Title

Content Connectivity for Next-Generation Networks

Author

Le, Giap Dang

Publication Date

2022

Content Connectivity for Next-Generation Networks

By

Giap Dang Le
B.Eng. in Electrical Engineering, Danang University of Technology
M.Sc. in Communications and Signal Processing, Ilmenau University of Technology

Dissertation

Submitted in partial satisfaction of the requirements for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

in the

Office of Graduate Studies

of the

University of California

Davis

Approved:

Distinguished Prof. Dr. Biswanath Mukherjee, Chair

Prof. Dr. Massimo Tornatore, Co-chair

Prof. Dr. Xin Liu

Committee in Charge

2022

To my family

Contents

List of Figures vi

List of Tables viii

Abstract ix

Acknowledgments xi

List of Abbreviations 1

1 Introduction 3

1.1 Background 3

1.2 Organization and Contributions 5

2 Survivable Virtual Network Mapping with Content Connectivity Against Multiple Link Failures in Optical Metro Networks 7

2.1 Introduction 8

2.2 Related Works 12

2.3 Survivable Virtual Network Mapping With Content Connectivity Against Multiple Link Failures 12

2.4 Mathematical Formulation 19

2.5 Illustrative Numerical Results 22

2.5.1 Physical Networks and Risk Models 22

2.5.2 An Example of How SVNM with CC Achieves Higher Survivability 25

2.5.3 Cost Comparison: SVNM with CC vs. SVNM with NC 26

2.5.4 Availability Comparison: SVNM with CC vs. SVNM with NC . . 30

2.6 Conclusion 34

3 Reliable Provisioning with Degraded Service Using Multipath Routing from Multiple Data Centers in Optical Metro Networks 36

3.1 Introduction 37

3.2 Related Works 41

3.3 Reliable Provisioning with Degraded Service Using MPMD 42
3.3.1 Problem Statement 42
3.3.2 Mathematical Formulation 44
3.4 Heuristics For Reliable Provisioning with Degraded Service Using MPMD 48
3.4.1 Auxiliary Graph 48
3.4.2 Equal-Bandwidth, Maximum-K MPMD (K-MPMD) 49
3.4.3 Flexible MPMD (F-MPMD) 53
3.4.4 Complexity Analysis 56
3.4.5 Service Probability 56
3.5 Illustrative Numerical Results 58
3.5.1 Physical Network and Simulation Setup 58
3.5.2 MPMD-ILP vs. K-MPMD vs. F-MPMD 60
3.5.3 F-MPMD vs. F-MPSD vs. DPP 64
3.5.4 F-MPMD 67
3.6 Conclusion 69

4 Reliable Provisioning of Low-latency and High-bandwidth Extended Reality Live Streams 70

4.1 Introduction 71
4.2 Reliable Provisioning of Low-latency and High-bandwidth Extended Reality Live Streams 76
4.2.1 Problem Statement 76
4.2.2 Model of Latency Budget and Trade-off Between Latency and Bandwidth 78
4.2.3 Backup from Different DCs With Multicast and Flexible Bandwidth 79
4.3 Illustrative Numerical Results 84
4.3.1 Physical Networks and Simulation Setup 84
4.3.2 Simulation Results 87
4.4 Conclusion 89

5 Conclusion **90**

5.1 Summary 90

5.2 Future Works 90

List of Figures

2.1 Survivable virtual network mappings against a random single-link failure. 10

2.2 Finding network cutsets and content cutsets on a 4-node virtual network. 15

2.3 An example of virtual network mapping without and with content connectivity against double-link failures. 16

2.4 Milan52 network with 52 nodes and 101 bidirectional links, where the number attached to each link is normalized risk factor, ϵ_{ij}, $0 \leq \epsilon_{ij} \leq 1.0$, $\forall ij \in L_P$. 22

2.5 Tokyo23 network with 23 nodes and 43 bidirectional links, where the number attached to each link is normalized risk factor, ϵ_{ij}, $0 \leq \epsilon_{ij} \leq 1.0$, $\forall ij \in L_P$. 23

2.6 Virtual network mapping where content connectivity is survivable against a large-scale disaster in Tokyo23. 25

2.7 Virtual networks on the Milan52 network. 27

2.8 Virtual networks on the Tokyo23 network. 28

2.9 Cost of virtual networks in Fig. 2.7 mapped over Milan52. 28

2.10 Cost of virtual networks in Fig. 2.8 mapped over Tokyo23. 29

2.11 A family of virtual networks with the same virtual nodes and different virtual links for content availability computation. 31

2.12 Content availability against double-link failures as a function of number of content replicas for the VN in Fig. 2.11. 33

3.1 Different reliable service-provisioning schemes. 39

3.2 An auxiliary graph with a dummy node and dummy links. 48

3.3 Simulation framework. 59

3.4 Acceptance ratio of MPMD-ILP, K-MPMD, and F-MPMD as a function of request arrival rates. 61

3.5 Acceptance ratio of F-MPMD, F-MPSD, and DPP as a function of request arrival rates. 64

3.6 Acceptance ratio of F-MPMD as a function of number of content replicas (NR) at different request arrival rates. 66

3.7 Acceptance ratio of F-MPMD as a function of request arrival rates at three different levels of degraded service. 68

4.1 Service-provisioning schemes of extended reality live streams. 72

4.2 Simulation framework. 86

4.3 Backup from different data centers with multicast and flexible bandwidth vs. reference schemes. 88

List of Tables

2.1 Milan52 and Tokyo23 network characteristics. 24

2.2 C_k cost saving compared to N_k cost in percent when two content replicas are placed at optimal locations. 30

2.3 Number of k-link combinations on the Milan52 network disconnects CC and NC of the VN in Fig. 2.11. 32

3.1 MPMD-ILP vs. K-MPMD vs. F-MPMD 62

3.2 Average execution time of the algorithms. 63

3.3 F-MPMD vs. F-MPSD vs. DPP 65

3.4 Average Service Probability in Different Failure Scenarios. 66

Abstract

Content Connectivity for Next-Generation Networks

Network connectivity, i.e., the reachability of any network node from all other nodes, is often considered as the default network survivability metric against failures. However, in case of a large-scale disaster disconnecting multiple network components, network connectivity may not be achievable. On the other hand, with the shifting service paradigm towards cloud in today's networks, most services can still be provided as long as at least a content replica is available in all disconnected network partitions. As a result, the concept of content connectivity has been introduced as a new network survivability metric under a large-scale disaster. Content connectivity is defined as the reachability of content from every node in a network under a specific failure scenario. In this dissertation, we investigate how to ensure content connectivity in various applications under different scenarios of failures in the physical network.

In the first contribution of the dissertation, we consider the survivable virtual network mapping with content connectivity against multiple link failures. We derive necessary and sufficient conditions, and develop a novel mathematical formulation to map a virtual network over a physical network such that content connectivity for the virtual network is ensured against multiple link failures in the physical network. In our numerical results, obtained under various network settings, we compare the performance of mapping with content connectivity and network connectivity, and show that mapping with content connectivity can guarantee higher survivability, lower network bandwidth utilization, and significant improvement of service availability.

In the second contribution of the dissertation, we investigate the problem of reliable provisioning with degraded service using multipath routing from multiple data centers. We consider the scenario where contents are cached in multiple locations in a network. Such a wide content replication offers a unique opportunity to provide better services to users, especially for content-based services, e.g., video delivery. We propose a reliable service-provisioning scheme that selects the optimal subset of data centers hosting the desired content and inversely multiplexes a content request over multiple link-disjoint paths. We formulate an integer linear program and develop heuristics for the problem, and use them to solve various complex and realistic network instances. Numerical data show that, compared to conventional service-provisioning schemes such as multipath routing from a single data center or dedicated-path protection, our proposed scheme efficiently utilizes network resources, improves reliability, and reduces latency.

In the third contribution of the dissertation, we addresses the reliable provisioning of low-latency and high-bandwidth extended reality live streams in next-generation networks. We investigate the backup from different data centers with multicast and flexible offered bandwidth to fulfill extended reality live stream requests while guaranteeing strict requirements on reliability, latency, and bandwidth. We consider the scenario where contents are not cacheable (e.g., live streams) and propose a service-provisioning scheme to protect not only against failures of links in the network but also against failures of computing and storage in data centers. We develop scalable algorithms for the backup from different data centers with multicast and flexible offered bandwidth and use them to solve various complex network instances in a dynamic network environment. Numerical data show that, compared to a conventional service-provisioning scheme such as backup from the same data center, our proposed service-provisioning scheme provides higher reliability, reduces latency, and efficiently utilizes network resources.

Acknowledgments

My deepest gratitude goes to my supervisor, Distinguished Prof. Dr. Biswanath Mukjerhee, for his invaluable guidance through my PhD. He has been a great inspiration to me, not only in academia but also in all aspects of life. I received constant support from day one when he accepted me to the lab. I am always eternally grateful to him for the opportunity to work under world-class researchers and in prestigious projects.

I sincerely thank my co-supervisor, Prof. Dr. Massimo Tornatore, for his constant support and valuable comments. Many important aspects of my research were motivated from his observations and ideas during our meetings.

I thank my friends and colleagues in the Computer Networks Lab at UC Davis for their companionship. I especially thank Dr. Sifat Ferdousi and Dr. Andrea Marotta for giving me the initial push during the first year of my PhD.

I would like to thank Prof. Dr. Zhi Ding, Prof. Dr. Dipak Ghosal, and Prof. Dr. Xin Liu for being part of my PhD qualifying exam and dissertation committee.

I thank the National Science Foundation and Japan-US Network Opportunity for funding our works.

A special thank also goes to Dr. Pulak Chowdhury, Dr. Partha Bhaumik, and the colleagues at Ennetix Inc. for their supervision during my internship as a full-time software engineer from July 2021 to September 2022.

Finally, I am grateful to my family for their unconditional love and support.

List of Abbreviations

5G Fifth-Generation Broadband Cellular Networks.

6G Sixth-Generation Broadband Cellular Networks.

AR Augmented Reality.

BDD-MF Backup from Different Data Centers with Multicast and Flexible Bandwidth.

BSD Backup from the Same Data Center.

CC Content Connectivity.

CDN Content Delivery Network.

CO Central Office.

CU Centralized Unit.

DC Data Center.

DDC Differential Delay Constraint.

DPP Dedicated-Path Protection.

DU Distributed Unit.

EON Elastic Optical Network.

ER Extended Reality.

HSMR Hybrid Single/Multipath Routing.

ILP Integer Linear Program.

KC-VON *k*-Link Content-Connected Virtual Optical Network.

LCAS Link Capacity Adjustment Scheme.

MBRLLC Mobile Broadband Reliable Low-Latency Communications.

MCF Multi-Core Fiber.

MPMD Multipath Routing from Multiple Data Centers.

MPSD Multipath Routing from a Single Data Center.

NC Network Connectivity.

OPC Online Path Computation.

PN Physical Network.

SDM Space-Division Multiplexing.

SVNE Survivable Virtual Network Embedding.

SVNM Survivable Virtual Network Mapping.

URLLC Ultra Reliable Low-Latency Communications.

VCAT Virtual Concatenation.

VN Virtual Network.

VR Virtual Reality.

WDM Wavelength-Division Multiplexing.

XURLLC Next-Generation Ultra-Reliable and Low-Latency Communications.

Chapter 1

Introduction

1.1 Background

With the adoption of advanced technologies in connectivity, storage, and computing such as mobile communications (e.g., 5G/6G), cloud, and edge computing, traditional protection and restoration mechanisms must evolve to provide better services to users and to cope with emerging challenges such as large-scale failure scenarios (such as those in a large-scale disaster event). An important new trend in modern networking comes from the opportunity to replicate contents/services to increase network reliability.

Today, networks are becoming more content centric. Current projections show that 72% of the Internet traffic will cross content delivery networks (CDNs) by 2022 [1]. The deployment of CDNs, especially with edge data centers (DCs), is critical in next-generation networks to meet high throughput, low latency, and stringent reliability requirements. Popular contents should be cached in DCs closer to end users for traffic load and latency reduction. Therefore, most content-based services can be provided as long as a content replica is available in each disconnected network partition, even under a large-scale disaster.

In such content-centric networks, traditional approaches to network survivability should be updated to reflect the evolving reliability requirement. In this dissertation, we consider the evolution of the traditional survivability metric, namely network connectivity, and investigate how it can be evolved towards a new concept, called content connectivity.

Network connectivity (NC) is defined as the reachability of any network node from all other nodes in a network. Originally, NC has been used to measure network survivability in end-to-end communications and will probably remain the default option for a smaller failure scenario (such as a random single-link/node failure). Unfortunately, in case of a large-scale disaster, multiple links and nodes may be simultaneously interrupted, and ensuring NC can be very costly, or even infeasible. To ensure service continuity even in such extreme failure scenarios, content connectivity can be considered as a new approach to measure network survivability [2].

Content connectivity (CC) is defined as the reachability of content from every node in a network under a certain failure scenario. The main idea is that, even if the network becomes disconnected, as long as CC is guaranteed, every user can still reach at least a content replica in all disconnected network partitions. Therefore, service continuity is guaranteed [3, 4]. In this dissertation, we investigate how to ensure CC in various applications under different scenarios of failures in the physical network.

In Chapter 2, we focus our attention on how to find survivable virtual network mapping (SVNM) with CC. We investigate how to evolve the formulation of SVNM from NC to CC in case of multiple link failures. We identify necessary and sufficient conditions to map a virtual network (VN) over a physical network (PN) such that CC for the VN is guaranteed against multiple link (i.e., k-link) failures in the PN. Then, we use these conditions to formulate the problem of finding SVNM with CC against k-link failures as an integer linear program (ILP). We simulate various network settings and compare the performance of SVNM with CC to NC. Our numerical results show that SVNM with CC has higher survivability, particularly against a large-scale disaster, saves network bandwidth, and significantly improves service availability.

In Chapter 3, we propose multipath routing from multiple data centers (MPMD) as a reliable service-provisioning scheme to fulfill content requests (e.g., video-on-demand, virtual reality (VR), or augmented reality (AR) requests) in a dynamic network environment while guaranteeing strict requirements on bandwidth, and improving reliability and latency. This proposed service-provisioning scheme enjoys the benefits of multipath routing, provides protection against network (e.g., link) and content source (e.g., DC) failures, and uses minimal additional network resources due to the nature of multipath routing. We formulate the MPMD problem as an ILP, develop two scalable heuristics, and use them to solve various complex network instances. Numerical data show that, compared to conventional service-provisioning schemes such as dedicated-path protection (DPP) and multipath routing from a single data center (MPSD), MPMD efficiently utilizes network resources, provides higher reliability, and reduces latency; hence, it is highly suitable for the emerging content services.

In Chapter 4, we investigate backup from different data center with multicast and flexible offered bandwidth (BDD-MF) as a service-provisioning scheme to fulfill extended reality (ER) live stream requests while guaranteeing strict requirements on reliability, latency, and bandwidth. Our proposed service-provisioning scheme provides protection not only against failures of links in the network but also against failures of computing and storage in DCs. Moreover, for the first time and to the best of our knowledge, we consider the multicast functionality of optical nodes and the trade-off between latency and bandwidth of ER live streams to reduce unnecessary network capacity. We develop scalable algorithms for BDD-MF and use them to solve various complex network instances in a dynamic network environment. Numerical data show that, compared to a conventional service-provisioning scheme such as backup from the same data center (BSD), our proposed service-provisioning scheme provides higher reliability, reduces latency, and efficiently utilizes network resources; hence, it is highly suitable for ER live streams.

1.2 Organization and Contributions

This dissertation is organized as follows. In Chapter 2, we investigate survivable virtual network mapping with content connectivity against multiple link failures in optical metro networks. This work has been presented at IEEE International Conference on Advanced Networks and Telecom Systems [3] and published in IEEE/Optica Journal of Optical Communications and Networking [4].

In Chapter 3, we investigate reliable provisioning with degraded service using multipath routing from multiple data centers in optical metro networks. This work has been presented at Optical Fiber Communication Conference [5] and submitted to IEEE Transactions on Network and Service Management (second round of revisions completed) [6].

In Chapter 4, we investigate reliable provisioning of low-latency and high-bandwidth extended reality live streams. We will submit this work to a reputed journal.

In Chapter 5, we conclude this dissertation and discuss important future research directions.

Chapter 2

Survivable Virtual Network Mapping with Content Connectivity Against Multiple Link Failures in Optical Metro Networks

Network connectivity, i.e., the reachability of any network node from all other nodes, is often considered as the default network survivability metric against failures. However, in case of a large-scale disaster disconnecting multiple network components, network connectivity may not be achievable. On the other hand, with the shifting service paradigm towards cloud in today's networks, most services can still be provided as long as at least a content replica is available in all disconnected network partitions. As a result, the concept of content connectivity has been introduced as a new network survivability metric under a large-scale disaster. Content connectivity is defined as the reachability of content from every node in a network under a specific failure scenario. In this chapter, we investigate how to ensure content connectivity in optical metro networks. We derive necessary and sufficient conditions, and develop a novel mathematical formulation to map a virtual network over a physical network such that content connectivity for the virtual network is ensured against multiple link failures in the physical network. In our numerical results, obtained under various network settings, we compare the performance of mapping with content connectivity and network connectivity, and show that mapping with content connectivity can guarantee higher survivability, lower network bandwidth utilization, and significant improvement of service availability.

2.1 Introduction

Optical metro networks are gaining importance as a key segment of the telecom network infrastructure. They provide the physical substrate to enable novel network services that will shape our future society such as smart city services (e.g., smart transportation, smart energy, and smart health care) and incoming 5G services (e.g., ultra reliable low-latency communications (URLLC)) [7, 8]. Several of these services will require an optical metro network, which allows them to fulfill extremely-high availability requirements. For instance, URLLC services in 5G might require higher than five nines (e.g., 99.999%) availability [8, 9].

To fulfill these requirements, optical metro networks must accommodate different services over different logical partitions of their resources. These logical partitions are referred to as network slices, virtual functions, or VNs. Network softwarization in future networks will allow us to create multiple VNs, each tailored for a specific use case, on top of a common PN; and network slicing will be an important technology to effectively support diverse performance (e.g., availability) requirements of different services [10].

Traditional protection and restoration mechanisms used to ensure high network reliability must evolve to cope with these emerging challenges, and address large-scale failure scenarios (such as those in a large-scale disaster event). An important new trend in modern telecom networks comes from the opportunity to replicate contents/services to increase network reliability.

Content can be web objects (text, graphics, and scripts), downloadable objects (media files, software, documents), applications (e-commerce, portals), live streaming/on-demand media, caches, or social media sites which are requested by users (i.e., virtual nodes in a VN). Note that some contents are static so replicas are stateless and synchronization in networks is not required. Other services are dynamic, stateful, and require frequent synchronization. While our proposed approach works without any problem with static services, it might suffer a synchronization problem in case of dynamic services. Still, in case of large-scale failures, slightly old content is better than no content at all.

Today, networks are becoming more content centric. Current projections show that 72% of the Internet traffic will cross CDNs by 2022 [1]. The deployment of CDNs, especially with edge DCs, is critical in next-generation networks to meet high throughput, low latency, and stringent reliability requirements. Popular contents should be cached in DCs closer to end users for traffic load and latency reduction. Therefore, most content-based services can be provided as long as a content replica is available in each disconnected network partition, even under a large-scale disaster.

In such content-centric networks, traditional approaches to network survivability should be updated to reflect the evolving reliability requirement. In this chapter, we consider the evolution of the traditional survivability metric, namely NC, and investigate how it can be evolved towards a new concept, called content connectivity (CC).

NC is defined as the reachability of any network node from all other nodes in a network. Originally, NC has been used to measure network survivability in end-to-end communications and will probably remain the default option for a smaller failure scenario (such as a random single-link/node failure). Unfortunately, in case of a large-scale disaster, multiple links and nodes may be simultaneously interrupted, and ensuring NC can be very costly, or even infeasible. To ensure service continuity even in such extreme failure scenarios, CC can be considered as a new approach to measure network survivability [2].

CC is defined as the reachability of content from every node in a network under a certain failure scenario. The main idea is that, even if the network becomes disconnected, as long as CC is guaranteed, every user can still reach at least a content replica in all disconnected network partitions. Therefore, service continuity is guaranteed.

In the context of future optical metro networks, CC and NC can be used to measure the survivability of VNs/slices. In an optical network, a VN comprises a set of virtual nodes connecting to each other using lightpaths (a.k.a. virtual links) [11]. The set of virtual nodes can be central offices (COs) requesting a leased VN to connect them. One important question is how to assign resources from a PN (e.g., an optical metro network)

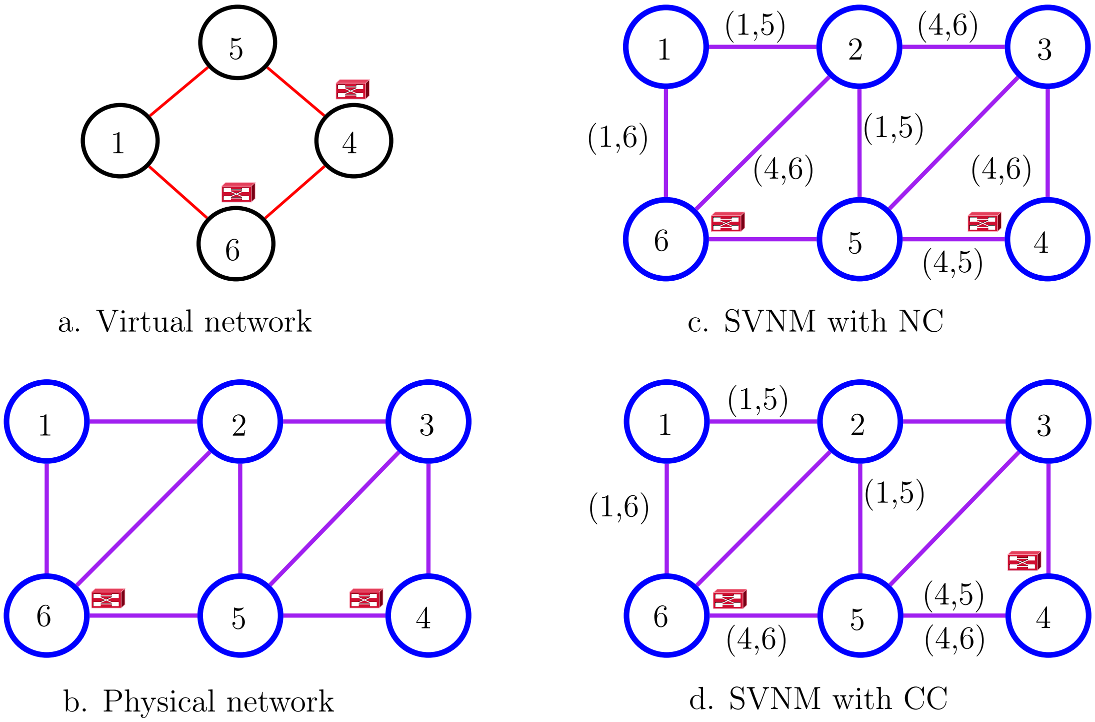


Figure 2.1: Survivable virtual network mappings against a random single-link failure.

to a given VN such that the VN is survivable against a certain failure scenario. This problem is called SVNM. In a SVNM problem, the physical location of virtual nodes (no mapping) is predefined and virtual links must be allocated (mapped). In case we are required to assign resources to both virtual nodes and virtual links, we refer to it as the problem of survivable virtual network embedding (SVNE) [12–14].

