual properties of either certain elements or the entire system and its services
- Millimeter-wave communications** Communications over extremely high-frequency radio communication channels in the electromagnetic spectrum from 30 to 300 GHz (ITU definition)
- Minimal node degree** A structural metric defined as the minimal value of degrees of nodes in the networked system
- Minimum cost flow problem** An optimization problem aimed at determining the cheapest solution of sending a certain amount of flow through the network
- Min-max** A scheme of determining the set of k end-to-end mutually disjoint paths for a given demand with the objective of minimizing the cost of the most expensive path

- Min-min** A scheme of determining the set of k end-to-end mutually disjoint paths for a given demand with the objective of minimizing the cost of the cheapest path (commonly the cost of the working path)
- Min-sum** A scheme of determining the set of k end-to-end mutually disjoint paths for a given demand with the objective of minimizing the sum of costs of these paths
- Mixed-Integer Linear Programming (MILP)** A paradigm of solving optimization problems, in which the objective function and constraints are linear, some variables are integer, while the other ones can remain continuous
- Mixed-Integer Nonlinear Programming (MINLP)** A paradigm of solving optimization problems in which either the objective function or some of the constraints are nonlinear, some variables are integer, while the other ones can remain continuous
- Multicast routing** A one-to-many routing scheme suitable for group communications where a message needs to be sent to a group of destination nodes
- Multi-cost network** A scheme with differentiated costs assigned to network links in computations of multiple disjoint paths of the same demand
- Multi-domain routing** Routing of information over multiple network domains
- Multi-hop Inter-Vehicular Communications (MIVC)** Inter-vehicular communications utilizing multi-hop transmission scheme
- Multi-hop routing** Routing of information via multiple transit nodes
- Multilayer network** A general scheme for contemporary wide-area networks composed of multiple layers, each layer acting as a network of a particular type (e.g., WDM, SONET, and IP), allowing for the existence of the upper layer virtual links provided by the physical lower layer paths
- Multipath routing** A routing scheme enabling simultaneous transmission of information over multiple end-to-end (frequently disjoint) paths
- Multiple failure scenario** A scenario involving a failure of more than one network element at a time, following from either a simultaneous failure of many network elements or the occurrence of subsequent failures of network elements before completing a physical repair of formerly failed elements
- Multiple-input multiple-output (MIMO)** A technique to multiply the capacity of a radio link using multiple transmit and receive antennas to benefit from multipath propagation
- Multiple random failures** A scenario of multiple failures occurring simultaneously at random locations of a system
- Multiprotocol Label Switching (MPLS)** A forwarding mechanism that relays information between network nodes based on path labels rather than network addresses, which prevents time-consuming searches in a routing table
- Named Data Object (NDO)** The main abstraction in information-centric networking representing the addressable content
- Natural disaster** An event following from the activities of nature, such as earthquakes, floods, or fires
- Nesting** See “recursion”

- Network flow** A graph theory concept used in modeling the movements of certain entities from the respective source nodes to the related destination nodes following particular paths formed by sequences of transit nodes, applicable, e.g., in computer science, electrical engineering, management, operations research, or physics
- Network-level functional metric** A metric used to analyze the system quality of service at the network level
- Network–Network Interface (NNI)** An interface to signaling and management functions between neighboring networks enabling the interconnection of signaling, IP-MPLS, or ATM networks
- Network redundancy** The ratio of protection capacity to working capacity
- Network resilience** The ability of the network to provide and maintain an acceptable level of service in the face of various faults and challenges to normal operation
- Network Virtualization Environment (NVE)** A set of multiple heterogeneous network architectures (often from different service providers) that can be utilized to form a virtual network by the InP
- Networked system** A system composed of interconnected elements (such as servers, computing units, switches, routers, etc.) providing storage, computation, and communication services
- Node-encircling p -cycle** A protection scheme useful in scenarios of failures of a given node located inside the protection cycle
- Node-link formulation** Formulation of an optimization problem including variables referring to the utilization of a link connecting the source node s_r and leading to a destination node t_r by communication paths to serve given demands d_r
- Node load** A functional metric proposed to measure node importance in overlay networks. It provides information on the number of overlay links passing through a given physical node
- Nonlinear Programming** A paradigm of solving the optimization problems, in which either the objective function or some of the constraints are nonlinear, while variables are continuous
- Non-predictable disaster** A natural event that cannot be forecasted (e.g., earthquakes)
- Nonrepudiability** Assurance provided by a neutral third party that a given transaction/event did (or did not) occur
- Non-shareable spare capacity** Capacity already reserved at a link for backup path purposes that cannot be shared by the backup path of the considered demand
- Normalization (in relation to the recovery process)** Recognition of the repaired element and return to the normal operational state of a network
- NP-complete problem** A problem that belongs to the class of *NP* problems, as well as can be obtained by a polynomial reduction from another *NP*-complete problem
- NP problem** A problem for which it can be verified in polynomial time whether the answer “yes” to its recognition version is indeed “yes”

- Number of concurrent faults** Number/ratio of faults a selected recovery scheme can cover
- Number of spanning trees** A structural metric used to calculate the total number of distinct spanning trees (i.e., trees that include all nodes of the networked system) that exist for a given network graph
- OC-48** A network link with a transmission rate of up to 2488.32 Mbit/s
- On-Board Unit (OBU)** The appropriate in-vehicle wireless communications device enabling VANET communications
- Open Shortest Path First (OSPF)** A routing protocol belonging to the class of link-state routing algorithms widely used in IP networks to establish and maintain the communication paths. In this protocol, each network node is aware of the state (up/down) of each link in the network and the associated link cost metric and calculates the cheapest communication paths based on that metric using Dijkstra's algorithm
- Operational fault** A fault related to the exploitation phase of a system
- Opportunistic routing** See "anypath routing"
- Optical Cross Connect (OXC)** A network device designed to switch optical signals in a fiber optic network at high-speed rates
- Overlay networking** A multilayer network scheme assuming that routing is performed in each layer separately (i.e., no routing information is shared between the network layers)
- p*-Cycles** See "protection cycles"
- Packet delivery ratio (PDR)** The ratio of the number of delivered data packets to the destination node
- Packet error rate (PER)** The number of incorrectly received data packets (i.e., including at least one erroneous bit) divided by the total number of received packets
- Packet-level functional metric** A metric used to analyze the system quality of service at the packet level
- Packet loss ratio (PLR)** The ratio of the number of lost data packets transmitted by a given node
- Packet switching** A method of grouping data into packets of a limited length consisting of the packet header and the packet payload
- Partial failure** A failure referring to some parts of a network element, e.g., some ports of a switch
- Path-protecting *p*-cycle** A scheme involving a single protection cycle to protect the entire working path
- Path protection** A resilient routing scheme assuming utilization of a single backup path installed in advance (prior to a failure) to protect the entire working path of a demand
- Path restoration** A resilient routing scheme assuming utilization of a single backup path established after the occurrence of a failure to protect the entire working path of a demand
- Path symmetry** A functional metric used to measure the symmetry of paths between source and destination nodes. It focuses on analyzing the end-to-end

latency (expressed by the round trip time) and the hop count for the related forwarding and reverse paths

Peer model A multilayer network model allowing for the sharing of routing information between network layers

Peer-to-peer (P2P) networking A scheme of partitioning tasks or workloads among peers (equally privileged entities)

Percent of IP service unavailability (PIU) Percentage of total scheduled IP service time categorized as unavailable using the IP service availability function

Performability A discipline that is used to provide measures on the performance of a system compared with the respective Quality of Service requirements following from service specifications in terms of delay, jitter, bandwidth, and packet losses

Permanent fault A fault whose presence is not limited in time to certain internal/external conditions

Physical fault A fault that is a result of unfavorable physical phenomena

Physical layer (PHY) The lowest layer in the seven-layer network model, responsible for sending/receiving signals, and, therefore, comprising the respective hardware transmission technologies

Point of Interest (POI) A specific location point that may be found helpful/interesting (in VANET communications)

Predictable disaster A natural event that can be forecasted (e.g., hurricanes or floods)

Preferential attachment rule A principle of adding a new node to the network by linking it with existing nodes with probability proportional to the degree of existing nodes

Preplanned protection A resilient communication scheme based on backup paths installed in advance (when establishing the respective primary path)

Primary path The main transmission path of a demand

Problem reduction An algorithm for transforming one problem into another problem

Propagation time over a link A packet-level metric used to evaluate the time for a packet necessary to travel via the considered link

Protection cycles A scheme to protect a mesh network from a link failure based on ring structures characterized by ring-like high recovery speed and mesh-like high capacity efficiency

Protection path See “alternate path”

Protection switching time A time interval from the occurrence of a network fault until the completion of protection switching operations

Qualitative robustness A functional metric to evaluate the variation of QoS parameters for a broad range of occurrences of impairments (including random attacks, targeted attacks, dynamic epidemical failures, and dynamic periodical failures)

Quality of Experience (QoE) The degree of delight or annoyance of the user of an application or service

Quality of Resilience (QoR) A separate aspect of quality provisioning focusing on QoS metrics related to network resilience