In Fig. 2.1, we consider the SVNM example of a 4-node, 2-link connected VN (Fig. 2.1.a) over a 6-node PN (Fig. 2.1.b). In the PN, nodes 4 and 6 host two content replicas of interest for the VN. In Fig. 2.1.c, the VN is mapped over the PN such that no physical link supports more than one virtual link. Thus, the VN is network-connected survivable (NC-survivable) against a random single-link failure in the PN. That is, there is no single link in the PN whose removal disconnects the VN. In Fig. 2.1.d, the VN is mapped over the PN in a way that the physical link (4, 5) carries both virtual links (4, 5) and (4, 6), and its failure would disrupt NC for the VN. Nonetheless, content replicas at nodes 4 and 6 remain accessible to every virtual node; hence, the VN is content-connected survivable (CC-survivable). Note that: 1) the mapping in Fig. 2.1.d utilizes two physical links less than the mapping in Fig. 2.1.c (both virtual links and physical links are bidirectional, i.e., VN and PN are directed graphs); and 2) the mapping in Fig. 2.1.d is survivable in the higher layer (e.g., IP layer, i.e., node 5 can use node 1 as a transit node to reach the content replica at node 6) rather than in the optical layer. Note that compared to providing protection in the optical layer, providing protection in the higher layer is cost-effective in terms of network resource utilization, and may be practical in case of a large-scale disaster [15]. It is worth mentioning that, although a disaster can disrupt both nodes and links in a network, we focus our attention on link failures. Based on data collected from network operators, link failures are ten times more likely to happen than node failures in a day [16]. Moreover, each node in a network normally deploys backup hardware for primary units. Also, while a random single-link failure is still the dominant failure scenario in optical networks, multiple links can be simultaneously affected in case of a large-scale disaster. For example, on July 2, 2019, damage to multiple concurrent fiber bundles serving network paths in the US-EAST-1 disrupted Google cloud networks for 24 hours [17]. More recently, on December 19, 2019, a coincident cut of multiple fiber-optic cables disconnected the Internet and disrupted network access for two hours in parts of eastern Europe, Iran, and Turkey [18].

In this study, we focus our attention on how to find SVNM with CC. We investigate how to evolve the formulation of SVNM from NC to CC in case of multiple link failures. We identify necessary and sufficient conditions to map a VN over a PN such that CC for the VN is guaranteed against multiple link (i.e., k-link) failures in the PN. Then, we use these conditions to formulate the problem of finding SVNM with CC against k-link failures as an ILP. We simulate various network settings and compare the performance of SVNM with CC to NC. Our numerical results show that SVNM with CC has higher survivability, particularly against a large-scale disaster, saves network bandwidth, and significantly improves service availability.

The rest of this work is organized as follows. In Section 2.2, we review related works. In Section 2.3, we state the necessary and sufficient conditions to map a VN over a PN with CC. In Section 2.4, we formulate the problem of SVNM with CC against k-link failures as an ILP. In Section 2.5, we perform numerical validation by comparing the performance of SVNM with CC and NC. We conclude this work in Section 2.6.

2.2 Related Works

Some research has been conducted on SVNM with CC. The authors in [2] developed an algorithm to map a VN over a PN with CC and NC against single-link failures. In [19], the authors extended the problem of SVNM with CC to provide protection against double-link failures. Compared to both these works, in this study, we provide a theoretical insight into the problem of SVNM with CC, and generalize it against an arbitrary number of failures. Some other works have leveraged the concept of CC, but applied it directly in the optical layer. In [20], the authors proposed an approach to provide shared protection with CC using spectrum sharing in elastic optical DC networks. In [21], the authors developed algorithms to place contents and establish k link-disjoint lightpaths from each node in the network to content replicas. In [22], the authors proposed to group together users requesting the same level of CC reliability (e.g., k-link connected to content replicas) to form a k-link content-connected virtual optical network (KC-VON), and embedded it over an optical DC network. Compared to providing k independent end-to-content lightpaths for each individual user, the KC-VON approach has higher spectrum efficiency and acceptance rate.

In the following section, we elaborate on the problem of SVNM with CC against k-link failures. We derive necessary and sufficient conditions, and develop a novel approach to map a VN over a PN such that CC is guaranteed for every virtual node against k-link failures.

2.3 Survivable Virtual Network Mapping With Content Connectivity Against Multiple Link Failures

In this section, we formulate the problem of mapping a VN over a PN with CC against k-link failures. We consider a directed graph $G_P(N_P, L_P)$ to represent a PN, where N_P and L_P are the set of physical nodes and the set of physical links (e.g., optical fibers).

We assume the physical nodes are equipped with wavelength converters, and leave the extension to the case of wavelength continuity for future work. $G_V(N_V, L_V)$ is a directed graph denoting a VN, where N_V and L_V are the set of virtual nodes and the set of virtual links. Physical links and virtual links are bidirectional, and a single failure disrupts connections in both directions. We assume that content replicas are available at a set of virtual nodes R, $R \subseteq N_V$. This work does not consider the cost of content replica storage and synchronization. Readers interested in content management can refer to [23].

Our objective is to map the VN over the PN such that every virtual node can reach at least one content replica after failures on k distinct physical links. Note that: 1) since we do not provide protection in the optical layer, direct lightpaths from a virtual node to content replicas are not required (i.e., a virtual node can use other virtual nodes as transit nodes to reach content replicas); and 2) a virtual node hosting a content replica is CC-survivable against failures on physical links. In this context, we assume that: a) the virtual node is attached to a local/colocated DC using local (short-reach) links that do not belong to the main network topology; and b) the DC is hosting the desired content. Since we do not consider local link failures, the virtual node is CC-survivable against failures on physical links (i.e., physical links in the main network topology). We use a matrix $\boldsymbol{B}$ to present the VN traffic request with each integer element β^{st}, $\forall st \in L_V$, denoting the virtual link st bandwidth. If a virtual link st is mapped on multiple, H, physical links in the PN, the virtual link occupies $H \times \beta^{st}$ bandwidth units. We aim at minimizing network resource utilization of SVNM measured in the total number of bandwidth units. In case β^{st} is measured in number of wavelengths, the mapping operation minimizes the total number of wavelength channels. Henceforth, we also define C_k and N_k as the mapping of $G_V(N_V, L_V)$ over $G_P(N_P, L_P)$ such that CC and NC for the VN are ensured after k-link failures. S_k denotes the mapping of $G_V(N_V, L_V)$ over $G_P(N_P, L_P)$ in which each virtual link takes the path with the minimal number of physical links (i.e., the shortest path in number of hops). We now define the necessary conditions for C_k existence.

Theorem 1. *Given $G_P(N_P, L_P)$, $G_V(N_V, L_V)$, and R, to find the mapping of G_V over G_P that guarantees C_k, the following conditions must be satisfied:*

- *each virtual node $s \in N_V \text{ - } R$ has a nodal degree $\delta(s) \geq k+1$, and*
- *each physical node $i \in N_P : s \rightarrow i,\ s \in N_V \text{ - } R$ has a nodal degree $\delta(i) \geq k+1$.*

These conditions imply that, if C_k exists, then every virtual node not hosting a content replica must have a nodal degree of at least $k+1$. Furthermore, the virtual node not hosting a content replica must be embedded over a physical node with a nodal degree of also at least $k+1$. Here, $s \rightarrow i$ implies that the physical node i hosts the virtual node s.

Proof. Assuming C_k exists and there is a virtual node not hosting a content replica with a nodal degree less than $k+1$ or there is a virtual node not hosting a content replica embedded onto a physical node with a nodal degree less than $k+1$, different failures on k physical links disconnect the virtual node from content. Hence, the conditions in Theorem 1 must be satisfied. □

In graph theory, a *cut* is the partition of a graph $G_V(N_V, L_V)$ into two disconnected parts, and separates N_V into two disjoint sets of nodes $N_V \text{ - } X$ and X. Each *cut* defines a *cutset*, $\theta(N_V \text{ - } X, X)$, which is the set of links with one endpoint in $N_V \text{ - } X$ and the other in X. By definition of a cutset, it can be derived that, after the removal of all links in a cutset, the graph becomes disconnected. This latter property is a corollary of Menger's theorem [24].

Since cuts, network cutsets, content cutsets, and Menger's theorem are fundamental for our problem formulation, we elaborate on these concepts in a small example. We define $\Theta = \{\theta_n(N_V \text{ - } X_n, X_n) : n = 1..N\}$ as the set of all N possible cutsets in a VN, and refer to Θ as the set of network cutsets. As an example in Fig. 2.2.a, we consider a 4-node VN and enumerate all possible network cutsets from θ_1 to θ_6 (i.e., $N = 6$).

We define a *content cutset* of a VN as a cutset where the removal of all virtual links in the cutset disconnects the VN with one network partition without content replicas. We use $\Gamma = \{\gamma_c(N_V \text{ - } X_c, X_c) : X_c \cap R = \varnothing,\ c = 1..C\}$ to denote the set of all C possible content cutsets in a VN. In Fig. 2.2.b, we consider a 4-node VN with two content replicas available at nodes 2 and 4, and enumerate all possible content cutsets from γ_1 to γ_2 (i.e., $C = 2$). In this example, γ_1 consists of the virtual links (1, 2) and (1, 4), and the removal

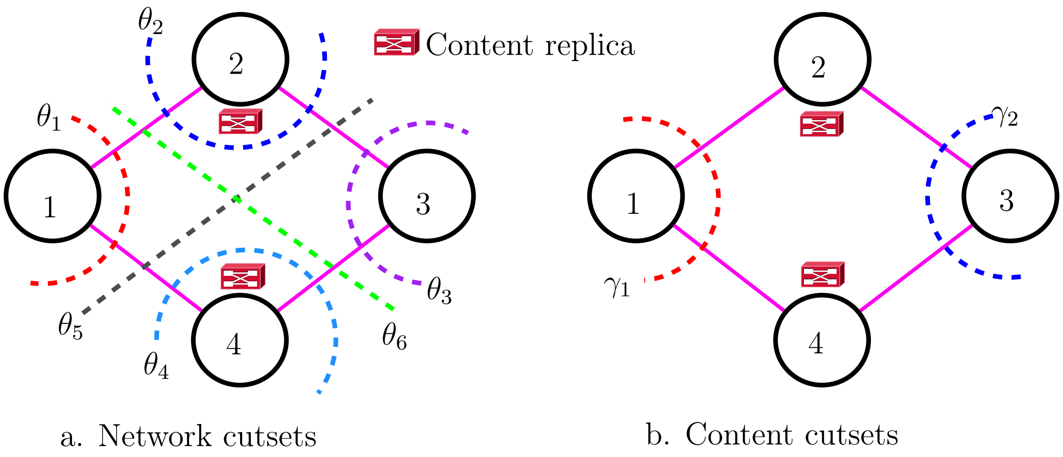


Figure 2.2: Finding network cutsets and content cutsets on a 4-node virtual network.

of (1, 2) and (1, 4) disconnects node 1 from content replicas. It is trivial to verify that $\Gamma \subseteq \Theta$, where tightness holds if there is only one content replica in the VN.

We also define $\Omega^k = \{\omega_z^k : \omega_z^k \subset L_P,\ |\omega_z^k| = k,\ z = 1..Z^k\}$ as the set of all k-link combinations in a PN. Here, Z^k is the number of k-link combinations, $Z^k = C(|L_P|, k)$ (i.e., combination without repetition). Let $q_{z,st}^k$ be a binary variable and $q_{z,st}^k = 1$ if the removal of k physical links in ω_z^k disconnects virtual link st, and zero otherwise. The following theorem gives a necessary and sufficient condition for the mapping of a VN over a PN to be C_k.

Theorem 2. *Given $G_P(N_P, L_P)$, $G_V(N_V, L_V)$, Γ, and Ω^k, the mapping of G_V over G_P is C_k if and only if:*

$$\sum_{st \in \gamma_c} q_{z,st}^k \leq |\gamma_c| - 1,\ \forall \omega_z^k \in \Omega^k,\ \forall \gamma_c \in \Gamma. \tag{2.1}$$

One can observe that $q_{z,st}^k = 1$ implies the virtual link st is mapped over one or more physical links in ω_z^k. The summation in Expression (2.1) computes the number of virtual links in the content cutset γ_c being mapped in ω_z^k. For the content cutset γ_c to be survivable against the removal of all k physical link in ω_z^k, this summation must be less than the cardinality of the content cutset, $|\gamma_c|$. The mapping of G_V over G_P is C_k if and only if Expression (2.1) is true for every content cutset in the VN and every

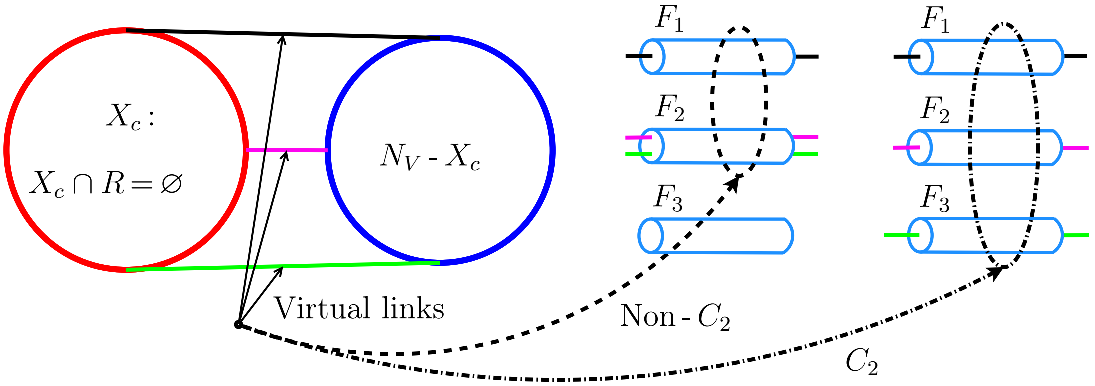


Figure 2.3: An example of virtual network mapping without and with content connectivity against double-link failures.

combination of k physical links in the PN. Traditionally, to find a survivable mapping (i.e., NC-survivable) against k link failures, the optimization model searches for every network cutset and every combination of k links. Therefore, the number of constraints in the ILP model increases exponentially with the number of virtual nodes as the number of network cutsets is proportional to $2^{|N_V|-1}$. In this study, the number of content cutsets is proportional to $2^{|N_V|\text{-}|R|}$. Even though the number of constraints in (2.1) still increases exponentially with the number of virtual nodes, our problem formulation significantly reduces the number of constraints compared to the conventional NC formulation. In the worst case (i.e., $|R| = 2$ because, if $|R| = 1$, the CC formulation reduces to the NC formulation), the number of constraints in Theorem 2 is reduced by 50%.

Proof. To prove this theorem, we must show that the above condition is both necessary and sufficient. The condition is necessary because, if there exists a set of k physical links carrying all virtual links in a content cutset, failures of these k links disconnect the VN in such a way that one network partition has no content replicas. Consequently, CC is not guaranteed. The condition is also sufficient because the removal of k random physical links leaves at least one virtual link in each content cutset survivable. Therefore, the VN must remain content-connected. □

This theorem is fundamental for the development of the optmization formulation in the following section. To clarify the logic of Theorem 2, we consider the illustrative example in Fig. 2.3. Here, two circles on the left side of the figure represent two parts of a VN connecting to each other via a cutset of three virtual links. The left circle denotes a network partition including X_c virtual nodes and no content replicas (i.e., $X_c \cap R = \varnothing$). By definition, this cutset is a content cutset because, if we remove all three virtual links in the cutset, the VN is disconnected with one partition without content replicas. For example, if all virtual links in the content cutset are mapped on two fibers (F_1 and F_2), C_2 (i.e., $C_k,\ k = 2$) is not ensured since failures on F_1 and F_2 disconnect X_c nodes from content replicas. However, if we map the virtual links in the content cutset over three fibers, the VN remains connected after failures on two arbitrary fibers out of three fibers; hence, C_2 is guaranteed. Note that, in this specific case (i.e., the mapping of the content cutset on three fibers in Fig. 2.3), both Non - C_2 and C_2 mappings for the content cutset in Fig. 2.3 use three bandwidth units (i.e., Non - C_2 uses one bandwidth unit on F_1 and two bandwidth units on F_2 while C_2 uses one bandwidth unit on each fiber). On a different note, if there is a content replica in X_c (i.e., $X_c \cap R \neq \varnothing$ and N_V - $X_c \cap R \neq \varnothing$), the cutset is not a content cutset. In fact, it is a network cutset; and if we remove all virtual links in the cutset, NC for the VN is disrupted but CC for the VN remains survivable.

The following theorem provides the upper bound of C_k cost and a sufficient condition for the existence of C_k with smaller cost than its respective N_k.

Theorem 3. *Given $G_P(N_P, L_P)$, $G_V(N_V, L_V)$, and R, then C_k exists if N_k exists.*

Proof. This theorem implies that N_k existence is the sufficient condition for C_k existence. The condition is always true because the set of network cutsets is the superset of the set of content cutsets. □

If we measure the cost to map a VN over a PN in the total number of bandwidth units, S_k (i.e., each virtual link takes the path with the minimal number of network hops) provides the lower bound of the mapping cost. Theorem 3 also implies that N_k cost is the upper bound of C_k cost. These observations conclude that, if S_k, C_k, and N_k exist, their costs relate to each other by $S_k \leq C_k \leq N_k$. Consequently, if N_k cost is equal to S_k cost, C_k cost is equal to N_k cost. In case there is only one content replica in the VN, $\Gamma = \Theta$ and C_k cost is equal to N_k cost, if both exist.

Now, let us introduce another theorem that allows us to identify under which condition C_k cost is lower than N_k cost.

Theorem 4. *Given $G_P(N_P, L_P)$, $G_V(N_V, L_V)$, R: $|R| > 1$, N_k, S_k, and let $v_M(s,t) : \mathbb{Z}^2_+ \mapsto \mathbb{Z}_+$ be the function that associates a virtual source node s and a virtual destination node t to the cost of the lightpath from s to t using the mapping M, then C_k cost is less than N_k cost if and only if:*

$$\exists\, t \in R \ : \sum_{\forall s \in V_L \text{ - } t} v_{N_k}(s,t) > \sum_{\forall s \in V_L \text{ - } t} v_{S_k}(s,t). \tag{2.2}$$

This theorem states that, if there is a virtual node t such that the total cost of lightpaths from other nodes in the VN to t using the mapping S_k is less than the total cost of lightpaths from other nodes in the VN to t using the mapping N_k, replicating a content to t guarantees C_k cost to be less than N_k cost.

Proof. To prove this theorem, we must recall that a virtual node hosting a replica is content-connected independently of physical link failures. Also, since we measure the cost of lightpath from s to t in number of bandwidth units, the optimal lightpath (i.e., the lightpath with minimal number of bandwidth units) is the lightpath spanning minimal number of physical links. In case virtual node t hosts a replica, we can map the lightpaths from other virtual nodes in the VN to t using optimal lightpaths to save cost while still ensuring CC for t. Therefore, if there exists t such that the inequality holds, C_k cost is less than N_k cost. By contradiction, if C_k cost is less than N_k cost and there is no t such that the inequality holds, S_k cost is equal to N_k cost. In this case, the cost relation we showed above implies that $C_k = N_k$. Therefore, the contradiction must be not true, and if C_k cost is less than N_k cost, there exists t such that the inequality holds. □

Given a virtual node t in a VN, we use D^t_M to denote the total cost of the lightpaths from other nodes in the VN to t using the mapping M. Namely,

$$D^t_{S_k} = \sum_{s \in N_V \text{ - } t} v_{S_k}(s,t), \ D^t_{N_k} = \sum_{s \in N_V \text{ - } t} v_{N_k}(s,t). \tag{2.3}$$

We term the cost difference $D^t = D^t_{N_k} - D^t_{S_k},\ D^t \geq 0$ as the replication gain for the virtual node t. One can observe that placing a content replica at a virtual node with a higher replication gain saves more cost for the C_k mapping.

2.4 Mathematical Formulation

In this section, we formulate the problem of mapping a VN over a PN with CC against k-link failures as an optimization problem (i.e, C_k formulation). Since all variables are integer, and objective and constraints are linear, the optimization formulation is an ILP.

Inputs:

- $G_P(N_P, L_P)$: directed graph representing a physical network.
- $G_V(N_V, L_V)$: directed graph representing a virtual network.
- k: number of physical link failures in C_k.
- $\Omega^k = \{\omega^k_z : \omega^k_z \subset L_P,\ |\omega^k_z| = k,\ z = 1..Z^k\}$: set of all k-link combinations in the physical network, $Z^k = C(|L_P|, k)$.
- $\Gamma = \{\gamma_c(N_V \text{-} X_c, X_c) : X_c \cap R = \varnothing,\ c = 1..C\}$: set of all C possible content cutsets in the virtual network.
- R: set of content replicas, $R \subseteq N_V$.
- $\boldsymbol{B}$: virtual network traffic matrix with each integer element β^{st} [bandwidth units], $\forall st \in L_V$, being the virtual link st bandwidth.
- $\boldsymbol{E}$: matrix representing the physical network risk model with each element ϵ_{ij} being the physical link ij normalized risk factor, $0 \leq \epsilon_{ij} \leq 1.0,\ \forall ij \in L_P$.
- F_{ij}: number of fibers on physical link ij, $\forall ij \in L_P$.
- W: fiber capacity (in bandwidth units).

Variables:

- f_{ij}^{st} is an integer variable and $f_{ij}^{st} = \beta^{st}$ if the virtual link st is mapped on the physical link ij, and zero otherwise. We assume the traffic from s to t is unsplittable.
- u_{ij}^{st} is a binary variable and $u_{ij}^{st} = 1$ if the virtual link st is mapped on the physical link ij, and zero otherwise.
- $q_{z,st}^{k}$ is a binary variable and $q_{z,st}^{k} = 1$ if the virtual link st is disconnected due to the removal of k physical links in ω_z^k, and zero otherwise.