- Quality of Service (QoS)** The overall performance of a communication network seen by the end users in terms of delay, jitter, bandwidth, and packet losses
- Quantitative robustness** A functional metric proposed to evaluate the efficiency in establishing connections in a given time step as the fraction of the number of established connections to the total number of connections that should have been established at that time step
- R-value** A functional metric defined as the weighted average of values of several other functional metrics of robustness
- Random failure** A failure of a network element (node/link) being independent of the element characteristics
- Reactive restoration** A methodology of redirecting the affected flows onto backup paths found reactively upon occurrence of a failure
- Recognition problem** A problem with “yes/no” answer
- Recovery switching** Redirection of the affected traffic onto the alternate path
- Recovery time** See “restoration time”
- Recovery token** A signal used in a multilayer recovery scheme allowing for synchronization of the recovery actions at consecutive layers
- Recovery ratio** A quotient of the actual recovery bandwidth divided by the traffic bandwidth that is intended to be protected
- Recursion** A parent–child relationship for virtual networks creating the VN hierarchy (i.e., VNs built on top of other VNs), often referred to as nesting
- Redundancy** Duplication of certain elements of a system (or its functions) to improve the overall system resilience
- Regional failure** A scenario of simultaneous failures of multiple network elements located close enough to the failure epicenter to suffer from the results of the event
- Relative delay penalty/stretch** A functional metric used to evaluate the efficiency of overlay networks. It is defined as the time needed for a packet to be transmitted end-to-end via the overlay path consisting of overlay links divided by the time needed when transmitting this packet between the same pair of end nodes, however, measured directly in the underlying transport network
- Relative size of the largest connected component** A structural metric defined as the ratio of the number of nodes of the largest connected cluster of the system and the total number of system nodes
- Reliability** A metric of service continuity referring to the probability that a system/service remains operable in a given time frame $(0, t)$
- Replica server** A node hosting the copy of the content in anycast communications
- Request for Comments (RFCs)** A publication of the Internet Engineering Task Force (IETF) and the Internet Society—the major standards setting and technical development Internet bodies
- Resilience** The ability of a network to provide and maintain an acceptable level of service in the face of various faults and challenges to normal operation
- Resilience differentiation** Distinction of differentiated Quality of Resilience features tailored to differentiated demands of end users
- Resilient routing** A routing scheme that can provide the continuity of service in the presence of disruptions

- Restoration efficiency** The success ratio of recovery defined as the number of connections that were restored divided by the total number of affected connections
- Restoration time** A time interval from the occurrence of a network fault to the instant of time when the affected traffic is either completely restored, or until spare resources are exhausted, or no more extra traffic exists
- Retainability** Probability that a service will continue to be provided
- Retransmission rate** A packet-level metric used to evaluate the ratio of retransmitted packets over the total number of transmitted packets in a given observation time window
- Revisitation** Characteristics of a virtualization scheme enabling hosting multiple virtual nodes from a given VN by a single physical node
- RFC 1058** IETF specification “Routing Information Protocol”
- RFC 1142** IETF specification “OSI IS-IS Intra-domain routing protocol”
- RFC 1195** IETF specification “Use of OSI IS-IS for Routing in TCP/IP and Dual Environments”
- RFC 2178** IETF specification “OSPF version 2”
- RFC 2328** IETF specification “OSPF version 2”
- RFC 2330** IETF specification “Framework for IP performance metrics”
- RFC 3031** IETF specification “Multiprotocol label switching architecture”
- RFC 3386** IETF specification “Network hierarchy and multilayer survivability”
- RFC 3469** IETF specification “Framework for Multi-Protocol Label Switching (MPLS)-based Recovery”
- RFC 3561** IETF specification “Ad hoc on-demand distance vector (AODV) routing”
- RFC 3945** IETF specification “Generalized Multi-Protocol Label Switching (GMPLS) Architecture”
- RFC 4090** IETF specification “Fast reroute extensions to RSVP-TE for LSP tunnels”
- RFC 4378** IETF specification “A Framework for Multi-Protocol Label Switching (MPLS) Operations and Management (OAM)”
- RFC 4379** IETF specification “Detecting Multi-Protocol Label Switched (MPLS) data plane failures”
- RFC 4427** IETF specification “Recovery (protection and restoration) terminology for Generalized Multi-Protocol Label Switching (GMPLS)”
- RFC 4428** IETF specification “Analysis of Generalized Multi-Protocol Label Switching (GMPLS)-based recovery mechanisms (including protection and restoration)”
- RFC 5286** IETF specification “Basic specification for IP fast reroute: Loop-free alternates”
- RFC 6981** IETF specification “A framework for IP and MPLS fast reroute using Not-Via addresses”
- Road-Side Unit (RSU)** A roadside communications infrastructure deployed to enable vehicle-to-infrastructure communications in VANETs
- Robustness** Indicator of the performance of a network under perturbative conditions