Objective function:

$$\min\left(\alpha \times \sum_{ij \in L_P,\, st \in L_V} f_{ij}^{st} + \sum_{ij \in L_P,\, st \in L_V} \epsilon_{ij} \times u_{ij}^{st}\right) \tag{2.4}$$

Subject to:

$$\sum_{st \in L_V} f_{ij}^{st} \le F_{ij} \times W, \;\; \forall ij \in L_P \tag{2.5}$$

$$u_{ij}^{st} \ge \frac{1}{\alpha} \times f_{ij}^{st}, \;\; \forall ij \in L_P, \;\; \forall st \in L_V \tag{2.6}$$

$$q_{z,st}^{k} \ge \frac{1}{\alpha} \times \sum_{ij \in \omega_z^k} u_{ij}^{st}, \; \forall \omega_z^k \in \Omega^k, \; \forall st \in L_V \tag{2.7}$$

$$\sum_{j:ji \in L_P} f_{ji}^{st} - \sum_{j:ij \in L_P} f_{ij}^{st} = \begin{cases} -\beta^{st} & \text{if } i = s \\ +\beta^{st} & \text{if } i = t, \quad {\scriptstyle \forall i \in N_P \atop \forall st \in L_V} \\ \;\;0 & \text{o/w} \end{cases} \tag{2.8}$$

$$\sum_{st \in \gamma_c} q_{z,st}^{k} \le |\gamma_c|\text{ - }1, \; \forall \omega_z^k \in \Omega^k, \; \forall \gamma_c \in \Gamma \tag{2.9}$$

In the objective function (2.4), the first summation minimizes network resource utilization of C_k measured in the total number of bandwidth units. Since α is a large integer number (e.g., $\alpha = 10,000$), the first term of the objective function is also the leading term. The second summation minimizes the total risk factor of all physical links on which the VN is mapped. When multiple lightpaths equally utilize network resources between a pair of virtual nodes, the path with the lowest risk is selected to minimize the objective function's second term. Constraint (2.5) ensures that the mapping of the VN over the PN does not exceed each physical link capacity. Constraint (2.6) computes a binarization of the integer variable f_{ij}^{st} and assigns it to u_{ij}^{st}. One can notice that $u_{ij}^{st} = 1$ implies that a failure on the physical link ij disrupts the virtual link st. Constraint (2.7) sets the binary variable $q_{z,st}^{k} = 1$ if the removal of k physical links in ω_z^k disrupts the virtual link st. In other words, if the virtual link st is mapped on one or more physical links in ω_z^k, it is disconnected due to the removal of all physical links in ω_z^k. Constraint (2.8) enforces flow conservation for every virtual link. For each virtual link, traffic originates at the source node and ends at the destination node. At a transit node, input traffic is equal to output traffic. Constraint (2.9) imposes C_k on the mapping of $G_V(N_V, L_V)$ over $G_P(N_P, L_P)$. That is, Constraint (2.9) ensures that failures on k random physical links do not disconnect all virtual links in every content cutset γ_c. Namely, at least one virtual link in each content cutset γ_c is survivable against k-link failures in the PN. Hence, C_k is guaranteed.

The problem of SVNM with NC against k-link failures, N_k, can be directly obtained from the C_k formulation if, instead of the set of content cutsets (Γ), the set of network cutsets (Θ) is used in Constraint (2.9). The set of network cutsets is the set of all cutsets in a VN without content replica consideration (Fig. 2.2). As mentioned in Theorem 2, since the number of content cusets is less than the number of network cutsets, the C_k problem formulation significantly reduces the time required for finding a survivable mapping. In case of N_1 (i.e., $N_k,\ k = 1$), this formulation reduces to the one in [12]. Compared to [2,12,19], our approach is more generic as it generalizes the formulation to an arbitrary number of link failures; hence, it is applicable in every disaster scenario. The problem

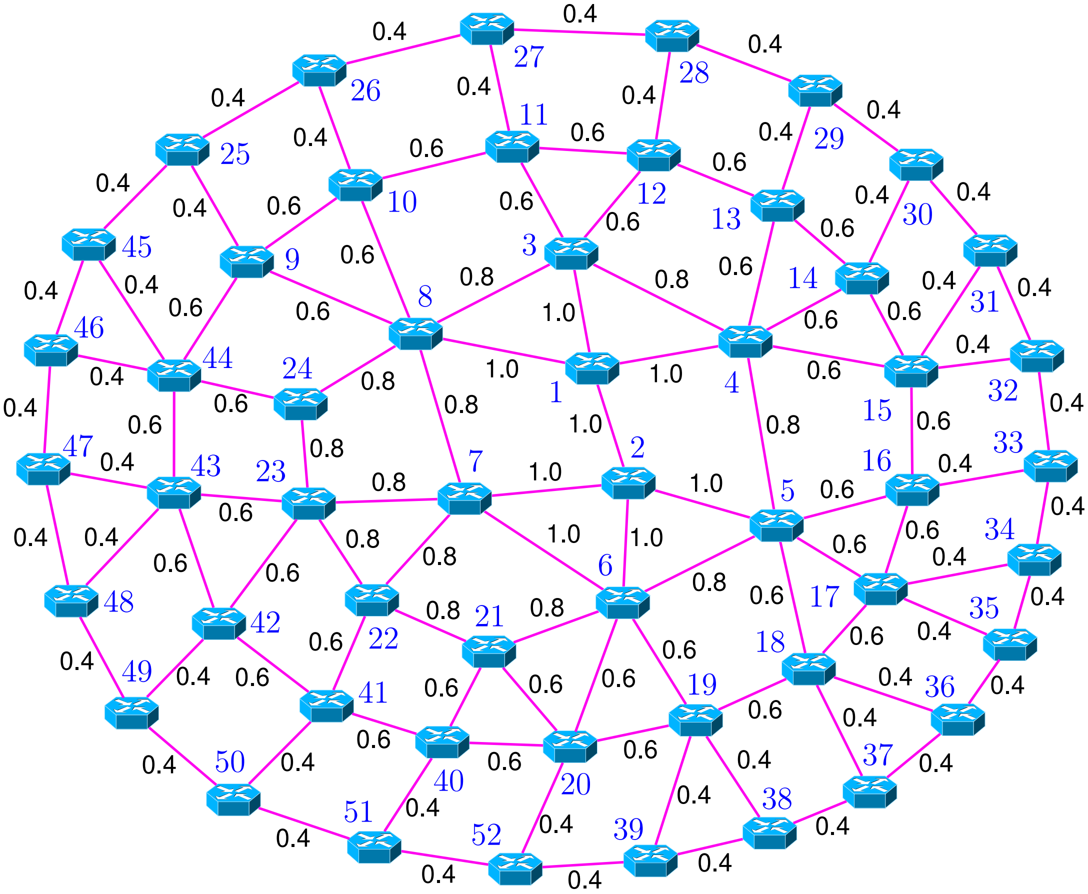


Figure 2.4: Milan52 network with 52 nodes and 101 bidirectional links, where the number attached to each link is normalized risk factor, ϵ_{ij}, $0 \leq \epsilon_{ij} \leq 1.0$, $\forall ij \in L_P$.

formulation is also extendable to provide network survivability against node failures.

2.5 Illustrative Numerical Results

2.5.1 Physical Networks and Risk Models

2.5.1.1 Physical Networks

We consider optical metro networks covering an urban or metro area up to tens of kilometers in diameter. Traditionally, optical metro networks consist of a main ring and multiple sub-rings connecting together. Currently, metro networks are evolving from a rigid ring-based infrastructure to a mesh-based network-and-computing ecosystem. Edge computing, i.e., a set of a small, highly-distributed DCs with processing and storage capabilities, is becoming popular in metro networks [25].

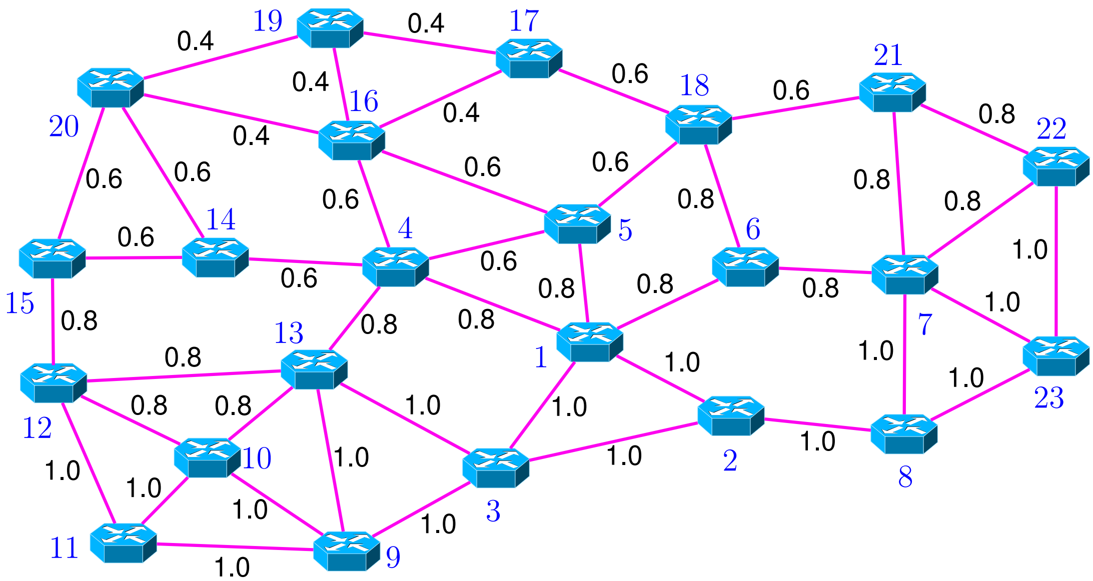


Figure 2.5: Tokyo23 network with 23 nodes and 43 bidirectional links, where the number attached to each link is normalized risk factor, ϵ_{ij}, $0 \leq \epsilon_{ij} \leq 1.0$, $\forall ij \in L_P$.

To perform our evaluation, we use the Milan52 and the Tokyo23 networks. Milan52 presents a Telecom Italia metro-regional reference network with 52 nodes and 101 bidirectional links (i.e., 202 unidirectional links) [26]. Compared to the original network, we modify Milan52 to make it planar (i.e., the network graph can be embedded in a plane). We also add links to increase the average node degree, and ensure a minimum node degree of three.

Tokyo23 network has been designed using regional characteristics such as population distribution, locations of local government offices, and railway lines with number of passengers getting on/off each station [27]. It consists of 43 bidirectional links (i.e., 86 unidirectional links) and 23 nodes with each node located at each ward office building in Tokyo metropolitan area. Characteristics of Milan52 and Tokyo23 networks are reported in Table 2.1.

2.5.1.2 Network Risk Models

Link failure probability depends on many parameters such as intensity of a disaster, distance from the disaster epicenter, intersection with the disaster zone, and frequency of construction works [28]. Since risk modelling is not the focus of this work, in our result setting, we use information publicly available from other fields such as climatology, geology, environmental science, and construction engineering to estimate network risk.

In Milan, severe natural disasters are unlikely and we assume the major cause of fiber cuts is dig-ups [11]. Electric, water, gas, and telephone companies perform excavation works more often in urban areas than in suburban areas. So, a link is more likely to experience a failure in urban areas. Thus, we assume the Milan52 network risk factor is higher in the city center and decreases radially. Each physical link is characterized by a normalized risk factor dependent on its location as shown in Fig. 2.4.

For the Tokyo23 network, we consider natural disasters such as earthquakes and tsunamis as the major cause of fiber cuts. Tokyo metropolitan area is located on three layers of plates – North American Plate, Philippine Sea Plate, and Pacific Plate which rub or collide with each other, and cause earthquakes to happen regularly [29]. A strong earthquake occurring in the sea (e.g., along Sagami Trough) may trigger tsunamis that ripple towards the Tokyo Bay [30]. Reasonably, we assume the links along the Tokyo Bay coast are at higher risk as shown in Fig. 2.5.

In the following subsections, we use the optimization formulation developed in Section 2.4 to solve various example networks. All simulations are run using IBM ILOG CPLEX Optimization Studio Version 12.10 on a workstation with an Intel Xeon(R) E3-1505M processor and 64.0 GB of RAM. We also pre-compute the set of k-link combinations of the PNs, and the set of content cutsets for each VN we use. For sake of simplicity, we assume every virtual link in a VN has unitary bandwidth in each direction. Note that we report the SVNM cost for each VN including bidirectional links. Running time is less

Characteristics	**Milan52**	**Tokyo23**
Number of nodes	52	23
Average node degree	3.8	3.7
Minimum node degree	3	3
Maximum node degree	6	5
Number of links	202	86

Table 2.1: Milan52 and Tokyo23 network characteristics.

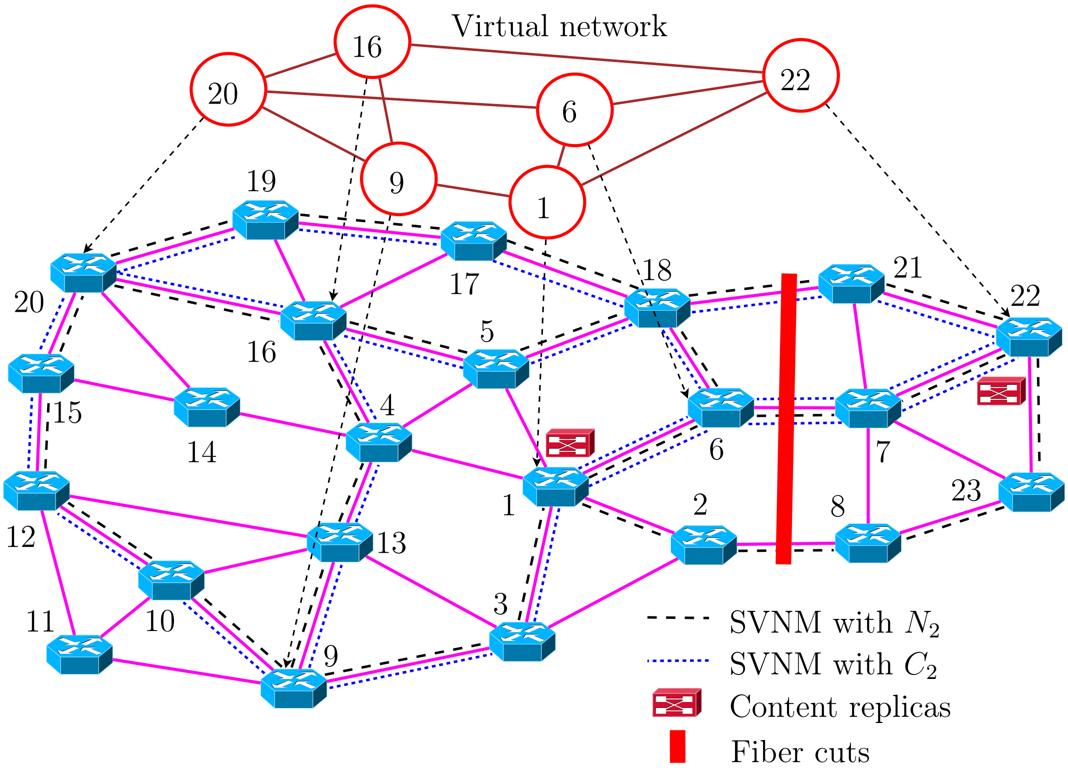


Figure 2.6: Virtual network mapping where content connectivity is survivable against a large-scale disaster in Tokyo23.

than 5 minutes even for the most complex input (e.g., finding C_2 mapping for the 10-node VN in Fig. 2.11.f on Milan52). To provide some theoretical insights into the problem of SVNM with CC, in this chapter, we report only exact/optimal solutions. Considering the problem is static/offline, our formulation is applicable to network topologies of realistic sizes, and leaves heuristic solutions for even larger topologies as future work. Following subsections show numerically that SVNM with CC has a series of unique advantages compared to SVNM with NC.

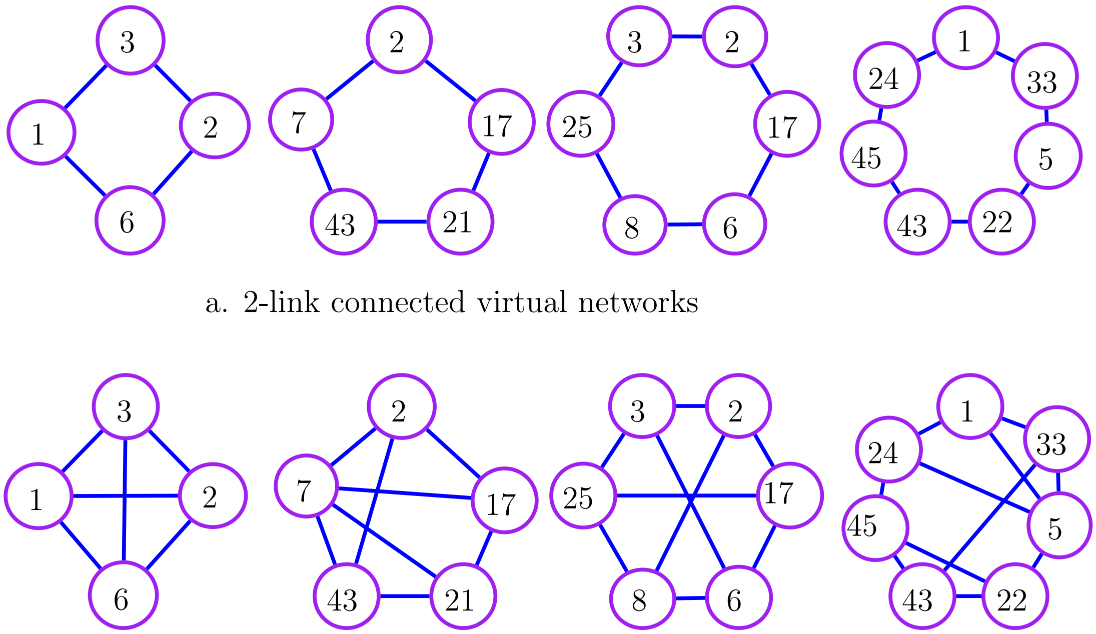
2.5.2 An Example of How SVNM with CC Achieves Higher Survivability

Here, we demonstrate that SVNM with CC has higher survivability against a large-scale disaster compared to SVNM with NC. We consider the mapping of a 6-node VN over the Tokyo23 network in Fig. 2.6. Since the VN and the PN are 3-link connected, we compute SVNM with CC and NC against a random double-link failure (i.e., C_2 and N_2). In Fig. 2.6, C_2 and N_2 are plotted using the dotted line and the dashed line in which most virtual links are mapped on the same lightpaths. In particular, C_2 and N_2 map the virtual links (1, 22) on different lightpaths. The reason is because nodes 1 and 22 host content replicas, hence they require no CC protection. As a result, CC constraints for the two nodes are relaxed and the virtual links (1, 22) can take the shortest path across nodes 6 and 7, and use 6 bandwidth units (both directions). In case of N_2, the virtual links (1, 22) take the longer path across nodes 2, 8, and 23, and use 8 bandwidth units (both directions). In cost comparison, N_2 and C_2 consume 50 and 48 bandwidth units, respectively. Moreover, we can note that failures on physical links (1, 6) and (3, 9) disrupt NC but maintain CC for the VN. In case of a large-scale disaster disrupting multiple physical links (e.g., (2, 8), (6, 7), and (18, 21)), NC for the VN is not survivable. However, CC for the VN is survivable because content replicas available at nodes 1 and 22 are still accessible to all virtual nodes. This example shows that SVNM with CC can guarantee survivability against larger number of failures for a lower cost than SVNM with NC.

2.5.3 Cost Comparison: SVNM with CC vs. SVNM with NC

In this subsection, we find SVNM for the VNs shown in Fig. 2.7 (mapped on Milan52) and Fig. 2.8 (mapped on Tokyo23). We consider 2-link connected and 3-link connected VNs since most physical nodes in the PNs have a nodal degree of three. We assume there are two content replicas in each VN and study how their locations affect C_k cost.

To recall, in Section 2.3, we defined D^t as the replication gain for the virtual node t in a VN. We also concluded that placing a content at a virtual node with a higher replication gain saves more C_k cost. In the following experiments, for each VN, we find SVNM with CC in two scenarios. In the first scenario, content replicas are placed at two virtual nodes with highest replication gains, denoted by superscript g. In the second scenario, content replicas are placed at two virtual nodes with lowest replication gains,



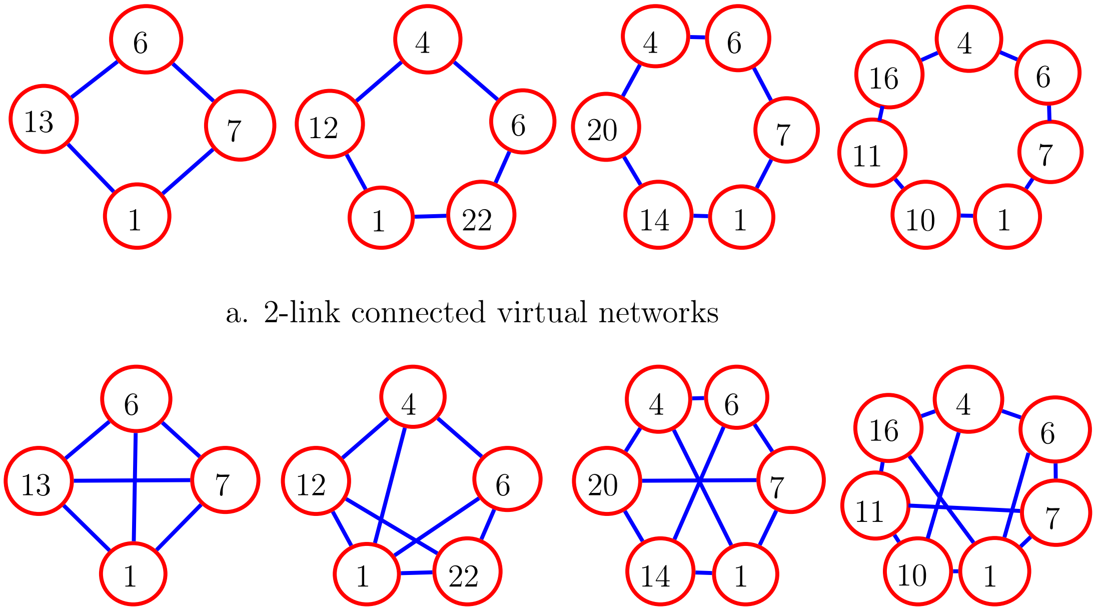
a. 2-link connected virtual networks

b. 3-link connected virtual networks

Figure 2.7: Virtual networks on the Milan52 network.

denoted by superscript b. Moreover, if a VN is 3-link connected, we set $k = 2$ and find N_2, C_2^g, and C_2^b. In case a VN is 2-link connected, we set $k = 1$ and compute N_1, C_1^g, and C_1^b. Figs. 2.9 and 2.10 show the costs to map the VNs in Fig. 2.7 and 2.8 on Milan52 and Tokyo23, respectively (i.e., N_k, C_k^g, and C_k^b for $k = 1,\ 2$). Based on simulation results, we observe the following.

- If two contents are placed at two virtual nodes with zero replication gains, C_k cost is equal to N_k cost. See, e.g., in the following cases: a) C_1^b, 4 nodes, 5 nodes, 6 nodes, and 7 nodes in Fig. 2.9 (i.e., costs to map 2-link connected, 4-node, 5-node, 6-node, and 7-node VNs in Fig. 2.7.a on Milan52 where two content replicas are placed at two nodes with lowest/zero replication gains); b) C_2^b, 4 nodes and 5 nodes in Fig. 2.9 (i.e., costs to map 3-link connected, 4-node and 5-node VNs in Fig. 2.7.b on Milan52 where two content replicas are placed at two nodes with lowest/zero replication gains); and c) C_1^b, 4 nodes, 5 nodes, and 7 nodes in Fig. 2.10 (i.e., costs to map 2-link connected, 4-node, 5-node, and 7-node VNs in Fig. 2.8.a on Tokyo23 where two content replicas are placed at two nodes with lowest/zero replication



a. 2-link connected virtual networks

b. 3-link connected virtual networks

Figure 2.8: Virtual networks on the Tokyo23 network.