- Root bridge** The root node of a spanning tree
- Round trip time (RTT)** The round trip delay for unicast probes between neighboring nodes
- Route Request (RREQ)** A message sent by a source node toward the destination node in AODV routing protocol to initiate the establishment of a communication path
- Route Response (RREP)** A message sent back by a destination node toward the source node in AODV routing protocol to confirm the establishment of a communication path
- Routing Information Protocol (RIP)** A routing protocol belonging to the class of distance-vector class of algorithms that uses the hop count metric to determine the end-to-end paths characterized by the lowest cost expressed by the number of links traversed by these paths
- Safety** A measure of a system's dependability under catastrophic failures, in particular, referring to the effect rather than the cause of a failure
- Scale-free network** A network characterized by the power law distribution of node degrees
- Scope of a recovery procedure** The size of the primary path segment protected by a single backup path
- Security** The ability of a system to protect itself from various unauthorized activities
- Segment protection** A scheme assuming utilization of a backup path installed in advance (prior to a failure) to redirect the affected traffic over a given segment of a primary path
- Segment recovery** See "segment protection"
- Segment restoration** A scheme assuming utilization of a backup path established after the occurrence of a failure to redirect the affected traffic over a given segment of a primary path
- Service channel (SCH)** A communication channel in VANETs used to transmit the application data
- Service continuity** The length of a period during which the service is not interrupted
- Service interruption time** The length of a period the service is interrupted
- Service Level Agreement (SLA)** A service contract in use between the service provider and the customer
- Service Loss Block (SLB)** An event occurring for a block of packets at an ingress node when the ratio of lost packets at an egress node exceeds some threshold
- Service provider (SP)** An entity providing clients with communications, storage, and/or processing services
- Service recovery** Actions a service provider performs as a response to the service failure
- Setup vulnerability** The amount of time that a working path is left unprotected during such tasks as recovery path computation and recovery path setup
- Shareable spare capacity** Capacity already reserved at a link for backup path purposes that can be shared by the backup path of the considered demand

- Shared protection** A scheme and conditions of backup path installation allowing for sharing the link capacities among multiple backup paths
- Shared Risk Link Group (SRLG)** A set of network elements, being either links, nodes, physical devices, or a mix of these, subject to a common risk of failure
- Shortest path problem (SPP)** A problem aimed at determining a path between a given pair of vertices in a given network graph characterized by the minimal total distance defined as the summary cost of all links traversed by that path
- Shortest path tree (SPT)** A problem aimed at determining a tree in a given network graph sourced at a given vertex and including all the other vertices of that graph characterized by the minimal total cost of all links belonging to that tree
- Signal-to-Noise Ratio (SNR)** A measure used to compare the level of a signal against the level of a background noise
- Single-cost network** A scheme with the same link cost assigned to a given link in computations of all paths for each demand
- Single failure scenario** A scenario involving a failure of a single network element at a time
- Single-hop Inter-Vehicular Communications (SIVC)** Inter-vehicle communications strategy using one-hop message dissemination
- Software-Defined Networking (SDN)** An approach to communication networks allowing for management of network services by abstraction of lower level functionality
- Software fault** A flaw in the design or development of software
- Software redundancy** A technique of using additional instructions, segments of a program, or even additional programs to take over the role of the main software in scenarios of software-related failures
- Spanning tree** A subgraph of a network graph being a tree that includes all the vertices of a network graph
- Spanning Tree Protocol (STP)** A network protocol that maintains a loop-free logical topology in Ethernet networks in a way to prevent loops in packet forwarding
- Spare capacity** Capacity reserved at network links for backup path purposes
- Sparse V2I system** A VANET system designed to provide vehicle-to-land communication services at hot-spots (e.g., parking availability, parking payment, or collection of tolls for roads/bridges/tunnels)
- Standard deviation of opinion scores (SOSs)** The standard deviation of the mean opinion score (MOS)
- Static hardware redundancy** Redundancy at the hardware level applied in a way that the effects of faults are not expected to appear in the outputs of such modules as long as the failures will not affect the replicated components at the same time
- Store-carry-forward transmission** A transmission scheme assuming that information is sent to an intermediate node where it is stored for some time (e.g., due to lack of connectivity) and next sent to another intermediate node to approach the destination node

- Structural metric** A metric referring to the topological properties of the entire system
- Subjective metric** A metric used to assess user satisfaction with the service (often called quality of experience—QoE)
- Survivability** Capability of a system to fulfill its mission in a timely manner in the presence of threats, including attacks or natural disasters
- Synchronous Digital Hierarchy (SDH)** A common technology for transmission of synchronous data over optical links being the world-wide equivalent of SONET (from the USA)
- Synchronous Optical Network (SONET)** North American equivalent of Synchronous Digital Hierarchy (SDH) network architecture
- Technology-related failure/disaster** An event triggered by technological events such as power blackouts or faults incorporated into the structure of system elements at various phases of their lifetime
- Temporary fault** A fault occurring during a certain period of time which can be terminated/cleared without any interrupting operation
- Throughput** A measure of a successful message delivery rate for the analyzed communication channel
- Time Division Multiplexing (TDM)** A method of transmitting and receiving independent signals over a common communication path using a synchronized time-dependent exclusive access to the medium
- Time redundancy** A technique of performing additional operations meant to repeat or acknowledge a correct execution of former operations
- Timing failure** A scenario when a service is not delivered in time—either too early, too late, or not delivered at all
- Top-down recovery** A recovery scheme in a multilayer network where recovery actions concerning the affected flows are initiated in the uppermost layer and are then continued at the lower layers
- Traffic grooming** Consolidation of lower rate flows into larger units using TDM scheme
- Traffic tolerance** The ability of a network to tolerate additional (unusual) and often unpredictable traffic load (e.g., as a result of excessive activities of end users)
- Transient fault** A fault lasting relatively shortly (e.g., less than a minute) and tending to subside when the factor affecting the network element ceases to exist
- Transmission Control Protocol (TCP)** A connection-based, reliable, streaming communication protocol (being part of the widely used TCP/IP protocols family) used to send data between processes
- Trap problem** A scenario when the algorithm fails to establish the next disjoint path of a demand, even though it would be feasible for a given topology
- Trustworthiness** A resilience category defined in terms of measurable service delivery characteristics as the assurance that the communication system will perform as expected
- Ubiquitous V2I system** A VANET communication system offering vehicle-to-land-based communication services to end users not restricted to selected locations

- Unicast routing** A one-to-one routing transmission scheme
- Unidirectional Path-Switched Ring (UPSR)** A ring network in which two copies of information are sent in either direction around a ring
- User–Network Interface (UNI)** An interface between a user and a network provider defining responsibilities of the service provider and of the user
- Value failure** A scenario when the service value deviates from the specification
- Vehicle Safety Communications Consortium (VSCC)** A consortium consisting of BMW, Daimler Chrysler, Ford, GM, Nissan, Toyota, and VW to contribute to standards/specifications focusing on vehicular safety issues
- Vehicle-to-infrastructure (V2I)** A VANET communication scheme between vehicles and a roadside infrastructure
- Vehicle-to-vehicle (V2V) communications** Short-range wireless communications between vehicles in VANETs without the support of a roadside infrastructure
- Vehicular Ad hoc NETWORK (VANET)** An ad hoc self-organized network using vehicles as mobile nodes
- Vertex connectivity** A structural metric defined as the smallest number of vertices of graph G , the removal of which causes disconnection of system elements (i.e., partitioning of the system architecture into separated zones)
- Virtual link** A logical link in the overlay structure created over a physical communication infrastructure as an end-to-end (commonly multi-hop) physical path
- Virtual Local Area Network (VLAN)** A local area virtual network
- Virtual Network (VN)** A network created based on resources of a physical network, including virtual links and communication nodes (that can also be virtual) having its broadcast domain separated from other coexisting virtual networks
- Virtual node** Functionality of a communication node hosted on one/several physical nodes
- Virtual Private Network (VPN)** An extension of a private network across the public network (e.g., Internet) enabling communication devices to exchange data across a shared or a public network as if they were in a direct scope in a private network
- Virtualization** Creation of a virtual instance of a communication network
- Voice over IP (VoIP)** A methodology of delivery of voice communications and multimedia sessions over IP networks (e.g., Internet)
- Vulnerable road user (VRU)** A pedestrian in a VANET communications scheme
- Wavelength Division Multiplexing (WDM)** A communications technology enabling frequency division multiplexing of multiple optical carrier signals onto a single optical fiber with multiple wavelengths of laser light, providing bidirectional communications per each wavelength over a fiber link
- Weapon of mass destruction (WMD)** A nuclear, radiological, or other type of weapon able to cause significant damage to human-made structures (e.g., buildings, communication networks) resulting in multiple failures bounded in certain regions of occurrence
- Weighted adjacency matrix** A square matrix used to determine the node–node neighborhood relation with values of its elements storing additional information, e.g., on the nominal capacity of direct links between the related nodes