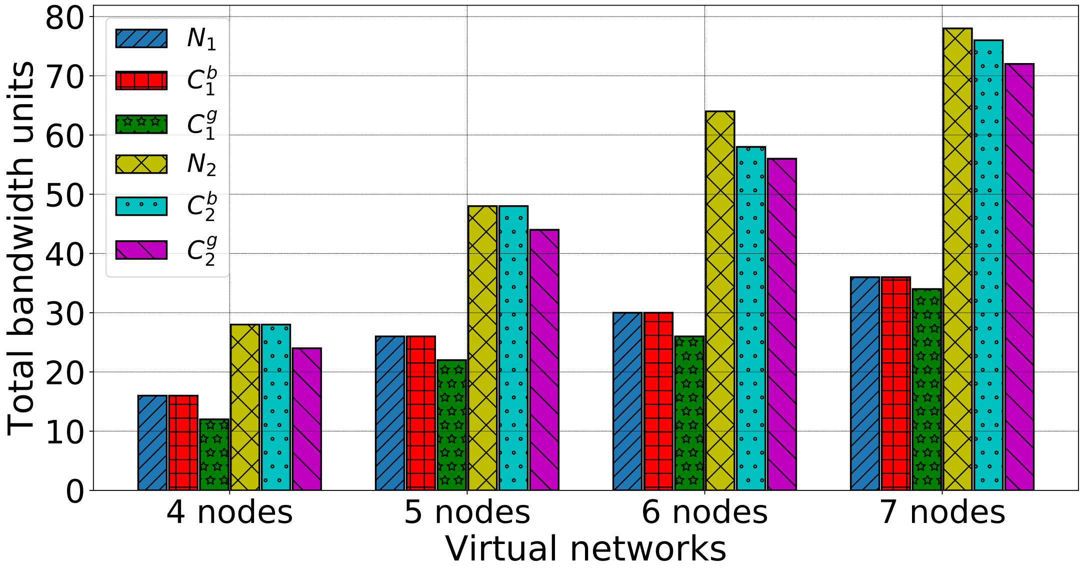


Figure 2.9: Cost of virtual networks in Fig. 2.7 mapped over Milan52.

gains).

- In case there are no two virtual nodes in a VN with zero replication gains, C_k cost is less than N_k cost, regardless of replication locations. We can find this scenario for the following cases: a) C_2^b, C_2^g, 6 nodes and 7 nodes in Fig. 2.9 (i.e., costs to map

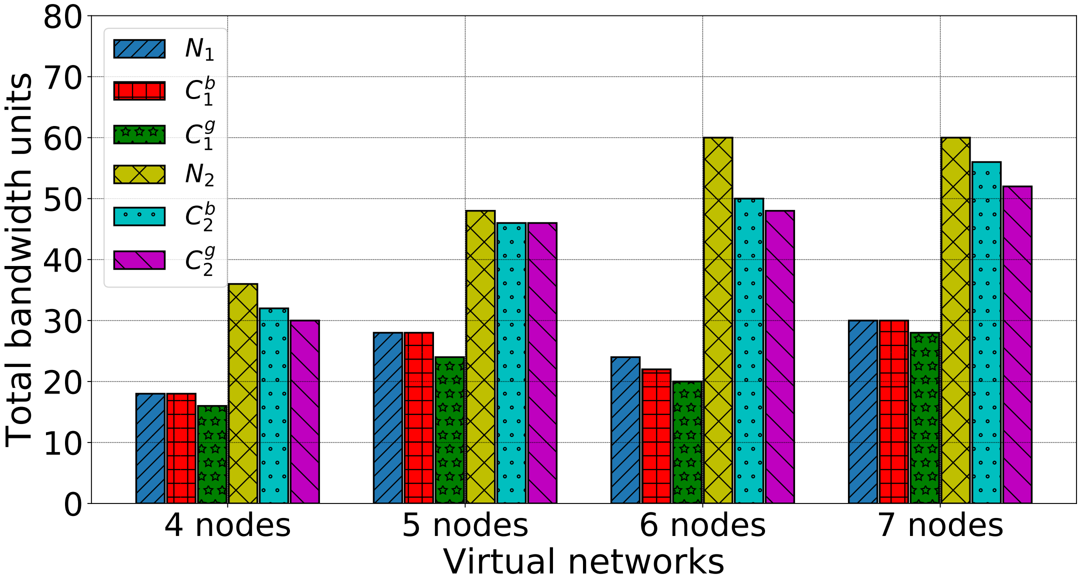


Figure 2.10: Cost of virtual networks in Fig. 2.8 mapped over Tokyo23.

3-link connected, 6-node and 7-node VNs in Fig. 2.7.b on Milan52); b) C_1^b, C_1^g, 6 nodes in Fig. 2.10 (i.e., costs to map 2-link connected, 6-node VNs in Fig. 2.8.a on Tokyo23); and c) C_2^b, C_2^g, 4 nodes, 5 nodes, 6 nodes, and 7 nodes in Fig. 2.10 (i.e., costs to map 3-link connected, 4-node, 5-node, 6-node, and 7-node VNs in Fig. 2.8.b on Tokyo23).

- In most scenarios such as in: a) C_1^b, C_1^g, 4 nodes, 5 nodes, 6 nodes, and 7 nodes in Fig. 2.9 (i.e., costs to map 2-link connected, 4-node, 5-node, 6-node, and 7-node VNs in Fig. 2.7.a on Milan52); b) C_2^b, C_2^g, 4 nodes and 5 nodes in Fig. 2.9 (i.e., costs to map 3-link connected, 4-node and 5-node VNs in Fig. 2.7.b on Milan52); and c) C_1^b, C_1^g, 4 nodes, 5 nodes, and 7 nodes in Fig. 2.10 (i.e., costs to map 2-link connected, 4-node, 5-node, and 7-node VNs in Fig. 2.8.a on Tokyo23), there are two virtual nodes with zero replication gains and two virtual nodes with non-zero replication gains; hence, a careful placement of content replicas at optimal locations (i.e., the two nodes with highest/non-zero replication gains) saves C_k cost compared to N_k.

Table 2.2 shows the cost saving of C_k compared to N_k in percent for the VNs in Figs. 2.7 and 2.8 when two content replicas are placed at optimal locations (i.e., two

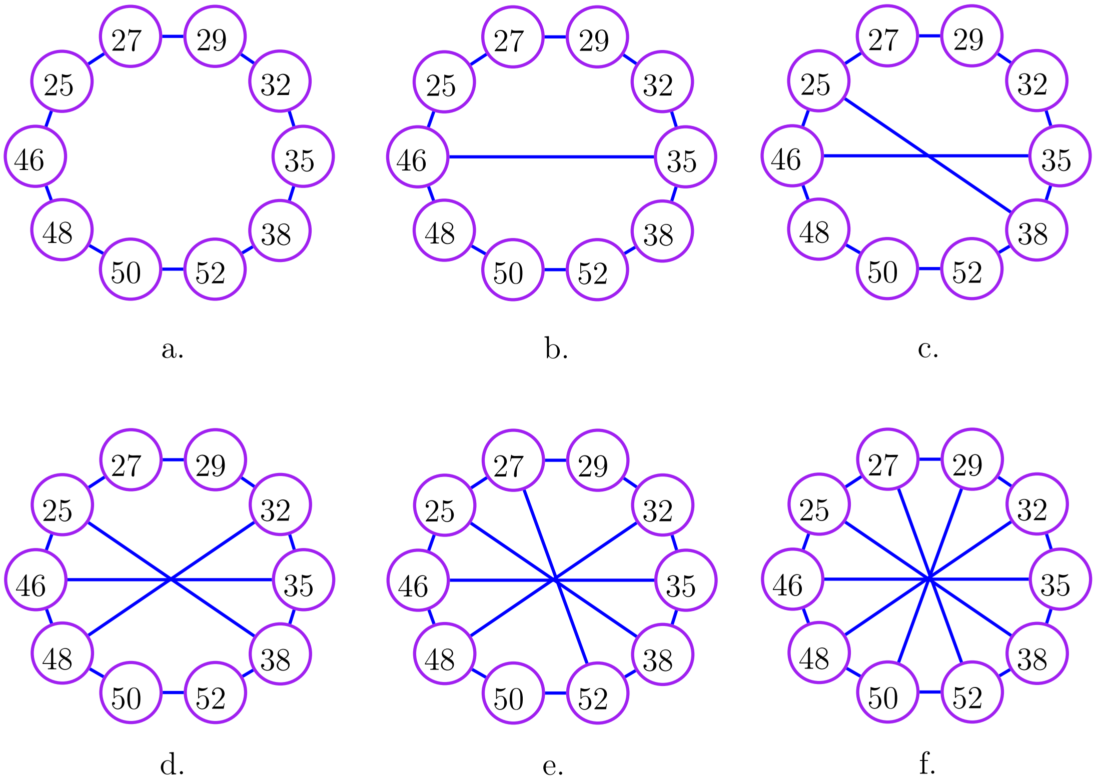
#Nodes	**Milan52 network**		**Tokyo23 network**	
	C_1^g	C_2^g	C_1^g	C_2^g
4	25.0	14.3	11.1	16.7
5	15.4	8.3	14.3	4.2
6	15.4	12.5	16.7	21.5
7	5.6	7.7	6.7	14.3
Avg. [%]	15.3	10.7	12.2	14.2

Table 2.2: C_k cost saving compared to N_k cost in percent when two content replicas are placed at optimal locations.

content replicas are placed at two virtual nodes with highest replication gains, denoted by superscript g). On the second and fourth columns, we report the cost saving of C_1^g compared to N_1 and their average values. Similarly, the third and fifth columns show the cost saving of C_2^g compared to N_2 and their average values. Overall, a careful plan for content replication saves C_k network bandwidth consumption (i.e., total number of bandwidth units) on the order of 10% compared to N_k. However, higher survivability and cost saving are not the only advantages of SVNM with CC. In what follows, we introduce a new metric, called content/service availability, and extend SVNM with CC potential to another significant advantage.

2.5.4 Availability Comparison: SVNM with CC vs. SVNM with NC

Availability is the probability that a system will be found in the operating condition at a random time [31]. We define content availability for a network as the probability that every node in the network is content-connected against a certain failure scenario. In this section, we show that SVNM with CC has significantly higher availability than SVNM with NC. For a VN being mapped over a PN, we use Z_c^k and Z_n^k to denote the number of k-link combinations in the PN whose failures disconnect the CC and NC of the VN, respectively. Content availability (A_c) and network availability (A_n) for the VNs are given



a. b. c.

d. e. f.

Figure 2.11: A family of virtual networks with the same virtual nodes and different virtual links for content availability computation.

by:

$$A_c = \frac{Z^k \text{ - } Z_c^k}{Z^k}, \; A_n = \frac{Z^k \text{ - } Z_n^k}{Z^k},$$

where $Z^k = C(|L_P|, k)$ is the total number of k-link combinations in the PN (defined in Section 2.3).

In the first experiment of this subsection, we consider two 10-node VNs in Figs. 2.11.a and 2.11.f, and map them on the Milan52 network. The two VNs have the same virtual nodes. However, the VN in Fig. 2.11.f has more virtual links. Moreover, for the 2-link connected VN (Fig. 2.11.a), we find N_1 and C_1. In case the VN is 3-link connected (Fig. 2.11.f), we find N_2 and C_2. Table 2.3 shows Z_c^k, Z_n^k, and Z^k, $k = 1..4$, for the two VNs when they are mapped over the Milan52 network. As can be seen from the table, there is no single-link failure disconnecting the 2-link connected VN. Also, there is no double-link failure disrupting the 3-link connected VN. In other words, the 2-link connected VN and the 3-link connected VN maintain 100% CC and NC survivability against single-link failures ($k = 1$) and double-link failures ($k = 2$), respectively. *In other cases (i.e., $k \geq 2$ for 2-link connected VNs and $k \geq 3$ for 3-link connected VNs), the number of k-link combinations disconnecting CC is significantly smaller than the number of k-link combinations disconnecting NC.* For example with the 2-link connected VN, the number of double-link failures disrupting CC is decreased by 42% compared to the number of double-link failures disconnecting NC. Similarly, with the 3-link connected VN, the number of three-link failures disconnecting CC is reduced by 30% compared to the number of three-link failures disconnecting NC.

So far, we have studied the problem of mapping a VN over a PN where important input parameters such as the VN, the PN, and a set of content replicas are given, and hence fixed. We showed that service continuity can be ensured using SVNM with CC instead of SVNM with NC with better survivability and lower cost. *Let us now try to address a fundamental question: is it better to add virtual links or to add content replicas?* To answer this question, we study how number of virtual links and number of content replicas in a VN affect content/service availability.

Now, let us consider a family of VNs with 10 virtual nodes,

$$N_V = \{25, 27, 29, 32, 35, 38, 46, 48, 50, 52\}.$$

We consider the 2-link-connected VN in Fig. 2.11.a as the reference VN, and then add more virtual links to form the VNs in Figs. 2.11.b, 2.11.c, 2.11.d, 2.11.e, and 2.11.f.

In this experiment, we evaluate content availability, A_c, for the VN in Fig. 2.11 against

k	Z^k	**2-link-connected VN**		**3-link-connected VN**	
		Z_n^k	Z_c^k	Z_n^k	Z_c^k
1	202	0	0	0	0
2	20301	337	195	0	0
3	1353400	2168	1717	514	358
4	67331650	9855	8834	26726	19152

Table 2.3: Number of k-link combinations on the Milan52 network disconnects CC and NC of the VN in Fig. 2.11.

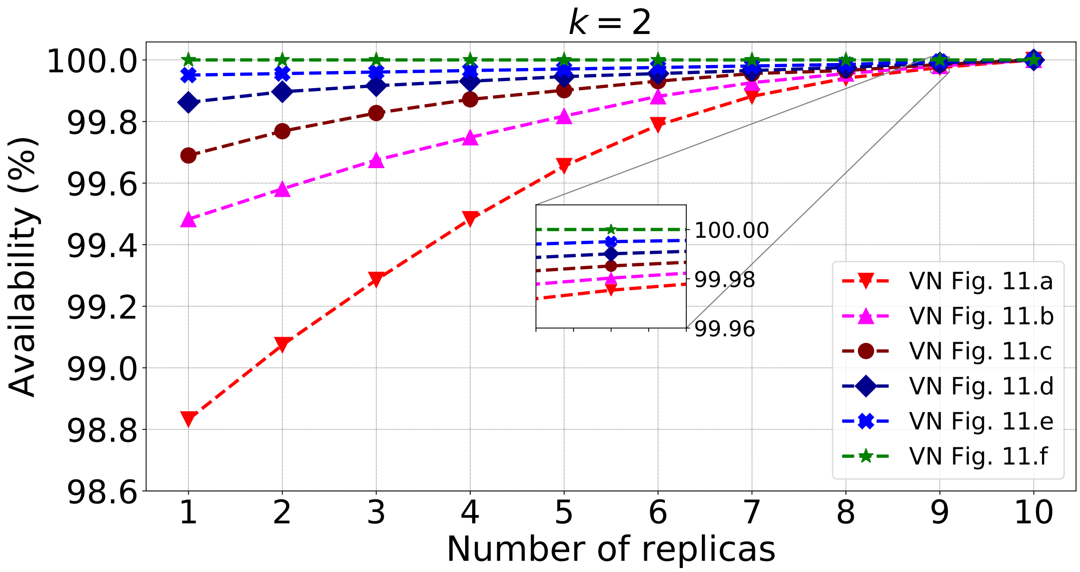


Figure 2.12: Content availability against double-link failures as a function of number of content replicas for the VN in Fig. 2.11.

double-link failures if there is/are: 1) one content replica (node 25); 2) two content replicas (nodes 25 and 27); 3) three content replicas (nodes 25, 27, and 29); 4) four content replicas (nodes 25, 27, 29, and 32); 5) five content replicas (nodes 25, 27, 29, 32, and 35); 6) six content replicas (nodes 25, 27, 29, 32, 35, and 38); 7) seven content replicas (nodes 25, 27, 29, 32, 35, 38, and 46); 8) eight content replicas (nodes 25, 27, 29, 32, 35, 38, 46, and 48); 9) nine content replicas (nodes 25, 27, 29, 32, 35, 38, 46, 48, 50); and ten content replicas (all virtual nodes). Fig. 2.12 shows content availability against double-link failures as a function of the number of content replicas for every VN in Fig. 2.11 (i.e., Figs. 2.11.a, 2.11.b, 2.11.c, 2.11.d, 2.11.e, and 2.11.f).

As expected, adding more virtual links or placing more content replicas improves content availability. As we showed in Section 2.3 (Theorem 3), if there is only one content replica in the VN, SVNM with CC is the same as SVNM with NC; thus, content availability is equal to network availability. When more content replicas are added to the VNs, for example in case the number of content replicas is increased from one to two in Figs. 2.11.a, 2.11.b, and 2.11.c, content availability is significantly improved. Let us consider a situation in which the virtual nodes in Fig. 2.11 require a content availability higher than 99.8%. We can fulfill this requirement using: 1) 10 virtual links and 6 content replicas

(Fig. 2.11.a); 2) 11 virtual links and 5 content replicas (Fig. 2.11.b); or 3) 12 virtual links and 3 content replicas (Fig. 2.11.c). For Figs. 2.11.d and 2.11.e, because the VNs are highly connected (i.e., the average node degree > 2.6), the number of double-link failures disrupting CC is small. Consequently, content availability is higher than the minimum requirement (i.e., 99.8%) regardless of content replica number. In case the VN is 3-link connected (i.e., including 15 virtual links) such as in Fig. 2.11.f, C_2 and N_2 exist and content availability is guaranteed 100% for every virtual node against double-link failures, regardless of the number and locations of replicas. In an extreme scenario, which is not practical, where content replicas are available at every virtual node, content availability is 100% for all VNs. In a practical situation where a virtual network requests a given target of content availability, this experiment demonstrates which are the options to fulfill this requirement. To meet a certain target of content availability, an operator has two options: 1) adding more virtual links; or 2) adding more content replicas. In general, adding virtual links implies adding more network resources while adding content replicas means adding more computing/storage in DCs and network resources for synchronization. For this issue, only the operator knows which option is optimal cost-wise (i.e., it depends on the relative cost of adding a virtual link vs. adding a content replica). Therefore, the decision between adding more content replicas or augmenting additional virtual links to a VN may vary from operator to operator.

2.6 Conclusion

We have proposed a novel approach to map a virtual network over a physical network with content connectivity against multiple link failures. We developed necessary and sufficient conditions for the existence of survivable content-connected mapping and introduced the content cutset concept. Since the number of content cutsets is considerably smaller than the number of network cutsets, the problem formulation reduces the time required to find a survivable mapping. Therefore, we can solve various complex problem instances with a larger number of nodes and links while maintaining the solution optimality. Numerical results show that survivable virtual network mapping with content connectivity has higher survivability than survivable virtual network mapping with network connectivity, particularly against large-scale failures, saves network bandwidth, and significantly improves service availability. It is worth mentioning that the virtual network in this chapter is leased to a customer by a service provider and the problem in this chapter is static. In next chapter, we consider a scenario where contents are frequently updated in data centers and requests come, hold, and depart in a dynamic manner.

Chapter 3

Reliable Provisioning with Degraded Service Using Multipath Routing from Multiple Data Centers in Optical Metro Networks

With the adoption of edge computing, several data centers are available within the footprint of an optical metro network, and contents are replicated in multiple locations. Such a wide content replication offers a unique opportunity to provide better services to users, especially for content-based services, e.g., video delivery. Thus, a service-provisioning scheme can embrace this opportunity to optimize network resource utilization, improve reliability, and achieve lower latency. In this chapter, we propose a reliable service-provisioning scheme that selects the optimal subset of data centers hosting the desired content and inversely multiplexes a content request over multiple link-disjoint paths. We formulate an integer linear program (ILP) and develop heuristics for the problem, and use them to solve various complex and realistic network instances. Numerical data show that, compared to conventional service-provisioning schemes such as multipath routing from a single data center or dedicated-path protection, our proposed scheme efficiently utilizes network resources, improves reliability, and reduces latency; hence, it is suitable for the above-mentioned services.

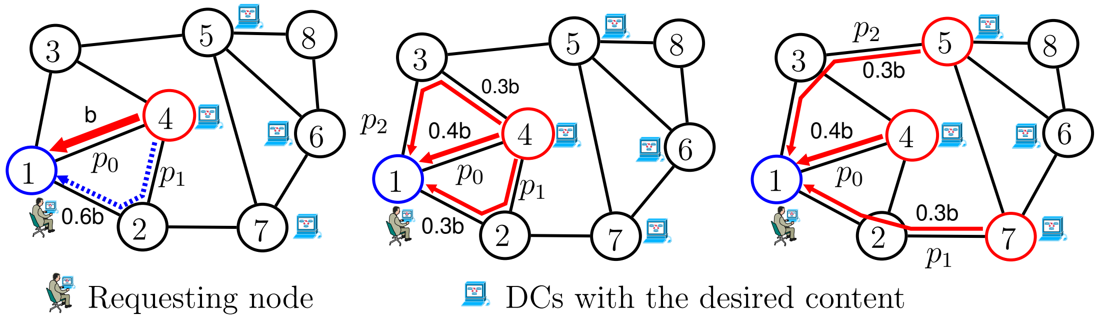
3.1 Introduction

Optical metro networks are attracting significant investments to evolve from a rigid ring-based aggregation infrastructure to a composite network-and-computing ecosystem where new applications and services can be implemented and supported [25]. In particular, with the adoption of edge computing, several DCs are now available within the footprint of an optical metro network. Typically, micro DCs are available in metro-access nodes, medium-size DCs are available in metro-core nodes, while hyper-scale DCs are available in core network nodes, and they communicate with the metro network via metro-core backbone nodes as gateways. Hence, with edge computing, contents (e.g., media files, applications, web services, and documents) are now widely replicated in multiple DCs closer to users to offload core network traffic and to lower latency [32,33].

Such a wide replication of contents in multiple locations offers a unique opportunity to provide better services to users, especially for content-based services. Among such services, video delivery is playing the main role, as, by 2022, it will be 79% of the world's mobile data traffic and 82% of all consumer Internet traffic [1]. Other content-based services, such as augmented reality (AR) and virtual reality (VR), are emerging, which allow users to interact intuitively with the environment through six degrees of freedom [34]. These services require high bandwidth, low latency, reliable connectivity, and are classified as mobile broadband reliable low-latency communications (MBRLLC) in the vision of 6G communications [35]. While some of these services require full protection, others can continue to operate with reduced, i.e., degraded, quality in case of failures and can be served with partial protection (e.g., a video stream can switch to a lower resolution depending on available bandwidth).

High-capacity optical metro networks are exposed to many threats such as malicious attacks, equipment failures, human errors (e.g., misconfigurations), and natural and human-made large-scale disasters (e.g., earthquakes, hurricanes, and terrorist attacks). To ensure reliability, protection and restoration schemes are traditionally used. In a protection scheme, extra network resources are reserved when a connection is provisioned. Conventionally, a pair of paths is provided to a connection: one is used to carry traffic during normal operation, referred to as primary path; and the other path, referred to as backup path, is reserved and will be activated after a failure occurs on the primary path. In a restoration scheme, no extra resources are reserved for the backup path, and the network must react to find an alternative path after a failure occurs on the primary path [11]. Since failures are hard to predict and statistically rare, providing protection in a dynamic network environment, especially full protection against multiple failures, would require massive and economically-unsustainable bandwidth overprovisioning. On the other hand, a restoration scheme would require a longer recovery time and provide no guarantee to restore a disrupted path.