- Weighted incidence matrix** A matrix providing information on the neighborhood relation of network nodes and links, typically by storing information on the nominal capacity of links associated with certain nodes
- Wi-Fi** Specification of a local area wireless communication network allowing for communications of devices via 2.4 GHz and 5 GHz radio bands
- Wireless Mesh Network (WMN)** A wireless network organized in a mesh topology, consisting of mesh clients and mesh routers interconnected by wireless links (frequently of high speed—as, e.g., in the case of links between mesh routers)
- Wireless Sensor Network (WSN)** A set of autonomous sensors interconnected via wireless links set up to monitor physical/environmental conditions, e.g., pressure, temperature, or sound, etc., and to forward such information cooperatively to the main location in the network
- Wireless transceiver** A networking device capable of sending and receiving information via a wireless communication channel
- Working capacity** Capacity reserved at network links for working paths purposes
- Working path** See “primary path”
- Y.1540** ITU-T recommendation “Internet protocol data communication service – IP packet transfer and availability performance parameters”
- Y.1541** ITU-T recommendation “Network performance objectives for IP-based services”
- Y.1542** ITU-T recommendation “Framework for achieving end-to-end IP performance objectives”
- Y.1561** ITU-T recommendation “Performance and availability parameters for MPLS networks”
- Y.1562** ITU-T recommendation “Framework for higher layer protocol performance parameters and their measurement”

Index

A

Acceptable level of service, 8
Access station, 113
Accidental fault, 28
Active path first, 105, 194
Add/drop multiplexer, 94
Ad hoc on-demand distance vector, 281
Adjacency matrix, 61
Aging-related bug, 28
Alternate path, 9, 91, 177
Alternate port, 128
Anycasting, 213
Anypath forwarding, 294
Attack, 226
Auditability, 43
Augmented model, 112
Authenticity, 43
Authorisability, 43
Automatic protection switching, 99
Availability, 37, 39, 45
Average content accessibility, 80
Average node degree, 69
Average shortest path length, 71
Average two-terminal reliability, 75

B

Backup LSP sharing, 136
Backup path, 9, 91, 177
Backup path sharing, 156, 159
Backup port, 128
Basic feasible solution, 168
Basic variable, 169
Basic vector, 168

Basis, 168

Benign failure, 32
Betweenness centrality, 64, 227
Bhandari's algorithm, 188
Bidirectional line switched ring, 93
Bidirectional network link, 61
Big Data, 202
Binary linear programming, 144
Bohrbug, 28
Bottom-up recovery, 112
Branch-and-bound, 171
Bridge protocol data unit, 127

C

Canonical matrix, 168
Capacity-constrained network, 148
Cascading failure, 7
Catastrophic failure, 32
Central node, 77, 227
Challenge, 8, 23
Challenge tolerance, 33
Chromatic number, 162
Clean-slate, 203
Closeness centrality, 65
Cloud computing/communications, 202
Clustering coefficient, 76
Coexistence, 205
Column generation, 173
Column generation subproblem, 173
Common and internal spanning tree, 130
Common pool, 113
Common spanning tree, 130
Complete failures, 6

Confidentiality, 43
 Conflicting backup paths, 159
 Consistent failure, 33
 Content-centric networking, 213
 Content-oriented networking, 202, 212
 Control channel, 274
 Cooperative awareness message, 299
 Cost of resilience, 14
 Critical latency, 277
 Current vertex, 181

D

Data dissemination, 276
 Data-oriented networking, 212
 Dedicated protection scheme, 103
 Dedicated short range communications, 274
 Degree centrality, 65
 Delay-tolerant networking, 25
 Demand node, 145
 Demand volume, 143
 Dense wavelength division multiplexing, 93, 111
 Dependability, 37
 Dependent failure, 25
 Development fault, 28
 D-geodiversity, 102
 Diameter, 71
 Dijkstra's algorithm, 180
 Directional network link, 61
 Disruption tolerance, 33, 36
 Distributed denial of service, 7, 25
 Distributed fault, 29
 Diversity, 35
 Downstream neighbor LFA, 132
 D^2R^2+DR strategy, 12
 Dual-cost network, 194
 Dynamic hardware redundancy, 35

E

ECMP alternate, 132
 Economical challenge, 25
 Edge connectivity, 73
 Efficiency, 72
 Eigenvector centrality, 66
 Electromagnetic pulse attack, 7, 245
 Element-related metric, 60
 End-to-end delay, 81
 Environmental challenge, 25
 Error, 32
 Expected transmission count, 85
 Exposure to disruptions, 32

External fault, 29
 Extreme point, 166

F

Facility backup scheme, 119, 136
 Failure, 32
 Failure rate, 38
 Fast reroute, 119, 136
 Fault, 8, 28
 Fault avoidance, 35
 Fault detection, 10
 Fault localization, 10
 Fault notification, 11
 Fault removal, 35
 Fault tolerance, 33, 34
 Feasible point, 166
 Fixed addressing, 279
 Flow p -cycle, 108
 Free-space optical, 25
 Free capacity, 104
 Functional metric, 60
 Future Internet, 202

G

Generalized multiprotocol label switching, 112
 Geocasting, 282
 Geographical addressing, 280
 Geographical diversity, 177
 Global protection, 102
 Graph diversity, 73
 Graph of conflicts, 161
 Greedy forwarding, 281

H

Hamiltonian p -cycle, 108
 Hardware fault, 28
 Hardware redundancy, 35
 Heisenbug, 28
 Heterogeneity, 70
 Hold-off timer, 113
 Human-computer interaction, 202
 Human error, 23
 Human-made fault, 28
 Hybrid model, 112
 Hypertext transfer protocol, 42

I

Incidence matrix, 62
 Inconsistent failure, 33

Individual viewpoint, 13
 Information-centric networking, 212
 Information society, 205
 Infotainment, 273
 Infrastructure provider, 204
 Inheritance, 205
 Integer linear programming, 144
 Integrated model, 112
 Integrated recovery, 112
 Integrity, 43
 Intentional fault, 28
 Interdependent systems, 25
 Inter-domain recovery, 109
 Intermediate system-to-intermediate system, 132, 180
 Intermittent fault, 29
 Internal fault, 29
 Internet of Things, 203
 Internet service provider, 204
 Inter-vehicular communications, 273
 IP fast-reroute, 131

J

Jitter, 81

K

k-Penalty algorithm, 193

L

Label distribution protocol, 135
 Label edge router, 134
 Label switched path, 135
 Label switch router, 134
 Laplacian matrix, 63
 Large-scale disaster, 23
 Large-scale testbed, 206
 Largest first, 162
 Latency, 81
 Lightpath, 113
 Linear programming, 144
 Line of sight, 240
 Link betweenness centrality, 65
 Link/node protection, 103
 Link/node restoration, 103
 Link-path formulation, 150
 Link-protecting LFA, 132
 Link stress, 79
 Link utilization, 78
 Local fault, 29
 Local protection, 102
 Local-to-egress protection, 136

M

Maintainability, 43
 Malicious attack, 6, 24
 Mandelbug, 28
 Master problem, 173
 Mean content accessibility, 80
 Mean down time, 38
 Mean opinion score, 83
 Mean time between failures, 38
 Mean time between maintenance, 38
 Mean time to (first) failure, 38
 Mean time to recovery, 38
 Mean time to repair, 38
 Mean up time, 38
 Mesh, 239
 Metric, 59
 Minimal node degree, 70
 Minimum cost flow, 144
 Min-max, 178
 Min-min, 178
 Min-sum, 178
 Mixed-integer linear programming, 144
 Mobile ad-hoc network, 276
 Modular capacity, 147
 Multi-commodity flow, 145
 Multi-cost network, 179, 194
 Multi-domain routing, 109
 Multiple failures, 100
 Multiple failure scenario, 5
 Multiple-input multiple-output, 239
 Multiple random failures, 101
 Multiple spanning tree protocol, 129
 Multiprotocol label switching, 134

N

Named data object, 212
 Natural disaster, 6, 23
 Nesting, 206
 Network, 3
 Networked system, 3
 Network flows, 143
 Network-level functional metric, 60
 Network-level metric, 78
 Network-network interface, 112
 Network redundancy, 95
 Network resilience, 7, 33
 Network virtualization, 204
 Network virtualization environment, 205
 Node-encircling *p*-cycle, 108
 Node-link formulation, 150
 Node load, 78
 Node-protecting LFA, 132
 Non-basic variable, 169

Non-linear programming, 144
 Non-predictable disaster, 7, 24
 Nonrepudiability, 43
 Non-shareable spare capacity, 104
 Non-transient failure, 5
 Non-visited vertex, 181
 Normalization, 11
 Not-via, 133
 Number of spanning trees, 74

O

On-board unit, 275
 1:1 protection, 103
 One-to-one backup scheme, 119, 136
 Open shortest path first, 84, 132, 180
 Operational fault, 28
 Opportunistic routing, 294
 Optical cross connect, 111, 113
 Optical transport network, 93
 Optimal vertex, 166
 Organizational viewpoint, 13
 Overlay model, 112