This chapter concentrates on multipath routing, a flexible and resource-efficient protection scheme in which a service request is provisioned over multiple paths by routing part of the requested bandwidth on each path [36, 37]. With respect to the baseline multipath routing, we consider the opportunity to route different paths towards different destinations, representing different metro DCs hosting the required content. To illustrate our proposed service-provisioning scheme, in Fig. 3.1, we consider a dynamic network environment where, at time t, a user at node 1 is requesting for a content replicated in multiple DCs at nodes 4, 5, 6, and 7. In addition, the user requires bandwidth b during normal operation and can tolerate degraded service (i.e., degraded bandwidth) $0.6b$ in case of a single-link failure. Note that, even though multiple-link/node failures can occur, a single-link failure is still the dominant failure scenario in an optical network [3–5, 38]. Also, due to the asymmetric traffic characterizing content retrieval, we only consider downstream traffic (i.e., from DCs to the requesting node). Conventionally, a *dedicated-path protection* (DPP) scheme selects the optimal DC (e.g., the DC at node 4 which is closest to the requesting node) and reserves a pair of primary and backup paths (e.g., p_0 and p_1) for the request [39]. As shown in Fig. 3.1.a, the bandwidths on the primary and backup paths are b and $0.6b$, respectively. In case of a failure occurring on the primary path, the requested degraded service is still guaranteed after the backup path is activated. In total, DPP requires bandwidth $1.6b$ over the primary and backup paths and occupies network bandwidth $2.2b$. Here, we define the *network bandwidth* as the sum of the bandwidth on



a. DPP b. MPSD c. MPMD

Figure 3.1: Different reliable service-provisioning schemes.

each path weighted by the number of hops (i.e., $2.2b = b + 2 * 0.6b$).

Fig. 3.1.b shows a different protection scheme using *multipath routing from a single data center* (MPSD) [36, 37, 40–42]. In this scenario, data from the closest DC at node 4 to the requesting node are simultaneously transmitted on three link-disjoint paths p_0, p_1, and p_2 with bandwidths of $0.4b$, $0.3b$, and $0.3b$, respectively. In case a failure occurs on a path, the requested degraded service is still fulfilled since survivable bandwidth remains at least $0.6b$. In total, MPSD requires bandwidth b over three paths and occupies network bandwidth $1.6b$. In Figs. 3.1.a and 3.1.b, DPP and MPSD share a major shortcoming as they do not provide protection against failures in the source DC. For instance, in case of a failure occurring in the DC at node 4, provisioned services are disrupted.

In Fig. 3.1.c, we describe a service-provisioning scheme where the user at node 1 is simultaneously served by three different DCs at nodes 4, 5, and 7 on three link-disjoint paths p_0, p_1, and p_2 with bandwidths of $0.4b$, $0.3b$, and $0.3b$, respectively. Hereafter, we refer to this service-provisioning scheme as *multipath routing from multiple data centers* (MPMD). In case a path is disrupted, MPMD ensures that survivable bandwidth remains at least $0.6b$; hence, the required degraded service is still guaranteed. Compared to DPP and MPSD, MPMD also provides protection against failures in DCs (i.e., survivable bandwidth remains at least $0.6b$ if a failure occurs in one of the serving DCs, e.g., at node 4). In total, MPMD requires bandwidth b over three paths and occupies network bandwidth $1.6b$. For each content request, MPMD must: 1) select the optimal subset of DCs hosting the desired content, 2) find link-disjoint paths from each selected DC to the requesting node, and 3) allocate bandwidth to each path such that the total requested bandwidth during normal operation and degraded service in case a path is disrupted are fulfilled. In the literature, this service-provisioning scheme is often referred as an inverse manycast scheme to indicate that source nodes (i.e., optimal subset of DCs) must be selected from a larger candidate set [43]. Compared to an inverse multicast scheme where source nodes are specified ahead of time, an inverse manycast scheme has greater flexibility in choosing the source nodes from which data (e.g., contents) are retrieved. Here, MPMD exploits the wide replication of a content in multiple DCs to provide better services to users and requires no additional resources (e.g., storage capacity) in each DC. Note that the underlying multipath routing can be implemented based on techniques such as virtual concatenation (VCAT) and link capacity adjustment scheme (LCAS), in which, different parts of the same content can be transported on different lightpaths from different DCs [44, 45]. In case one path is disrupted, total offered bandwidth can be reduced to a degraded level using bandwidth squeezing restoration [46].

In this chapter, we propose MPMD as a reliable service-provisioning scheme to fulfill content requests (e.g., video-on-demand, VR, or AR requests) in a dynamic network environment while guaranteeing strict requirements on bandwidth, and improving reliability and latency. This proposed service-provisioning scheme enjoys the benefits of multipath routing, provides protection against network (e.g., link) and content source (e.g., DC) failures, and uses minimal additional network resources due to the nature of multipath routing. We formulate the MPMD problem as an ILP, develop two scalable heuristics, and use them to solve various complex network instances. Numerical data show that, compared to conventional service-provisioning schemes such as DPP and MPSD, MPMD efficiently utilizes network resources, provides higher reliability, and reduces latency; hence, it is highly suitable for the emerging content services.

The rest of this study is organized as follows. In Section 3.2, we review related works. In Section 3.3, we formulate the MPMD problem as an ILP. In Section 3.4, we develop heuristics for the MPMD problem. In Section 3.5, we perform numerical validation in various scenarios and compare the performance of MPMD to those of reference protection strategies (e.g., MPSD, DPP). We conclude this study in Section 3.6.

3.2 Related Works

Some studies have been conducted on reliable service provisioning using multipath routing. In [36, 37, 40], the authors developed algorithms for multipath routing in a static scenario in which, for each traffic demand, the source and destination nodes are specified ahead of time. Due to the nature of multipath routing, the algorithms ensure a certain level of degraded service in case of failures on one or several paths. In [41], the authors extended their work in [40] to a dynamic scenario where survivable bandwidth is guaranteed in case a path is disrupted. In [43], the authors studied the problem of routing and wavelength assignment for static manycast demands in wavelength-division multiplexing (WDM) networks. They proposed a solution for upstream traffic in which the requesting node sends data to several nodes in a larger set of candidate nodes using multipath routing and manycast.

The authors in [47] leveraged the concept of CC in DC networks and developed algorithms to place contents at optimal DCs for a static scenario such that K node/link-disjoint paths are guaranteed from the requesting node to the DCs hosting the desired content. In [48], the authors proposed dynamic service provisioning in elastic optical network (EON) with hybrid single/multipath routing (HSMR). They investigated the flexibility of selection between single-path routing, multipath routing, and how the paths between a pair of nodes are computed. Numerical results showed that HSMR with online path computation (OPC) can achieve the lowest bandwidth-blocking probability among all HSMR schemes. However, HSMR was designed for unicast requests and did not consider survivability and differential delay of paths. In [49, 50], the authors developed reliable service-provisioning schemes using multipath routing and inverse manycast. However, the authors developed the solution for a static scenario and the paths between each pair of nodes are pre-computed (i.e., offline path computation). This assumption makes the solutions less practical for content-retrieval applications where requests (e.g., video on demand) arrive, hold, and depart dynamically, the network topology (based on availability of resources) changes over time, and OPC is more practical.

In this study, our focus is on the dynamic problem for content requests, leveraging multipath routing and inverse manycast while ensuring strict requirements on bandwidth, and improving reliability and latency.

3.3 Reliable Provisioning with Degraded Service Using MPMD

In this section, we formally state the MPMD problem and formulate it as an ILP (MPMD-ILP).

3.3.1 Problem Statement

In this study, each dynamic content request, θ, is characterized by a tuple $\theta = (t, n, c, b, m)$ where t is arrival time (s), n is the requesting node, c is the desired content (i.e., content ID), b is requested bandwidth (Mbps), and m is the ratio of survivable bandwidth to requested bandwidth in case a path is disrupted. In other words, if a path is disrupted, survivable bandwidth must remain at least $m * b$. We consider a graph $G_t(V_t, E_t)$ to represent a network where V_t is the set of nodes with available computing capacity and E_t is the set of links with available bandwidth at request arrival time. Since content requests come, hold, and depart, the sets V_t and E_t can vary over time. Moreover, if the network finds insufficient resources for an incoming content request at its arrival time, the content request is blocked. The desired content, c, has size h (GB), and is replicated in multiple DCs denoted by the set of nodes D, $D \subset V_t$, $|D| \geq 2$. Without loss of generality, the requesting node does not host the desired content, i.e., $n \notin D$. Also, we assume that each selected DC can support only one path. To guarantee degraded service (i.e., bandwidth $m * b$) in case a path is disrupted, total offered bandwidth over all paths, b' (Mbps), can be larger than requested bandwidth (i.e., $b' \geq b$). Moreover, if a request is offered bandwidth b', it departs at $t' = t + 8000 * h \,/\, b'$ (s), where $8000 * h \,/\, b'$ is the holding time (s) of the request (i.e., h (GB) and b' (Mbps)). Here, we assume that each path has the same bandwidth and retrieves an equal amount of the desired content (i.e., holding time is equal in each DC). Note that we consider large contents, so a content's transmission delay is the major contributor to its total delay (and its propagation delay through the network is relatively negligible).

In this chapter, we focus on MPMD's bandwidth efficiency, survivability, and latency; and leave the choice of technologies in the physical layer (e.g., traffic grooming, wavelength assignment, impairment, WDM, and EON) as open research topics. Also, even though the ILP and heuristic algorithms developed in this and next sections can be applied for any optical network, we analyze their performance using an optical metro network covering an urban area up to tens of kilometers because, in most content-retrieval applications, a content is replicated in multiple edge DCs close to the requesting node which is typically a CO. The data from the CO is ultimately delivered to an end user via either a mobile or fixed network. Note that, when MPMD is applied to a backbone network covering thousands of kilometers, constraints on impairments such as noise and nonlinearity must be incorporated in the ILP and heuristic algorithms. In case of space-division multiplexing (SDM) realized by EON and multi-core fibers (MCFs), the additional constraint on survivability of lightpaths (possibly in different fibers on different strands) must be considered [51]. We reserve this topic as a possible extension of our current work.

The MPMD problem can be formally stated as follows. *Given* the content request $\theta = (t, n, c, b, m)$, the network graph at request arrival $G_t(V_t, E_t)$, and availability of the desired content characterized by its size h and set of hosting DCs D, *find*: 1) optimal subset of DCs hosting the desired content (for convenience, we use D_0 to denote this optimal subset, $D_0 \subseteq D$), 2) link-disjoint paths from each DC in D_0 to requesting node n, and 3) bandwidth on each path *such that* total requested bandwidth b during normal operation and degraded service $m * b$ in case a path is disrupted are fulfilled.

Noting that multiple objectives cannot be optimized simultaneously, a weighted objective function is defined. The objective is to minimize total bandwidth over all paths, total network bandwidth (weighted sum), and total propagation delay of the paths from each selected DC to the requesting node.

The optimal solution is subject to constraints on the available capacity of each physical link, maximum differential delay between paths (i.e., differential delay constraint (DDC) which is maximum difference of propagation and processing delay between paths that can be compensated at the receiver [42]), and survivability of paths.

In the following sections, we develop the algorithms for MPMD-ILP and MPMD heuristics. It is worth noting that the algorithms developed in this chapter are for a single content arrival request. To obtain numerical results for a dynamic network environment, the algorithms must be integrated into a dynamic simulation framework.

3.3.2 Mathematical Formulation

Inputs:

- $G_t(V_t, E_t)$: network graph at request arrival time.
- $\theta = (t, n, c, b, m)$: content request.
- h: size of the desired content (GB).
- D: DCs hosting the desired content, $D \subset V_t$, $|D| \geq 2$.
- Λ_t: hash table representing capacity of physical links where each key-value pair $(i,j) : \lambda_t^{i,j}$, $\forall (i,j) \in E_t$, is link (i,j)'s available capacity at request arrival time (Mbps).
- Ξ: hash table representing propagation delay of physical links with each key-value pair $(i,j) : \xi^{i,j}$, $\forall (i,j) \in E_t$, being propagation delay of link (i,j) (μs).
- Ω: maximum differential delay between paths (μs).

Variables:

- $w_{d,n}$: a binary variable, and $w_{d,n} = 1$ if requesting node n uses DC d, $d \in D$; and 0 otherwise.

- $x_{d,n}$: an integer variable denoting the bandwidth reserved on the path from DC d, $d \in D$, to requesting node n.

- $y^{i,j}_{d,n}$: an integer variable denoting the mapping of the path from DC d, $d \in D$, to requesting node n on physical links (i,j). Here, $y^{i,j}_{d,n} = x_{d,n},\ x_{d,n} > 0$ if the path from DC d to requesting node n is mapped on physical link (i,j); and 0 otherwise.

- $z^{i,j}_{d,n}$: a binary variable, and $z^{i,j}_{d,n} = 1$ if the path from DC d, $d \in D$, to requesting node n is mapped on link (i,j); and 0 otherwise.

Objective function:

$$\min_{\forall\theta} \left(\alpha * \sum_{d\in D} x_{d,n} + \beta * \sum_{\substack{d\in D,\\ (i,j)\in E_t}} y^{i,j}_{d,n} + \gamma * \sum_{\substack{d\in D,\\ (i,j)\in E_t}} \xi^{i,j} z^{i,j}_{d,n} \right). \tag{3.1}$$

Subject to:

$$\sum_{d\in D} w_{d,n} \geq 2, \ \ \forall\theta. \tag{3.2}$$

$$\sum_{d\in D} x_{d,n} \geq b, \ \ \forall\theta. \tag{3.3}$$

$$w_{d,n} \geq x_{d,n}/\Psi, \ \ \forall d \in D, \ \forall\theta. \tag{3.4}$$

$$\sum_{d\in D} y^{i,j}_{d,n} \leq \lambda^{i,j}_t, \ \ \forall (i,j) \in E_t, \ \forall\theta. \tag{3.5}$$

$$\sum_{j:(j,i)\in E_t} y^{j,i}_{d,n} - \sum_{j:(i,j)\in E_t} y^{i,j}_{d,n} = \begin{cases} -x_{d,n} & \text{if } i = d \\ +x_{d,n} & \text{if } i = n, \\ 0 & \text{o/w} \end{cases} \quad \substack{\forall i\in V_t,\\ \forall d\in D,\\ \forall\theta.} \tag{3.6}$$

$$z^{i,j}_{d,n} \geq y^{i,j}_{d,n}/\Psi, \ \ \forall d \in D, \ \forall (i,j) \in E_t, \ \forall\theta. \tag{3.7}$$

$$x_{d,n} - \sum_{d \in D} x_{d,n} \leq -m * b, \ \ \forall d \in D, \ \forall \theta. \tag{3.8}$$

$$\sum_{d \in D} z_{d,n}^{i,j} \leq 1, \quad \forall (i,j) \in E_t, \ \forall \theta. \tag{3.9}$$

$$\left| \sum_{(i,j) \in E_t} \xi^{i,j} z_{d_1,n}^{i,j} - \sum_{(i,j) \in E_t} \xi^{i,j} z_{d_2,n}^{i,j} \right| \leq \Omega, \quad \forall d_1, d_2 \in D : d_1 \neq d_2, \ \forall \theta. \tag{3.10}$$

In objective function (3.1), we introduce the scalars α, β, and γ to control the weight of each term. The first summation minimizes total bandwidth over all paths, the second summation minimizes total network bandwidth, and the third summation minimizes total propagation delay of the paths from each selected DC to the requesting node. We obtained numerical results for different values of α, β, γ, and derived the following observations. First, if α is very large compared to β and γ (e.g., $\alpha = 500000$, $\beta = 1$, and $\gamma = 1$), MPMD-ILP tends to find a solution with minimal total bandwidth over all paths to satisfy a content request, while the paths from each selected DC to the requesting node can become longer. These longer paths typically imply that the total propagation delay may not be minimal. Second, if γ is very large compared to α and β (e.g., $\alpha = 1$, $\beta = 1$, and $\gamma = 500000$), MPMD-ILP tends to find paths with minimal total propagation delay. As a result, MPMD-ILP normally avoids circuitous paths and uses a lower number of paths (e.g., two paths to the first and second closest DCs) to admit a content request. However, the total bandwidth over all paths must be larger to guarantee a degraded service level in case a path is disrupted. Third, if β is very large compared to α and γ (e.g., $\alpha = 1$, $\beta = 500000$, and $\gamma = 1$), MPMD-ILP tends to find a solution with minimal network bandwidth, while the total bandwidth over all paths might not be minimal. Moreover, since network bandwidth is directly related to number of hops, minimal network bandwidth implicitly tends to reduce total propagation delay. Numerical results also showed that, compared to the other two weight tuples, this weight tuple has a higher acceptance ratio of incoming requests. For these reasons, in our simulation setup, we set $\alpha = 1$, $\beta = 500000$, and $\gamma = 1$.

Constraint (3.2) ensures that each request is simultaneously served by at least two DCs (i.e., multiple DCs). Constraint (3.3) guarantees that the total bandwidth over all paths for each request is at least b during normal operation. Constraint (3.4) sets the binary variable $w_{d,n} = 1$ if the requesting node n uses DC d (i.e., $x_{d,n} > 0$) where Ψ is a large positive integer, e.g., 10000. Constraint (3.5) requires that the mapping of the request on the physical network does not exceed the capacity of each physical link at request arrival time. Constraint (3.6) enforces flow conservation in which, for each path, traffic originates from DC d and ends at requesting node n. At a transit node, input traffic is equal to output traffic. Constraint (3.7) computes a binarization of the integer variable $y_{d,n}^{i,j}$ and assigns it to $z_{d,n}^{i,j}$. Constraint (3.8) guarantees that the desired degraded service is satisfied (i.e., survivable bandwidth must remain at least $m * b$ in case a path is disrupted). Constraint (3.9) restricts the mapping of the request on the physical network such that a single physical link cannot be shared by two or more paths. In other words, traffic from the DCs in D_0 to the requesting node is carried on link-disjoint paths. Therefore, constraints (3.8) and (3.9) strictly enforce survivable bandwidth to be at least $m * b$ in case a path is disrupted. Constraint (3.10) ensures that the differential delay of two distinct paths fulfills DDC. Since this work focuses on MPMD's bandwidth efficiency, survivability, and latency, we assume that physical-layer components (e.g., transceivers, transponders, and computing and storage capability) in each node can provide enough throughput and skip the constraints on these components in the MPMD-ILP.

In our MPMD-ILP, the numbers of variables and constraints for each request are upper bounded by $2 * |D| * (1 + |E_t|)$ and $2 * (1 + |D| + |E_t|) + |D| * (|V_t| + |E_t|) + \binom{|D|}{2}$, respectively. Here, we use $|\cdot|$ to denote the cardinality of a set and $\binom{\cdot}{\cdot}$ is the combination without repetition. Since $\binom{|D|}{2}$ can be reduced to $|D| * (|D| - 1)/2$, the number of variables increases linearly with number of hosting DCs and number of physical links, the number of constraints increases linearly with number of physical nodes and number of physical links, while it increases quadratically with number of hosting DCs.

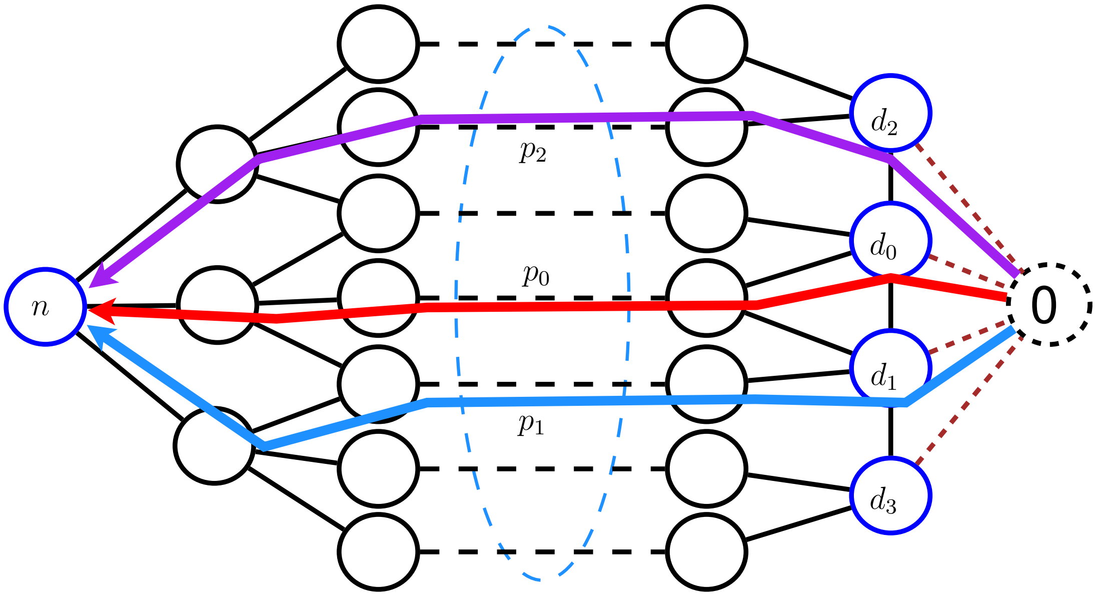


Figure 3.2: An auxiliary graph with a dummy node and dummy links.

3.4 Heuristics For Reliable Provisioning with Degraded Service Using MPMD

Since the MPMD-ILP presented in Section 3.3 is intractable for large network instances, it is impractical in a dynamic network environment where a fast solution is more desirable. In this section, we propose various heuristic algorithms for reliable provisioning with degraded service using MPMD. Before that, below, we introduce the auxiliary graph used in the following heuristics.

3.4.1 Auxiliary Graph

To find link-disjoint paths from the DCs hosting the desired content to the requesting node, we introduce an auxiliary graph by leveraging a dummy node and several dummy links. To ensure that the final output is not affected by the addition of the dummy node and links, we assume that the dummy node has unlimited computing capacity and the dummy links have unlimited bandwidth and zero propagation delay.

As shown in Fig. 3.2, the dummy node (i.e., node 0) is connected to each DC hosting the desired content using the dummy links (denoted by dotted lines). The solid and dashed lines represent the network graph at request arrival time (i.e., $G_t(V_t, E_t)$) in which a user at node n is requesting for a content replicated in DCs d_0, d_1, d_2, and d_3 (i.e., $D = \{d_0, d_1, d_2, d_3\}$). Here, we use the dashed lines to abstract the real network with more nodes and links. Henceforth, we use $G_t^d(V_t^d, E_t^d)$ to denote the auxiliary network graph which includes $G_t(V_t, E_t)$, the dummy node, and dummy links.

One can observe that, to find the link-disjoint paths from the DCs hosting the desired content to the requesting node, we can find the link-disjoint paths from the dummy node to the requesting node on $G_t^d(V_t^d, E_t^d)$. Once the link-disjoint paths are found, the MPMD problem in the previous section can be restated as follows. *Given* the link-disjoint paths from the DCs hosting the desired content to the requesting node, *allocate* bandwidth to each path *such that*, for each content request, total requested bandwidth b during normal operation and degraded service $m * b$ in case a path is disrupted are fulfilled. The amount of bandwidth allocated to each path must not exceed the available/bottleneck bandwidth of the path, and differential delay of two distinct paths must satisfy DDC. In the specific scenario in Fig. 3.2, the numbers of link-disjoint paths (i.e., K), also the number of serving DCs, (i.e., $K = |D_0| = 3$), is fewer than the number of DCs hosting the desired content (i.e., $|D| = 4$). In this scenario, the algorithm selects the DCs in ascending order of their distances to the requesting node.

3.4.2 Equal-Bandwidth, Maximum-K MPMD (K-MPMD)

In this subsection, we design an algorithm that finds the maximum number of link-disjoint paths from the dummy node to the requesting node, and equally allocates bandwidth to each path such that the total requested bandwidth and degraded service are fulfilled. Hereafter, we refer to this service-provisioning scheme as the equal-bandwidth, maximum-K MPMD (K-MPMD).

In `Algorithm` 1, as inputs, $\theta = (t, n, c, b, m)$, $G_t(V_t, E_t)$, h, D, Λ_t, Ξ, and Ω denote the tuple representing a content request, network graph at request arrival time, desired content size (GB), set of DCs hosting the desired content, hash table representing the

Algorithm 1: Equal-BW, Max-K MPMD (K-MPMD).

```
    Input: $\theta = (t, n, c, b, m)$, $G_t(V_t, E_t)$, $h$, $D$, $\Lambda_t$, $\Xi$, $\Omega$
    Output: $K$, $D_0$, $P$, $b'$, $t'$
    Data: $K = 0$, $D_0 = \emptyset$, $P = \emptyset$, $b' = 0$, $t' = 0$, $\rho = 0$, $\xi_{min} = +\infty$, $\xi_{max} = 0$
1   construct $G_t^d(V_t^d, E_t^d)$
2   $K$, $G_t^K(V_t^K, E_t^K) \leftarrow$ find_paths$(0, n, G_t^d(V_t^d, E_t^d))$
3   if $K \geq 2$ then
4       $b'_p = \max(\text{ceil}(b/K), \text{ceil}(m * b/(K-1)))$
5       while $\rho < K$ do
6           $p_\rho \leftarrow$ Dijkstra$(0, n, G_t^{K-\rho}(V_t^{K-\rho}, E_t^{K-\rho})$; $b_\rho \leftarrow p_\rho$; $d_\rho \leftarrow p_\rho$; $\xi_\rho \leftarrow p_\rho$
7           if $\xi_\rho < \xi_{min}$ then $\xi_{min} = \xi_\rho$
8           if $\xi_\rho > \xi_{max}$ then $\xi_{max} = \xi_\rho$
9           if $b_\rho \geq b'_p$ then
10              add $d_\rho$ to $D_0$; add $p_\rho$ to $P$
11              for link $(i, j)$ in $p_\rho$ do
12                  remove $(i, j)$ from $G_t^{K-\rho}(V_t^{K-\rho}, E_t^{K-\rho})$
13          $\rho = \rho + 1$
14      if $\xi_{max}$ - $\xi_{min} \leq \Omega$ then
15          for path $p_\rho$ in $P$ do
16              for link $(i, j)$ in $p_\rho$ do
17                  $\Lambda_t[(i, j)] = \Lambda_t[(i, j)] - b'_p$
18          $b' = K * b'_p$; $t' = t + 8000 * h/b'$
19          return $K$, $D_0$, $P$, $b'$, $t'$
```

available capacity of each link in E_t at request arrival time, hash table representing the propagation delay of each link in E_t, and maximum differential delay between paths (μs), respectively. As outputs, `Algorithm` 1 finds K as the number of link-disjoint paths for the content request, D_0 as the set of optimal DCs to serve the request ($D_0 \subseteq D$, $|D_0| = K$), P as the list of K link-disjoint paths from each DC in D_0 to the requesting node, b' as the total offered bandwidth for the request, and t' as request departure time (s). Since the total offered bandwidth, b', is equally allocated to K link-disjoint paths, the bandwidth on each path is $b'_p = b'/K$.

`Algorithm` 1 starts by constructing the auxiliary graph with the dummy node and dummy links, $G_t^d(V_t^d, E_t^d)$ (line 1). In `find_paths`, we first set the capacity of each link in E_t^d to one bandwidth unit and use the Ford-Fulkerson's algorithm to find K as the maximum number of link-disjoint paths from the dummy node to requesting node [52,53]. `find_paths` also returns the flow graph, $G_t^K(V_t^K, E_t^K)$, where V_t^K and E_t^K are the actual nodes and links carrying the flow from the dummy node to the requesting node. If the maximum number of link-disjoint paths is larger than one (i.e., there are enough paths for multipath routing), the algorithm continues to find the minimum offered bandwidth on each path (i.e., b'_p, which is rounded up to the nearest integer). The algorithm computes the minimum offered bandwidth on each path (b'_p) such that the constraints on the total requested bandwidth (b) and degraded service ($m * b$) in case a path is disrupted (i.e., there remains K - 1 survivable paths) are fulfilled (line 4). From lines 5 to 13, `Algorithm` 1 performs a path-decomposition loop to find the actual link-disjoint paths from the dummy node to the requesting node, and equally allocate bandwidth to each path. Here, we use an augmented Dijkstra's algorithm to find the shortest path (p_ρ) from the dummy node to the requesting node on the flow graph (line 6) [54]. In addition to the actual path, the algorithm also finds the available bandwidth of the path (i.e., b_ρ, bottleneck bandwidth), optimal DC (i.e., d_ρ, from which the desired content is retrieved), and total propagation delay of the path (i.e., ξ_ρ). Lines 7 and 8 update the propagation delay of the shortest path (ξ_{min}) and longest path (ξ_{max}) in each iteration. From lines 9 to 12, the algorithm verifies that the available bandwidth of the path is enough for the requested bandwidth ($b_\rho \geq b'_p$), then adds the optimal DC d_ρ to set D_0, appends path p_ρ to list P, and removes the links along path p_ρ from $G_t^K(V_t^K, E_t^K)$. Note that a path traced out from the dummy node on the flow graph always terminates at the requesting node since $G_t^K(V_t^K, E_t^K)$ is an acyclic, directed graph, and the path-decomposition loop (lines 5-13) assuredly finds K link-disjoint paths [52,53]. However, the path-decomposition loop provides no backtracking, thus it offers no guarantee that the total propagation delay of all paths is minimal. In case the available bandwidth of the path is not enough for the request, the content request is not admitted and the algorithm terminates with no results.

Once the bandwidth on each path is sufficient for the request, from lines 14 to 19, the algorithm verifies that the differential propagation delay of the longest path and the shortest path fulfills DDC (i.e., ξ_{max} - $\xi_{min} \leq \Omega$); then it subtracts resources (i.e., capacity used on each link) from the network and returns K, D_0, P, b', and t'. Lastly, if DDC is not fulfilled or the number of paths from the dummy node to the requesting node is not enough for multipath routing (i.e., $K < 2$), the content request is not admitted and the algorithm terminates with no results.

3.4.3 Flexible MPMD (F-MPMD)

```
Algorithm 2: Flexible MPMD (F-MPMD).
   Input: $\theta = (t, n, c, b, m)$, $G_t(V_t, E_t)$, $h$, $D$, $\Lambda_t$, $\Xi$, $\Omega$
   Output: $K$, $D_0$, $P$, $b'$, $t'$, $S$
   Data: $K = 0$, $D_0 = \emptyset$, $P = \emptyset$, $b' = 0$, $t' = 0$, $\rho = 0$, $S = \text{False}$
1  construct $G_t^d(V_t^d, E_t^d)$
2  $K$, $G_t^K(V_t^K, E_t^K) \leftarrow \text{find\_paths}(0, n, G_t^d(V_t^d, E_t^d))$
3  if $K \geq 1$ then
4      while $\rho < K$ do
5          $p_\rho \leftarrow \text{Dijkstra}(0, n, G_t^{K-\rho}(V_t^{K-\rho}, E_t^{K-\rho}))$; add $p_\rho$ to $P$
6          for link $(i, j)$ in $p_\rho$ do
7              remove $(i, j)$ from $G_t^{K-\rho}(V_t^{K-\rho}, E_t^{K-\rho})$
8          $\rho = \rho + 1$
9      if $K = 1$ then
10         $p_0 = P[0]$; $b_0 \leftarrow p_0$; $d_0 \leftarrow p_0$
11         if $b_0 \geq b$ then
12             add $d_0$ to $D_0$; $b' = b$; $t' = t + 8000 * h/b'$
13             for link $(i, j)$ in $p_0$ do
14                 $\Lambda_t[(i, j)] = \Lambda_t[(i, j)] - b'$
15             return $K$, $D_0$, $P$, $b'$, $t'$, $S$
16     else
17         sort($P$, ascending=True, key=path_delay)
18         $p_0 = P[0]$, $p_{K-1} = P[K-1]$; $\xi_{min} \leftarrow p_0$; $\xi_{K-1} \leftarrow p_{K-1}$
19         while $K > 1$ and $\xi_{K-1} - \xi_{min} > \Omega$ do
20             remove the longest path in $P$; $K = K - 1$
21             $p_{K-1} = P[K-1]$; $\xi_{K-1} \leftarrow p_{K-1}$
22         if $K > 1$ then
23             sort($P$, descending=True, key=path_BW)
24             $p_{K-1} = P[K-1]$; $b_{K-1} \leftarrow p_{K-1}$
25             $b'_p = \max(\text{ceil}(b/K), \text{ceil}(m * b/(K-1)))$
26             while $K > 1$ and $b_{K-1} < b'_p$ do
27                 remove the least_BW path in $P$; $K = K - 1$
28                 $p_{K-1} = P[K-1]$; $b_{K-1} \leftarrow p_{K-1}$
29                 $b'_p = \max(\text{ceil}(b/K), \text{ceil}(m * b/(K-1)))$
30             if $K > 1$ then
31                 for path $p_\rho$ in $P$ do
32                     $p_\rho = P[\rho]$; $d_\rho \leftarrow p_\rho$; add $d_\rho$ to $D_0$
33                     for link $(i, j)$ in $p_\rho$ do
34                         $\Lambda_t[(i, j)] = \Lambda_t[(i, j)] - b'_p$
35                 $b' = K * b'_p$; $t' = t + 8000 * h/b'$; $S = \text{True}$
36                 return $K$, $D_0$, $P$, $b'$, $t'$, $S$
37             else
38                 go to line 10
39         else
40             go to line 10
```

Since K-MPMD exploits great path diversity by inversely multiplexing a content request over the maximum number of link-disjoint paths, it can relax the required bandwidth on each path. However, it may have potential shortcomings, e.g., K-MPMD rejects a content request if the available bandwidth on one path is not enough for the requested bandwidth. In another scenario, K-MPMD also rejects a content request if the propagation delay of one path does not fulfill DDC. This approach may not be optimal since the remaining paths (i.e., the paths whose bandwidth and propagation delay do not violate the two above-mentioned constraints) can still be used to admit the content request. In case a fewer number of paths is used, more bandwidth needs to be allocated to each path. Furthermore, if there exists only one path from the dummy node to the requesting node, the algorithm can still make its best effort to admit the request but provides no guarantee of survivability (i.e., no multipath routing).

For an illustration, let us consider a scenario where K-MPMD first finds four link-disjoint paths (e.g., p_0, p_1, p_2, and p_3) from four distinct DCs to the requesting node. To fulfill a content request with bandwidth b and $m = 0.75$, K-MPMD requires the bandwidth on each path to be at least $0.25b$, and the differential propagation delay of the longest path and the shortest path must fulfill DDC. In case of insufficient bandwidth on one path (e.g., p_3), the remaining paths (p_0, p_1, and p_2 whose propagation delays fulfill DDC) can be used to admit the content request with reserved bandwidth on each path of $0.375b$. If we further consider the scenario where the propagation delay of p_2 does not fulfill DDC, the content request can be admitted using two paths (p_0 and p_1) with reserved bandwidth on each path of $0.75b$. So, we derived an alternative approach based on these observations, called flexible heuristic for the MPMD problem (F-MPMD) as follows (F for flexible). Note that, in case of a multipath provisioning (i.e., $K \geq 2$), F-MPMD strictly enforces survivable bandwidth and DDC.

As shown in `Algorithm` 2, F-MPMD shares the inputs, outputs, and most steps with K-MPMD. However, F-MPMD also provides S as the output to indicate whether or not a service provisioning is survivable (i.e., survivable, S = True). In contrast to `Algorithm` 1 (where the algorithm terminates if the number of paths is not enough for multipath provisioning, i.e, $K < 2$), `Algorithm` 2 continues even if there is only one path from the dummy node to the requesting node (line 3). The code block between lines 4 and 8 computes the actual path(s) between the two nodes.

If there exists only one path from the dummy node to the requesting node and bandwidth on this path is enough for the requested bandwidth ($b_0 \geq b$), the content request is admitted, but provisioning is not survivable (S = False, default value). `Algorithm` 2 updates the network and returns results (lines 9-15). In case the bandwidth on this single path from the dummy node to the requesting nodes is not enough for the requested bandwidth ($b_0 < b$), the content request is not admitted and the algorithm terminates with no results.

In case there are two or more link-disjoint paths from the dummy node to the requesting node (line 16), `Algorithm` 2 sorts the paths in P by their total propagation delay in ascending order, sets the shortest path as the reference, and removes the paths whose propagation delay violates DDC (lines 17-21). In case the number of paths is more than one after removing the paths whose propagation delay violates DDC (line 22), `Algorithm` 2 continues to sort the paths in P by their available bandwidths in descending order and removes the paths whose bandwidths are not enough to fulfill the requested bandwidth (lines 23-29). If the number of paths in P is enough for multipath provisioning after both operations (i.e., removals of paths whose propagation delay and bandwidth do not fulfill the two-mentioned constraints, line 30), the content request is survivably provisioned (i.e., S = True). `Algorithm` 2 subtracts resources (i.e., capacity used on each link) from the network and returns K, D_0, P, b', t', and S (lines 31-36). In each step, if the number of paths remains one, `Algorithm` 2 returns to the scenario where there exists only one path from the dummy node to the requesting node (i.e., returns to line 10, such as on lines 38 and 40).

In summary, `Algorithm` 2 is flexible in every step to provision a content request. The algorithm keeps removing invalid paths (i.e., paths whose propagation delay and bandwidth are insufficient for the request) until it can find the optimal K to provision a content request.

The heuristic formulations for F-MPSD (i.e., flexible, MPSD) and DPP can be directly obtained from the F-MPMD formulations if, instead of the dummy node, the closest DC to the requesting node is used in `Algorithm` 2.

3.4.4 Complexity Analysis

In `Algorithm` 1, since the number of link-disjoint paths (i.e., K) is deterministic (e.g., number of link-disjoint paths must not exceed the nodal degree of the requesting node), the number of iterations inside the `while` loop (lines 5-16) is also deterministic. Moreover, as the number of links on each path is deterministic, the number of iterations inside the `for` loops (lines 12-13 and 19-22) is also deterministic. Note that the construction of a graph from another graph requires linear time complexity (i.e., $\mathcal{O}(|V_t| + |E_t|)$), which is a non-dominant term. `Algorithm` 1's time complexity heavily depends on the time complexity of the method to find the maximum flow (line 2) and the method to find the shortest path (line 6) from the dummy node to the requesting node. In this study, we use the Ford-Fulkerson's algorithm to find the maximum flow from the dummy node to the requesting node whose time complexity is $\mathcal{O}(|E_t^d|)$ (line 2). Moreover, we use the augmented Dijkstra's algorithm and a min-heap data structure to find the shortest path from the dummy node to the requesting node whose time complexity is $\mathcal{O}\left((|V_t^K| + |E_t^K|) * \log(|V_t^K|)\right)$. If we omit non-dominant terms and deterministic constants, the time complexity of `Algorithm` 1 is $\mathcal{O}\left(|E_t^d| + (|V_t^K| + |E_t^K|) * \log(|V_t^K|)\right)$. Similarly, the time complexity of `Algorithm` 2 is $\mathcal{O}\left(|E_t^d| + (|V_t^K| + |E_t^K|) * \log(|V_t^K|)\right)$ if we omit deterministic constants and non-dominant terms such as sorting of the paths in P (lines 20, 27).

3.4.5 Service Probability

In the previous subsections, we designed the algorithms for the MPMD problem against the dominant failure scenario, namely a single-link failure. The algorithms guarantee degraded service against a random single-link failure in an optical metro network. In this subsection, we compare the reliability of F-MPMD, F-MPSD, and DPP from a service probability perspective. We consider five typical scenarios, including a single-link failure (L_1), a double-link failure (L_2), a single-DC failure (D_1), a double-DC failure (D_2), a single-link plus a single-DC failure ($L_1 + D_1$), and address the fundamental question: if a content request is already survivably provisioned (i.e., $K \geq 2$) and any of the failure scenarios occurs, what is the probability of fulfilling the requested degraded service? We define *service probability*, or Π, as the probability that the requested degraded service is guaranteed against a specific failure scenario. To simplify calculations without loss of generality, in this subsection, we assume that a link or a DC in an optical metro network is failed with equal probability.

Since F-MPMD, F-MPSD, and DPP are designed to guarantee degraded service against a single-link failure, their service probabilities in this scenario are 100%, i.e.,

$$\Pi^{L_1}_{\text{F-MPMD}} = \Pi^{L_1}_{\text{F-MPSD}} = \Pi^{L_1}_{\text{DPP}} = 100\%.$$

Moreover, in contrast to F-MPSD and DPP, F-MPMD also provides protection against a single-DC failure, hence, $\Pi^{D_1}_{\text{F-MPMD}} = 100\%$. For other failure scenarios, we use E_θ to denote the set of physical links used by the content request θ, E^ρ_θ to denote the sets of physical link(s) on each path, $\rho = \{0..K\text{ - }1\}$, and derive the formulas for service probability as follows.

$$\Pi^{D_1}_{\text{F-MPSD}} = \Pi^{D_1}_{\text{DPP}} = 1 \text{ - } \frac{1}{|D|}. \tag{3.11}$$

$$\Pi^{L_2}_{\text{F-MPMD}} = \Pi^{L_2}_{\text{F-MPSD}} = 1 \text{ - } \frac{\binom{|E_\theta|}{2} \text{ - } \sum_{\rho=0:|E^\rho_\theta| \geq 2}^{K\text{ - }1} \binom{|E^\rho_\theta|}{2}}{\binom{|E_t|}{2}}. \tag{3.12}$$

$$\Pi^{L_2}_{\text{DPP}} = 1 \text{ - } \frac{|E^0_\theta| * |E^1_\theta|}{\binom{|E_t|}{2}}. \tag{3.13}$$

$$\Pi^{D_2}_{\text{F-MPMD}} = 1 \text{ - } \frac{\binom{|D_0|}{2}}{\binom{|D|}{2}}. \tag{3.14}$$

$$\Pi^{D_2}_{\text{F-MPSD}} = \Pi^{D_2}_{\text{DPP}} = 1 \text{ - } \frac{|D| - 1}{\binom{|D|}{2}}. \tag{3.15}$$

$$\Pi^{L_1 + D_1}_{\text{F-MPMD}} = 1 - \frac{(|D_0| - 1) * |E_\theta|}{|D| * |E_t|}. \tag{3.16}$$

$$\Pi^{L_1 + D_1}_{\text{F-MPSD}} = \Pi^{L_1 + D_1}_{\text{DPP}} = 1 - \frac{|E_\theta|}{|D| * |E_t|}. \tag{3.17}$$

Here, for example, for D_2 (i.e., a double-DC failure), we compute the number of combinations (without competition) which disrupt the requested degraded service (e.g., the numerator of the second term in (3.14) which is $\binom{|D_0|}{2}$ where D_0 is the set of serving DCs), the total number of combinations (e.g., the denominator of the second term in (3.14) which is $\binom{|D|}{2}$ where D is the set of content-hosting DCs), and the service probability against this failure scenario, $\Pi^{D_2}_{\text{F-MPMD}} = 1 - \binom{|D_0|}{2} / \binom{|D|}{2}$. In equations (3.11) to (3.17), we derive the formulas to compute the service probability for each failure scenario and each protection scheme. We will use the above equations to show that, among the three schemes, F-MPMD has better service probability in most failure scenarios.

3.5 Illustrative Numerical Results

3.5.1 Physical Network and Simulation Setup

3.5.1.1 Physical Network

To evaluate our proposed solutions, we use the Tokyo23 metro network covering an urban area up to tens of kilometers in diameter as in Fig. 2.5 [27]. The Tokyo23 metro network has been designed using regional characteristics such as population distribution, locations of local government offices, and railway lines with the number of passengers getting on/off each station. It consists of 43 bidirectional links (i.e., 86 100-Gbps, unidirectional links) and 23 nodes, with each node located at each ward office building in the Tokyo metropolitan area. We also validated our proposed algorithms on the Milan52 metro network (as in Fig. 2.4) which represents a Telecom Italia metro-regional reference network with 52 nodes and 101 bidirectional links [26, 55]. Since the results obtained with the Milan52 metro network are in line with the ones obtained with Tokyo23 metro network, below, we only reported the results obtained with Tokyo23 metro network to not unnecessarily repeat the same general findings.

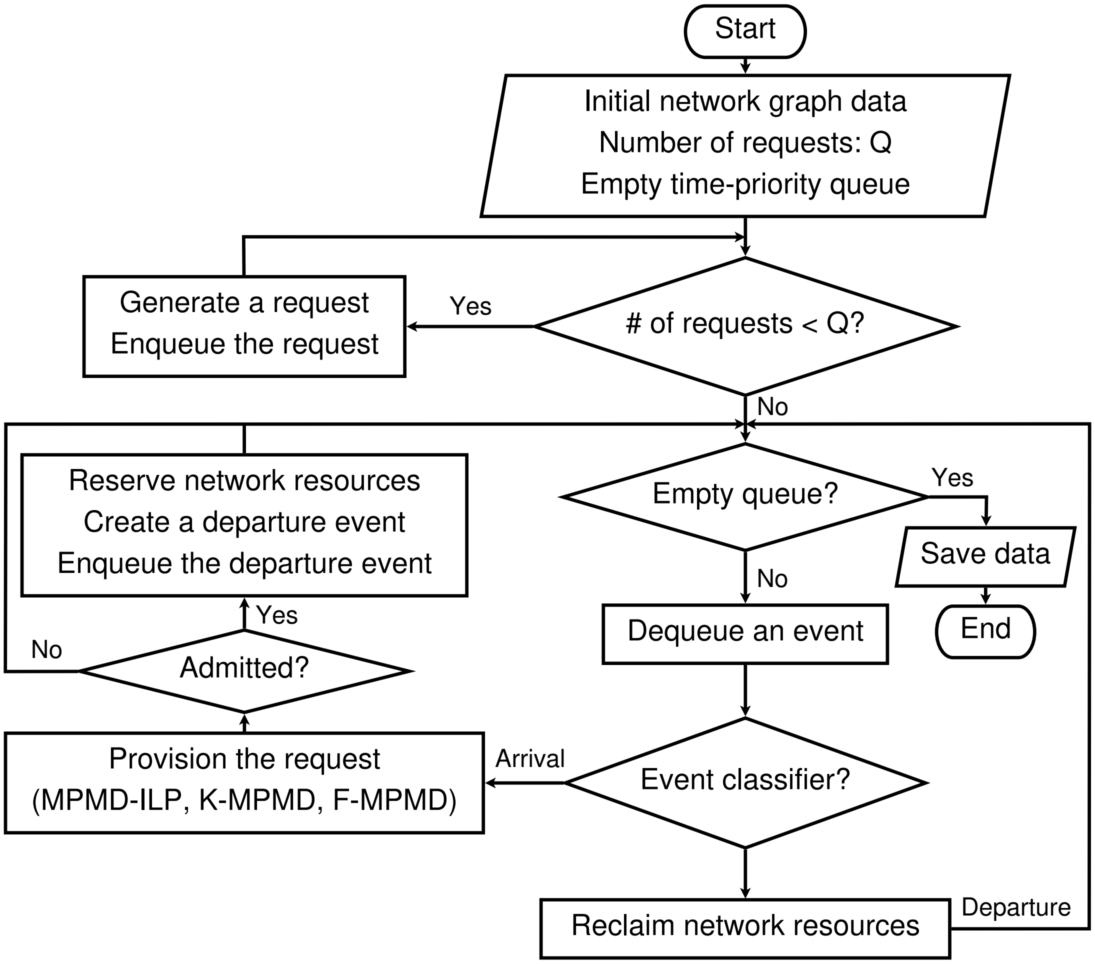


Figure 3.3: Simulation framework.