P

Packet header, 125
 Packet-level metric, 60, 81
 Packet loss ratio, 81
 Packet payload, 125
 Packet switching, 125
 Partial failures, 6
 Path diversity, 74
 Path-protecting p -cycle, 107
 Path protection, 102, 103
 Path restoration, 103
 Path symmetry, 78
 P-cycle, 106, 163
 Peer model, 112
 Penetration rate, 274
 Performability, 43
 Perimeter forwarding, 281
 Permanent fault, 29
 Physical fault, 28
 Point of local repair, 136
 Polyhedron, 166
 Polytope, 166
 Predecessor index, 181
 Predictable disaster, 7, 23
 Preferential attachment, 77, 226
 Preplanned protection, 99
 Pricing problem, 173
 Primary path, 9, 91, 177
 Propagation time over a link, 81

Protection-switching time, 114
 Protection cycle, 106
 Protection path, 91
 Public safety, 273

Q

Qualitative robustness, 80
 Quality of experience, 82
 Quality of resilience, 43, 96
 Quality of service, 3, 81
 Quantitative robustness, 80

R

Random failure scenario, 5
 Rank, 168
 Rapid spanning tree protocol, 128
 Reactive restoration, 99
 Recovery switching, 11
 Recovery time, 92
 Recovery token, 113
 Recursion, 206
 Reduction of environmental pollution, 273
 Redundancy, 9, 34
 Regional disjointness, 177
 Regional failures, 5, 101
 Relative delay penalty/stretch, 79
 Relative size of the largest connected component, 75
 Reliability, 37, 41, 52
 Reliability block diagram, 45
 Rerouting/restoration scheme, 119
 Resilience differentiation, 97
 Resilience disciplines, 33
 Resilient routing, 91
 Restoration efficiency, 117
 Restricted master problem, 174
 Retransmission rate, 82
 Revisitation, 206
 Ring network, 93
 Road-side unit, 275
 Robustness, 33
 Root bridge, 127
 Round trip time, 85
 Route request, 281
 Route response, 281
 Routing information protocol, 84
 R-value, 81

S

Safety, 43
 Scale-free network, 77

Security, 43
 Segment protection, 102, 103
 Segment restoration, 103
 Self-healing rings, 94
 Self-protecting multipath, 137
 Service availability, 288
 Service channel, 274
 Service continuity, 37, 41
 Service downtime, 37
 Service failure, 32
 Service level agreement, 3, 97
 Service provider, 204
 Shareable spare capacity, 104
 Shared protection, 103
 Shared risk link group, 99
 Shortest-path tree, 180
 Shortest path problem, 180
 Single-commodity flow, 145
 Single-cost network, 179, 193
 Single failure, 5, 100
 Single-hop message dissemination, 277
 Slack variable, 167
 Socio-political challenge, 25
 Software fault, 28
 Software redundancy, 35
 Spanning tree, 126
 Spanning tree protocol, 126
 Spare capacity, 104
 SRLG-disjoint, 99
 Stability index, 297
 Standard deviation of opinion scores, 83
 Static hardware redundancy, 35
 Structural metric, 60
 Subjective metric, 60, 82
 Supply node, 145
 Survivability, 33, 34
 Suurballe's algorithm, 184
 Synchronous digital hierarchy, 93
 Synchronous optical network, 93

T

Technology-related disaster, 6
 Technology-related failure, 23
 Technology viewpoint, 13
 Temporary fault, 29
 Tentative distance, 181
 Throughput, 82
 Time division multiplexing, 112

Time redundancy, 35
 Timing failure, 32
 Top-down recovery, 112
 Traffic coordination, 273
 Traffic grooming, 114
 Traffic tolerance, 33, 36
 Transient failure, 5
 Transient fault, 29
 Transit node, 145
 Transmission control protocol, 93
 Transshipment node, 145
 Trap problem, 194
 Trojan horse, 31
 Trustworthiness, 33, 37

U

Unavailability, 40
 Unidirectional path switched ring, 93
 Unusual traffic, 25
 User network interface, 112

V

Value failure, 32
 Vehicle-to-infrastructure, 275
 Vehicle-to-vehicle, 275
 Vehicular ad-hoc network, 6, 25, 273
 Vertex-coloring, 161
 Vertex connectivity, 72
 Vertex splitting, 191
 Virtualization, 204
 Virtual link, 205
 Virtual local area network, 129, 204
 Virtual network, 204
 Virtual node, 205
 Virtual private network, 204
 Visited vertex, 181
 Vulnerable road user, 276

W

Weapon of mass destruction, 7
 Weighted adjacency matrix, 62
 Weighted incidence matrix, 63
 Wireless mesh network, 6, 239
 Working capacity, 104
 Working path, 9, 91, 177