3.5.1.2 Experiment Setup

For our experiment setup, we use the simulation framework in Fig. 3.3 to simulate the network with 105000, 205000, 305000, 405000, and 505000 content requests whose arrivals follow a discrete Poisson process. We first generate all content requests and enqueue them in a time-priority queue and start with an empty network (i.e., a network with no active traffic). During simulation, each content request is dequeued from the queue and provisioned using one of the proposed algorithms (MPMD-ILP, K-MPMD, or F-MPMD). If a content request is admitted, required network resources are reserved for it, and a departure event with departure time equal to the content request arrival time plus the holding time is enqueued to the queue. If the event is a departure, reserved network resources are released. Note that the simulator processes one event at a time (either an arrival or a departure), and it proceeds with the next event in queue as soon as it finishes the current one. We also found that the network requires approx. 5000 content requests to reach a steady state, and numerical results obtained for simulating 105000 and 505000 content requests are comparable. To reduce experiment time, below we report the numerical results for simulating 105000 content requests (i.e., first 5000 content requests are discarded and the acceptance ratio is the ratio of the number of admitted requests over the number of simulated requests (i.e., 100000 in our simulations).

In a dynamic network environment, the acceptance ratio of incoming requests as a function of the arrival rates is crucial. In an ideal network with abundant resources (e.g., high-capacity links, super-fast computing capability in DCs/nodes, and contents replicated in multiple locations), the acceptance ratio, or η, is 100%. However, in practice, network resources can be scarce, and when content requests arrive at a very high rate, the network can become congested, and several incoming requests may be dropped (i.e., $\eta < 100\%$). In this study, we define the *congestion point* in a dynamic network as the arrival rate at which the network starts to drop several incoming requests, and use η_0 to denote this value. To run a network harder, with the same input setting, a service-provisioning scheme with a higher η_0 is desirable. In a congested network, a service-provisioning scheme with a higher η is preferred since it can admit more requests. In the next sections, we will use η_0 and η to evaluate our proposed service-provisioning schemes (MPMD-ILP, K-MPMD, and F-MPMD), and compare their performance to those of reference schemes (F-MPSD, DPP). Also, in Sections 3.5.2, 3.5.3, and 3.5.4.1, we consider content requests which are survivably provisioned (i.e., $K \geq 2$).

3.5.2 MPMD-ILP vs. K-MPMD vs. F-MPMD

Let us first compare the performance of MPMD-ILP, K-MPMD, and F-MPMD. We consider a service provider with a content catalog of 10000 contents, whose size, h, ranges from 5 GB (e.g., a medium HD video) to 1000 GB (e.g., a long VR/AR video). We assume that, for each content request, the desired content is replicated in multiple DCs, i.e.,

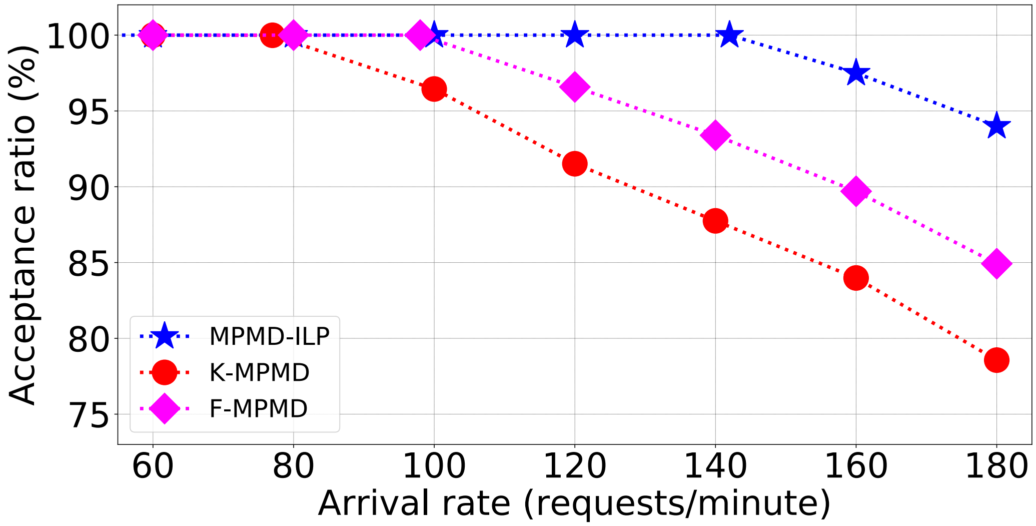


Figure 3.4: Acceptance ratio of MPMD-ILP, K-MPMD, and F-MPMD as a function of request arrival rates.

$D = \{1, 7, 13, 16\}$. Note that contents in DCs are dynamic, stateful, and require frequent updates (e.g., a content is replicated and synchronized in an edge DC by its popularity following a Zipf distribution [32]). However, the synchronization of contents among DCs is not considered in this study. The requested bandwidth, b, is uniformly selected from discrete values, ranging from 200 Mbps (e.g., a stream for a VR head-mounted display worn by the user) to 2000 Mbps (e.g., an uncompressed VR/AR flow). The required level of degraded service for each request, m, is randomly selected from discrete values, $m \in \{0.5, 0.7, 1.0\}$, where $m = 1.0$ denotes full protection in case a path is disrupted. The requesting node, n, is selected from the nodes not hosting the desired content following the population distribution. For DDC, we consider the propagation delay on fibers as the major delay in an optical network whose practical value is set to 2 ms (e.g., 6DoF VR immersive experience use case) [32, 36, 42].

In Fig. 3.4, we report the acceptance ratio of MPMD-ILP, K-MPMD, and F-MPMD at different request arrival rates. As expected, MPMD-ILP outperforms K-MPMD and F-MPMD. In fact, while K-MPMD and F-MPMD start dropping incoming requests around the arrival rates of 77 and 98 requests/minute, respectively, MPMD-ILP can run the same network harder and starts dropping incoming requests around the arrival rate of 136 requests/minute. In a congested network (e.g., where arrival rate is larger than

Table 3.1: MPMD-ILP vs. K-MPMD vs. F-MPMD

R	Schemes	K_{avg}	b'_{avg}	b''_{avg}	ξ_{avg}
60	MPMD-ILP	2.5	1411.6	2467.6	45.8
	K-MPMD	3.5	1272.1	2896.7	57.0
	F-MPMD	3.5	1269.6	2887.4	56.9
80	MPMD-ILP	2.5	1354.6	2471.6	46.2
	K-MPMD	3.5	1268.4	2899.1	57.2
	F-MPMD	3.5	1269.2	2891.2	57.0
100	MPMD-ILP	2.6	1327.7	2429.1	45.3
	K-MPMD	3.5	1258.3	2837.3	56.6
	F-MPMD	3.4	1275.7	2891.9	56.9
120	MPMD-ILP	2.5	1313.9	2398.8	47.8
	K-MPMD	3.5	1230.1	2722.7	56.1
	F-MPMD	3.3	1277.9	2842.6	56.4
140	MPMD-ILP	2.5	1302.1	2357.4	46.1
	K-MPMD	3.5	1219.5	2662.0	55.6
	F-MPMD	3.2	1272.5	2766.5	55.7
160	MPMD-ILP	2.5	1290.4	2308.4	48.3
	K-MPMD	3.5	1200.4	2606.3	55.0
	F-MPMD	3.2	1265.5	2697.1	55.1
180	MPMD-ILP	2.5	1286.9	2283.9	48.4
	K-MPMD	3.5	1172.4	2674.9	55.1
	F-MPMD	3.1	1251.0	2628.9	54.5

Data points in Table 3.1 are obtained by averaging over all accepted requests. Here, R, K_{avg}, b'_{avg}, b''_{avg}, and ξ_{avg} are request arrival rate (requests/minute), average number of paths per request (paths/request), average offered bandwidth (Mbps/request), average network bandwidth (Mbps/request), and average path propagation delay (μs/path), respectively.

136 requests/minute), compared to K-MPMD and F-MPMD, MPMD-ILP approximately accepts 5% and 12%, respectively, more incoming requests. Even though MPMD-ILP outperforms K-MPMD and F-MPMD, in a dynamic environment, F-MPMD, which outperforms K-MPMD, is more desirable since it can find a fast solution.

In Table 3.1, we report other relevant data for MPMD-ILP, K-MPMD, and F-MPMD. On average, the number of paths per request (i.e., K_{avg}) decreases when moving from K-MPMD (∼3.5 paths/request), to F-MPMD (∼3.3 paths/request), and to MPMD-ILP

Table 3.2: Average execution time of the algorithms.

MPMD-ILP	K-MPMD	F-MPMD
220.69	0.89	1.21

Data points in Table 3.2 are the average execution time per arrival request (ms) where MPMD-ILP, K-MPMD, and F-MPMD were run on Tokyo23 metro network using a mobile workstation with Intel(R) Xeon E3-1505M 2.8 GHz CPU and 64 GB of RAM.

(∼2.6 paths/request) because K-MPMD always finds the maximum number of paths from the dummy node to the requesting node and equally allocates requested bandwidth to each path. On the contrary, MPMD-ILP finds just enough number of paths and allocates more bandwidth to each path to fulfill a request. Also, since F-MPMD first finds the maximum number of paths from the dummy node to the requesting node and then drops invalid paths (i.e., whose bandwidths or propagation delay is insufficient for the request), the average number of paths per request for F-MPMD is less than the average number of paths per request for K-MPMD. On average, the number of paths per request is increased 32% and 40% from MPMD-ILP to F-MPMD and K-MPMD, respectively.

Another notable observation is that, since the path-decomposition loops in `Algorithms` 1 and 2 provide no guarantee to find the paths from each DC in D_0 to the requesting node whose total propagation delay is minimal, MPMD-ILP's average path propagation delay is less than K-MPMD and F-MPMD's average path propagation delay. On average, the path propagation delay (i.e., ξ_{avg}) of MPMD-ILP is about 10 μs less (or 18% less) than the path propagation delay of K-MPMD and F-MPMD. As a result, even though MPMD-ILP offers more bandwidth per request (i.e., b'_{avg}, on average, 11.5% more bandwidth), it uses less network bandwidth (i.e., b''_{avg}). On average, per request, compared to K-MPMD and F-MPMD, MPMD-ILP uses about 17.4% less network bandwidth.

In Table 3.2, we reported the average execution time per arrival request (ms) for MPMD-ILP, K-MPMD, and F-MPMD. Our heuristics can obtain a solution in a timely manner, considering also that a few milliseconds for computing a solution and buffering data is initially reserved for each content request. The ILP, instead, is not suitable for a dynamic content request but provides a reliable benchmark for the heuristics.

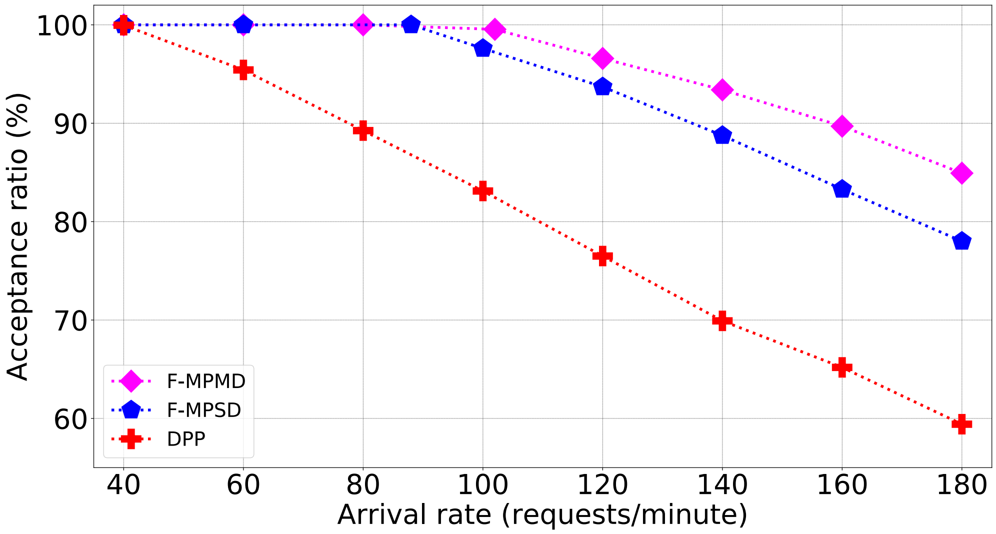


Figure 3.5: Acceptance ratio of F-MPMD, F-MPSD, and DPP as a function of request arrival rates.

3.5.3 F-MPMD vs. F-MPSD vs. DPP

In this subsection, we compare MPMD (using F-MPMD) with two reference protection schemes, namely F-MPSD and DPP. We use the same simulation setting as in Section 3.5.2, and since the congestion point of DPP is lower than the congestion point of F-MPMD and F-MPSD, we also report numerical data for the arrival rate as low as 40 requests/minute.

3.5.3.1 Acceptance Ratio and Latency

In Fig. 3.5, we report the acceptance ratio of F-MPMD, F-MPSD, and DPP at different request arrival rates. Numerical data show that F-MPMD outperforms F-MPSD and DPP as it can run the network harder with a higher congestion point. In detail, the congestion points of F-MPMD, F-MPSD, and DPP are 102, 88, and 40 requests/minute, respectively. In a congested network, compared to F-MPSD and DPP, respectively, F-MPMD accepts approximately 4% and 15% more incoming requests.

In Table 3.3, we report the relevant data for F-MPMD, F-MPSD, and DPP. Since both F-MPMD and F-MPSD rely on multipath routing, they use significantly less bandwidth. Compared to DPP, on average, F-MPMD uses about 39.5% less bandwidth per request

Table 3.3: F-MPMD vs. F-MPSD vs. DPP

R	Schemes	K_{avg}	b'_{avg}	b''_{avg}	ξ_{avg}
40	F-MPMD	3.5	1270.8	2893.2	57.0
	F-MPSD	3.5	1269.1	3831.3	75.2
	DPP	2.0	1895.0	3219.6	44.6
60	F-MPMD	3.5	1269.6	2887.4	56.9
	F-MPSD	3.5	1274.1	3851.7	75.5
	DPP	2.0	1872.0	3176.1	44.5
80	F-MPMD	3.4	1269.2	2891.2	57.0
	F-MPSD	3.4	1280.9	3854.6	75.2
	DPP	2.0	1853.0	3123.4	44.2
100	F-MPMD	3.4	1275.7	2891.9	56.9
	F-MPSD	3.4	1268.3	3819.1	75.3
	DPP	2.0	1808.6	3043.8	44.3
120	F-MPMD	3.3	1277.9	2842.6	56.4
	F-MPSD	3.3	1243.0	3729.8	75.1
	DPP	2.0	1763.1	2971.3	44.7
140	F-MPMD	3.3	1272.5	2766.5	55.7
	F-MPSD	3.3	1219.7	3648.0	74.8
	DPP	2.0	1716.7	2899.6	45.1
160	F-MPMD	3.3	1265.5	2697.1	55.1
	F-MPSD	3.3	1210.9	3598.4	74.7
	DPP	2.0	1683.6	2839.7	45.1
180	F-MPMD	3.2	1251.0	2628.9	54.5
	F-MPSD	3.2	1169.4	3448.3	74.3
	DPP	2.0	1631.8	2767.5	45.4

Data points in Table 3.3 are obtained by averaging over all accepted requests. Here, R, K_{avg}, b'_{avg}, b''_{avg}, and ξ_{avg} are request arrival rate (requests/minute), average number of paths (paths/request), average offered bandwidth (Mbps/request), average network bandwidth (Mbps/request), and average path propagation delay (μs/path), respectively.

(i.e., b'_{avg}). As another observation, among the three service-provisioning schemes, F-MPMD uses least network bandwidth per request (i.e., b''_{avg}), and compared to F-MPSD, it can save up to 30%. Moreover, on average, the path propagation delay of F-MPMD (i.e., ξ_{avg}) is approximately 20 μs less than the path propagation delay of F-MPSD, making F-MPMD more suitable for emerging services which have stringent latency constraint.

Table 3.4: Average Service Probability in Different Failure Scenarios.

Scenario	F-MPMD	F-MPSD	DPP
L_1	100%	100%	100%
D_1	100%	71.3%	71.3%
L_2	99.4%	98.9%	99.9%
D_2	60.1%	54.2%	54.2%
$L_1 + D_1$	94.7%	96.5%	98.6%

Here, L_1, D_1, L_2, D_2, and $L_1 + D_1$ are for a single-link, a single-DC, a double-link, a double-DC, and one link plus one DC failure scenarios, respectively.

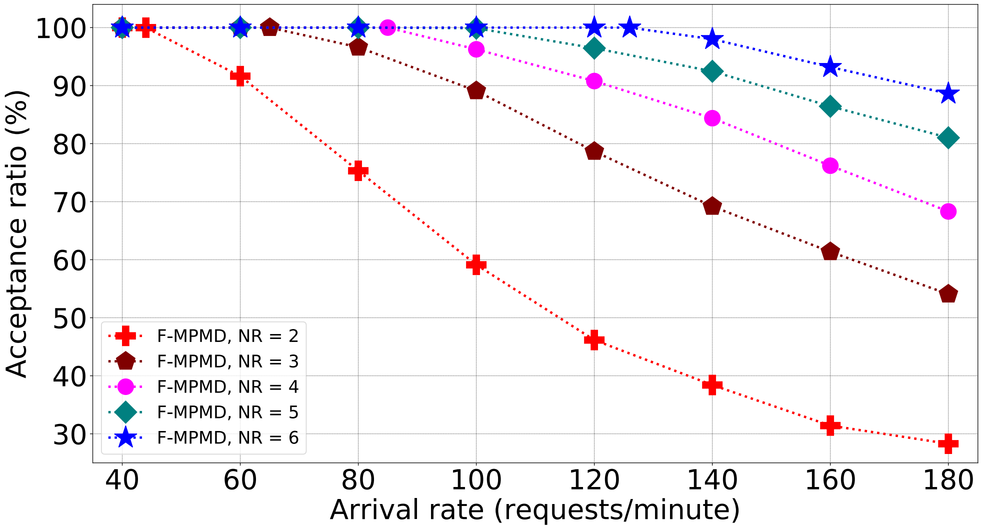


Figure 3.6: Acceptance ratio of F-MPMD as a function of number of content replicas (NR) at different request arrival rates.

Lastly, since DPP uses only two paths (i.e., the first and second shortest paths from the closest DC to the requesting node), the average propagation delay per path used by DPP is least among the three service-provisioning schemes (F-MPMD, F-MPSD, and DPP).

3.5.3.2 Service Probability

We now compare the service probability of F-MPMD to the service probability of F-MPSD and DPP. Here, we set the request arrival rate to 40 requests/minute (i.e., $\eta = 100\%$ for F-MPMD, F-MPSD, and DPP as in Fig. 3.5), and compute the average service probability per request for each failure scenario using the formulas in Section 3.4.5. We report the results in Table 3.4.

As expected, F-MPMD, F-MPSD, and DPP all guarantee degraded service against a random single-link failure on the physical network (i.e., $\Pi^{L_1}_{\text{F-MPMD}} = \Pi^{L_1}_{\text{F-MPSD}} = \Pi^{L_1}_{\text{DPP}} = 100\%$). Moreover, since F-MPMD offers protection against a single-DC failure, its service probability is 100% in this scenario (i.e., $\Pi^{D_1}_{\text{F-MPMD}} = 100\%$) while F-MPSD and DPP provide no service guarantee (i.e., $\Pi^{D_1}_{\text{F-MPSD}} = \Pi^{D_1}_{\text{DPP}} = 71.3\%$). We also see that, since DPP uses only two paths, for each request, the number of physical links used by DPP is fewer than the number of physical links used by F-MPMD and F-MPSD. In other words, the number the double-link failure combinations disrupting DPP is lower than the number the double-link failure combinations disrupting F-MPMD and F-MPSD. As a result, among the three protection schemes, DPP has the highest service probability against a double-link failure (i.e., $\Pi^{L_2}_{\text{DPP}} = 99.9\%$). Similarly, compared to F-MPSD, F-MPMD tends to use shorter paths (refer to ξ_{avg} in Table 3.3); hence, it has higher service probability against a double-link failure ($\Pi^{L_2}_{\text{F-MPMD}} = 99.4\%$, compared to $\Pi^{L_2}_{\text{F-MPSD}} = 98.9\%$). In the scenario where there are two simultaneous failures in two distinct DCs, F-MPMD outperforms F-MPSD and DPP in terms of service probability ($\Pi^{D_2}_{\text{F-MPMD}} = 60.1\%$ compared to $\Pi^{D_2}_{\text{F-MPSD}} = \Pi^{D_2}_{\text{DPP}} = 54.2\%$) because F-MPMD uses multiple DCs while F-MPSD and DPP use only one DC for each request. For the last scenario where one link plus one DC fail simultaneously (i.e., $L_1 + D_1$), with the same total number of links (i.e., $|E_\theta|$ in Eqns. (3.16) and (3.17)), the number of combinations disrupting F-MPMD is $|D_0| - 1$ times the number of combinations disrupting F-MPSD and DPP. Hence, compared to F-MPSD and DPP, F-MPMD has lowest service probability ($\Pi^{L_1+D_1}_{\text{F-MPMD}} = 94.7\%$, $\Pi^{L_1+D_1}_{\text{F-MPSD}} = 96.5\%$, and $\Pi^{L_1+D_1}_{\text{DPP}} = 98.6\%$).

3.5.4 F-MPMD

3.5.4.1 Number of Content Replicas (NR)

In this subsection, we use the same simulation setting as in Section 3.5.3 and obtain numerical results for F-MPMD for increasing number of content replicas (NR) in the network from two to six (i.e., $NR = 2$, $D = \{1, 4\}$; $NR = 3$, $D = \{1, 4, 7\}$; $NR = 4$, $D = \{1, 4, 7, 10\}$;

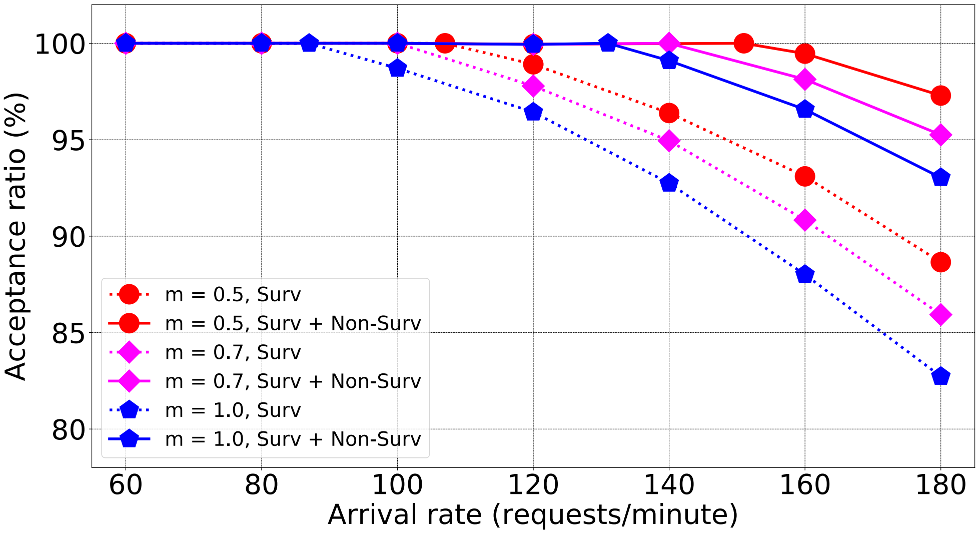


Figure 3.7: Acceptance ratio of F-MPMD as a function of request arrival rates at three different levels of degraded service.

$NR = 5$, $D = \{1, 4, 7, 10, 13\}$; and $NR = 6$, $D = \{1, 4, 7, 10, 13, 16\}$). As shown in Fig. 3.6, increasing the number of content replicas from $NR = 2$ to $NR = 3$ significantly improves the acceptance ratio of incoming requests. In detail, the congestion point is increased from 44 to 65 requests/minute; and in a congested network, with $NR = 3$, compared to $NR = 2$, F-MPMD accepts around 25% more incoming requests, while acceptance ratio improves more slowly when content replicas are further increased to 4, 5, and 6. In general, adding more content replicas increases the acceptance ratio, but it also implies more synchronization overhead. For this trade-off, only a content provider knows which option is cost-optimal. Therefore, the decision on the number of content replicas in a specific network may vary from content provider to content provider.

3.5.4.2 Survivability (Surv) vs. Non-Survivability (Non-Surv)

To this end, we show how many more content requests F-MPMD can admit even though their survivability is not guaranteed. We report the results for three distinct levels of degraded service (i.e., $m = 0.5,\ 0.7,$ and 1.0).

As shown in Fig. 3.7, the dotted lines denote acceptance rates where admitted requests are survivable in case a path is disrupted (i.e., $S = \text{True}$ in `Algorithm` 2) while the solid lines represent acceptance rates where admitted requests can be non-survivable.

Considering the dotted lines, switching the levels of degraded service from 1.0 (i.e., full protection), down to 0.7, and to 0.5 increases the congestion points from 87, to 100, and to 107 (requests/minute), respectively. Furthermore, the solid lines show how many more content requests F-MPMD can admit even though it provides no guarantee of survivability against a single-link failure. By admitting several requests without ensuring survivability (solid lines), F-MPMD can run the same network much harder and only starts dropping incoming requests at the arrival rates of 131, 140, and 152 (requests/minute), for m = 0.5, m = 0.7, and m = 1.0, respectively. In a highly-congested network (e.g., arrival rate 180 (requests/minute)), compared to the scheme where B-MPMD ensures survivability against a single-link failure, B-MPMD can admit approx. 10% more incoming requests.

3.6 Conclusion

We proposed a reliable service-provisioning scheme that inversely multiplexes a dynamic content request over multiple link-disjoint paths from multiple data centers using manycasting. We developed an integer linear program and two scalable heuristics for the proposed scheme and used them to solve various complex network instances. Numerical data show that, compared to conventional service-provisioning schemes such as multipath routing from a single DC and dedicated-path protection, our proposed service-provisioning scheme efficiently utilizes network resources, admits more requests, improves reliability, reduces latency; hence, it is very suitable for emerging content-based services. In this chapter, we considered the scenario where contents are cacheable and replicated in multiple data centers. In the following chapter, we will consider the scenario where contents are not cacheable (e.g., live streams) and services require high bandwidth, ultra-low latency, and reliable connections.

Chapter 4

Reliable Provisioning of Low-latency and High-bandwidth Extended Reality Live Streams

The networking industry is offering new services as a result of the advanced technologies in connectivity, storage, and computing such as mobile communications (e.g., 5G/6G), cloud, and edge computing. In this regard, extended reality, a term encompassing virtual reality (VR), augmented reality(AR), and mixed reality (MR), can provide unprecedented experiences and possibilities such as live concerts, sports, and other events; interactive gaming and entertainment; immersive education, training, and demos. These services require high bandwidth, low latency, reliable connections, and are classified as next-generation ultra-reliable and low-latency communications in the vision of 6G mobile communication systems.

In this chapter, we address the reliable provisioning of low-latency and high-bandwidth extended reality live streams in next-generation networks. We consider the scenario where contents are not cacheable and investigate the backup from different data centers with multicast and flexible offered bandwidth to fulfill extended reality live stream requests. We propose a service-provisioning scheme to protect not only against failures of links in the network but also against failures of computing and storage in data centers. We develop scalable algorithms for the backup from different data centers with multicast and flexible offered bandwidth, and use them to solve various complex network instances in a dynamic network environment. Numerical data show that, compared to a conventional service-provisioning scheme such as backup from the same data center, our proposed service-provisioning scheme provides higher reliability, reduces latency, and efficiently utilizes network resources; hence, it is highly suitable for extended reality live streams.

4.1 Introduction

With the adoption of advanced technologies in connectivity, storage, and computing such as mobile communications (e.g., 5G/6G), cloud, and edge computing, the networking industry is introducing new services which are not possible before. In this regard, VR, AR, and MR have emerged as the first wave of killer applications [56]. VR, AR, and MR allow users to interact intuitively with the environment through six degrees of freedom (6DoF) and are typically referred to as *extended reality* (ER), a term encompassing all three technologies [57]. ER is offering unprecedented experiences and possibilities such as live concerts, sports, and other events; interactive gaming and entertainment; immersive education, training, and demos, just to name a few. These services require high-bandwidth, low-latency, reliable connections, and are classified as next-generation ultra-reliable and low-latency communications (XURLLC) in the vision of 6G mobile communication systems [58]. To meet these stringent requirements (especially on latency which should be less then a few milliseconds [56]), a service-provisioning scheme must embrace the opportunity created by the advancements of network technologies.

This work addresses the *reliable provisioning of low-latency and high-bandwidth extended reality live streams* in next-generation networks. To illustrate our proposed service-provisioning scheme, in Fig. 4.1, we consider a dynamic scenario where, at time t, node 1 is requesting for an ER live stream originally available in node 15. Here, we assume that the requesting node (e.g., node 1) is a CO from which data are ultimately delivered to end users via either a wireless (e.g. 5G/6G or Wi-Fi) or a fixed network. Moreover, node 15 is a remote DC, close to the live event, which processes multiple video streams from different shooting angles and produces the ER live stream. The location of the live event

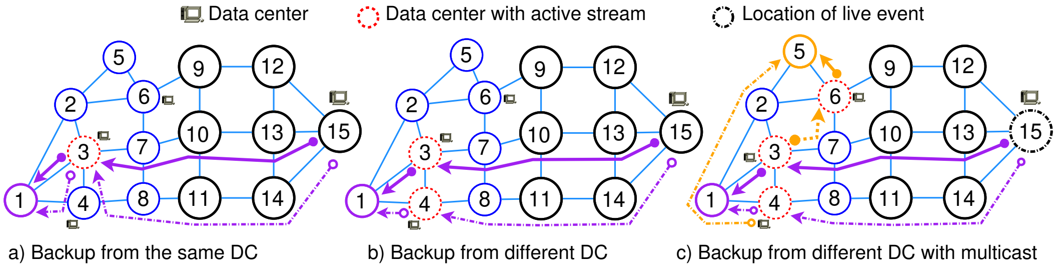

Data center
Data center with active stream
Location of live event
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
a) Backup from the same DC
b) Backup from different DC
c) Backup from different DC with multicast

Figure 4.1: Service-provisioning schemes of extended reality live streams.

can be very far away from the requesting node and causes significant delay (hundreds of milliseconds or even seconds) and a consequent shift in time. The ER live stream must be transmitted in quasi real time on ten-Gbps flows over a CDN and made available in edge DCs to allow end users to enjoy an immersive experience. It is worth noting that, since immersive experience is interactive (e.g., when an end user moves his/her head, the brain expects an instantaneous visual and aural update), low latency between the requesting node and edge DCs is very important. Ideally, the requesting node should be geographically served by the closest DC. In a CDN, requests from the same region are first served by a local DC cluster (i.e., a set of DCs which are geographical located in a local region such as a metro network to serve local requests) [59]. Also, since this work considers ER live streams, contents are not cacheable (i.e., an ER live stream and its reserved network resources including storage memory are released from edge DCs if there are no active connections) [60]. In Fig. 4.1, we assume that nodes 1-8 belong to a metro network and are served by nearby DCs in nodes 3, 4, and 6 (i.e., a DC cluster); nodes 9-15 belong to a backbone network [61].

Since ER live streams require high-bandwidth and reliable connections, an optical network is the most suitable solution to transport data from the remote DC to edge DCs and from edge DCs to the requesting node. The high-capacity optical network is exposed to many threats such as malicious attacks, equipment failures, human errors (e.g., misconfigurations), and natural and human-made large-scale disasters (e.g., earthquakes, hurricanes, and terrorist attacks). To ensure reliability, restoration and protection schemes are traditionally used. In a restoration scheme, no resources are reserved in advance and the network must react to find an alternate path after a failure occurs on the working path [11]. However, a restoration scheme would require a longer recovery time, provide no guarantee to restore a disrupted path, and hence, is not suitable for ER live streams where latency is critical. In a protection scheme, extra network resources are reserved when a connection is provisioned. Conventionally, a pair of paths is provided to a connection: one is used to carry traffic during normal operation, referred to as primary path; and the other path, referred to as backup path, is reserved and will be activated after a failure occurs on the primary path. This dedicated protection scheme (i.e., 1+1) provides fast recovery and is suitable for ER live streams.

In Fig. 4.1.a, we represent a protection scheme, namely *backup from the same data center* (BSD) where the ER live stream from the remote DC in node 15 is simultaneously transmitted to the edge DC in node 3 using the primary path (solid line) and the backup path (dashdot line). This edge DC requires powerful computing capability and large storage memory to host a local video server that receives the ER live stream from the remote DC and makes it available to the requesting node. Since the latency between the requesting node and the edge DC is critical, in this work, the DC closest to the requesting node (i.e., DC in node 3 which is closest to node 1) is selected to serve the request. Finally, immersive experience is delivered to the requesting node (and ultimately to the end user) using the primary and backup paths. This service-provisioning scheme guarantees that the end user has lowest latency (it is served by the closest DC), recovery time is minimal, and connections are survivable against a single-link failure in the optical network (which is still the dominant failure scenario in an optical network [Chapter 2], [38]). However, it provides no protection against failures of computing and storage in the selected DC which is also an important issue to ensure end-to-end network resilience [62].

In Fig. 4.1.b, we propose a service-provisioning scheme, namely *backup from different data centers* (BDD), where the ER live stream from node 15 is simultaneously transmitted to two different DCs in nodes 3 and 4 using the primary and backup paths (solid and dashdot lines, respectively). In addition to protection against a single-link failure in the optical network, this service-provisioning scheme also provides protection against failures of computing and storage in DCs. During normal operation, the requesting node is served by the closest DC in node 3 using the primary path. In case of a failure occurring on the primary path, the provisioned service is still guaranteed after the backup path (via the DC in node 4) is quickly activated. Some studies have also showed that this service-provisioning scheme provides great flexibility to reserve resources among DCs [5, 61].

In Figs. 4.1.a and b, high-bandwidth connections from the remote DC are required to make the ER live stream available in edge DCs. Hence, network resources may not be efficiently utilized. In Fig. 4.1.c, we consider BDD which is leveraged by exploiting the multicast functionality within the optical layer and metro network to reduce unnecessary capacity in the network [63]. For convenience, we use *BDD with multicast* (BDD-M, M for multicast) to denote this service-provisioning scheme. In this scenario, another request is arriving in node 5 for the same ER live stream which is still active in DCs in nodes 3 and 4 (to serve node 1). Instead of establishing connections from the remote DC (node 15) to the closest DC in node 6, this service-provisioning scheme utilizes the multicast functionality of optical nodes to establish a connection from node 3 to make the ER live stream available in the closest DC in node 6. To provide protection against link, computing, and storage failures, the backup path (for node 5) is now routed from node 4 via nodes 1 and 2. Also, the connection from node 15 to node 3 and the multicast connection (from node 3 to 6, dotted line) can share a link or node (e.g., link 3-7) because they are not required to be link/node disjoint. For simplicity, in Fig. 4.1.c, we only show the multicast connection from node 3 to node 6. In reality, there may be other nodes with multicast connections from node 3 which serves as the root of the multicast tree.

Finally, a service-provisioning scheme should be aware of the unique trade-off between latency and bandwidth for ER live streams [56, 64]. In general, lower latency requires less bandwidth to meet a certain ER resolution. This trade-off is directly related to the fact that, with lower latency, fewer number of viewpoints around the live event are required to be sent over the network. For an instance of advanced VR, if end-to-end latency is on the order of 1 ms, the bandwidth for the ER live stream can be reduced to 100 Mbps. However, if end-to-end latency is 20 ms, the bandwidth for the ER live stream must be up to 400 Mbps [64, 65]. As a result, the closer the hosting DC is pushed to the edge (the aim of this work), the more network bandwidth a service-provisioning scheme can save. In this study, we propose a service-provisioning scheme which pushes the requested ER live stream as close to the requesting node as possible, estimates the end-to-end latency (including latency of access networks, e.g., a 6G mobile network), and flexibly offers bandwidth to an ER live stream. Below, we use *BDD with multicast and flexible offered bandwidth* (BDD-MF) to denote BDD which supports both multicast and flexibility of offered bandwidth (i.e., based on latency). Therefore, BDD-MF is a service-provisioning scheme including several subordinate algorithms to admit an ER live stream request based on end-to-end latency and the availability of the requested stream (i.e., the requested stream is available in the closest DC, in the DC cluster, or in the remote DC). We will elaborate on these algorithms in Section 4.3.

Our contributions are as follows. In this chapter, we propose BDD-MF as a service-provisioning scheme to fulfill ER live stream requests while guaranteeing strict requirements on reliability, latency, and bandwidth. Our proposed service-provisioning scheme provides protection not only against failures of links in the network but also against failures of computing and storage in DCs. Moreover, for the first time and to the best of our knowledge, we consider the multicast functionality of optical nodes and the tradeoff between latency and bandwidth of ER live streams to reduce unnecessary network capacity. We develop scalable algorithms for BDD-MF and use them to solve various complex network instances in a dynamic network environment. Numerical data show that, compared to a conventional service-provisioning scheme such as BSD, our proposed service-provisioning scheme provides higher reliability, reduces latency, and efficiently utilizes network resources; hence, it is highly suitable for ER live streams.

The rest of this study is organized as follows. In Section 4.2, we state the BDD-MF problem and develop scalable heuristics for it. In Section 4.3, we perform numerical validation in various scenarios and compare the performance of BDD-MF to that of reference protection strategies (e.g., BSD). We conclude this study in Section 4.4.

4.2 Reliable Provisioning of Low-latency and High-bandwidth Extended Reality Live Streams

4.2.1 Problem Statement

We consider a service provider with a CDN represented by a graph $G_t(V_t, E_t)$ where V_t is the set of nodes with available resources (i.e., throughput, computing, and storage) and E_t is the set of links with available bandwidth at request arrival time (i.e., denoted by t in the tuple below). Since ER live stream requests come, hold, and depart, the sets V_t and E_t can vary over time. Within this network graph, we also use D to denote the set of edge DCs in a cluster close to requesting node ($D \in V_t$), S to denote the set of ER live streams offered by the service provider, and R to denote the set of remote nodes potentially being the locations of live events ($R \in V_t$). Without loss of generality, $D \cap R = \varnothing$ (i.e., the location of a live event is far away from edge DCs).

In this study, each ER live stream request, Ω, is characterized by a tuple,

$$\Omega = (b, h, l, m, n, r, s, t),$$

where b is requested bandwidth in normal operation (Mbps), h is holding time of the stream (s), l is requested latency (ms), m is degraded bandwidth (Mbps) if the primary path is disrupted, n is the requesting node (i.e., $n \in V_t \setminus D \setminus R$), r is the location of live event (i.e., $r \in R$), s is the stream ID (i.e., $s \in S$), and t is arrival time (s).

The problem of reliable provisioning of low-latency and high-bandwidth ER live streams can be formally stated as follows. *Given* the network graph at request arrival time $G_t(V_t, E_t)$ and the ER live stream request $\Omega = (b, h, l, m, n, r, s, t)$, *admit* the request *such that* the requested bandwidth b, latency l, and degraded bandwidth m are fulfilled. The solution is subject to the constraints on the available bandwidth of each physical link, throughput in each transit node, computing and storage capacity in each DC, and survivability of paths. We use BDD-MF (illustrated in Fig. 4.1) to fulfill the above problem in which, for each request arrival, we consider the following scenarios.

First, we consider the scenario where the requested ER live stream is geographically available in the closest DC (to the requesting node). This scenario also implies that the same ER live stream is available in another DC in the cluster since it is initially transmitted to two DCs. In this case, BDD-MF is reduced to finding a pair of link-disjoint paths from the first and second closest DCs (with the active ER live stream) to the requesting node and reserve bandwidth on each link, throughput in each node, and computing and storage capacity in each serving DC. We elaborate on this scenario in `Algorithm` 3.

Second, we consider the scenario where the requested ER live stream is not available in the closest DC (to the requesting node) but available in multiple DCs in the cluster. In this case, BDD-MF performs the following steps: a) find the active ER live stream in the DC cluster and make it available in the closest DC using multicast, b) find a pair of link-disjoint paths from the closest DC (now with the active ER live stream using multicast) and the second closest DC with the active ER live stream (as a backup DC) to the requesting node and reserve bandwidth on each link, throughput in each node, and computing and storage capacity in each serving DC. To avoid single point of failures for primary and backup paths, the backup DC must be different from the root of the multicast tree. We elaborate on this scenario in `Algorithm` 4.

Third, we consider the scenario where the requested ER live stream is only available in the location of the live event. In this case, BDD-MF performs the following steps: a) make the ER live stream available in the first and second closest DCs (to the requesting node) using a pair of link-disjoint paths from the location of the live event, b) find a pair of link-disjoint paths from the first and second closest DCs (now with the active ER live stream) to the requesting node and reserve bandwidth on each link, throughput in each node, and computing and storage capacity in each DC. We elaborate on this scenario in `Algorithm` 5.

Before providing the pseudo codes of `Algorithm` 3, `Algorithm` 4, and `Algorithm` 5, we provide the model of latency budget and a flexible scheme to offer bandwidth to an ER live stream request.

4.2.2 Model of Latency Budget and Trade-off Between Latency and Bandwidth

In this section, we elaborate on the model of latency budget and the scheme to flexibly offer bandwidth to an ER live stream request, leveraging the trade-off between latency and bandwidth for ER live streams.

To estimate the end-to-end latency between an end user to the serving edge DC, we assume that a mobile network is used as the access network. It is worth noting that it is harder to meet the latency requirement of the ER live stream on a mobile network (compared to a Wi-Fi or fixed network). We adopt the following model of latency budget [65, 66],

$$T_{e2e} = T_{Rad} + T_{Fh} + T_{Mh} + T_{Proc} + T_{Rend} + T_{Prop}, \tag{4.1}$$

where T_{e2e}, T_{Rad}, T_{Fh}, T_{Mh}, T_{Proc}, T_{Rend}, and T_{Prop} are the total end-to-end latency, radio transmission time between the antenna and user equipment, front-haul network latency, mid-haul network latency, processing time, rendering and refresh time, and propagation delay, respectively. Since our focus in this study is on the back-haul network (e.g., the optical network between a CO and the location of a live event), we pay our attention to T_{Prop} which is the propagation time between the requesting node and edge DC. Note that, for a specific access technology (e.g., 5G), network configuration (e.g., a placement of centralized units (CUs) and distributed units (DUs)) DUs and CUs), and ER resolution (e.g., advanced VR), T_{Rad}, T_{Fh}, T_{Mh}, T_{Proc}, and T_{Rend} are irreducible. We believe that these terms will be further improved in next-generation networks (e.g., 6G). The reader can refer to [66–68] for further details. In this study, we investigate the service-provisioning scheme which pushes an ER live stream and makes it available to the edge DC as close to the request node as possible, hence, it reduces the propagation (i.e., T_{Prop}) and overall delay. Based on the overall end-to-end latency obtained in (4.1), without loss of generality, we use a linear model (latency vs. offered bandwidth) to flexibly offer bandwidth to meet the requirements of an ER live stream request.

4.2.3 Backup from Different DCs With Multicast and Flexible Bandwidth

In the following sections, we consider each scenario in Section 4.2.1 and provide the pseudo codes of `Algorithm` 3, `Algorithm` 4, and `Algorithm` 5. In this study, we use Dijkstra's algorithm to find the shortest path between two nodes in a network graph [54]. Also, we use the extended Bhandari's algorithm to find all link-disjoint paths with minimal total length between a pair of nodes [69]. To assist the extended Bhandari's algorithm to find all link-disjoint paths from one node to a set of nodes, we introduce an auxiliary graph by leveraging a dummy node and several dummy links (i.e., `aux` in `Algorithm` 3). The reader can refer to Chapter 3 for details of how to build the auxiliary graph.

Below, we also use d_p, d_b, d_m, and d_r to denote the primary DC (d_p is the closest DC to the requesting node), backup DC, multicast DC (i.e., the root of the multicast tree), and location of live event (i.e., a remote DC), respectively (d_p, d_b, and $d_m \in D$; $d_r \in R$). To reduce overall latency, the closest DC to the requesting node is selected as the primary DC to serve an ER live stream request. For consistence, the three algorithms take $G_t(V_t, E_t)$, D, S, R, $\Omega = (b, h, l, m, n, r, s, t)$, and ER stream availability as *inputs* and return *outputs* Δ, Σ, Φ, and Ψ as the hash maps representing bandwidth reserved on each link, throughput reserved on each node, computing reserved in each DC, and storage reserved in each DC, respectively. In case the network finds insufficient resources, the ER live stream is blocked and the algorithms return no results (i.e., $\Delta = \Sigma = \Phi = \Psi = \varnothing$). Note that the following algorithms are derived for a single ER live stream request considering the updated network graph. To obtain the numerical results reported in this study, we integrate them into a dynamic simulation framework as in Fig. 4.2. Also, the dynamic simulation framework initially verifies the location of the closest DC to the requesting node and the availability of the requested ER live stream in the CDN to assign the request to a proper algorithm (d_p is given to each of the following algorithms). In this study, we use s in d_p, s in D, and s in d_r to denote the requested ER live stream is available in the primary DC, in the DC cluster, and in the remote DC, respectively.

4.2.3.1 Extended reality live stream is available in the closest data center

Algorithm 3: Live stream available in the closest DC.

```
   Input: $G_t(V_t, E_t)$, $D$, $S$, $R$, $\Omega = (b, h, l, m, n, r, s, t)$, $d_p$, $s$ in $d_p$
   Output: $\Delta$, $\Sigma$, $\Phi$, $\Psi$
   Data: $\Delta = \Sigma = \Phi = \Psi = P_D = \varnothing$, $d_b =$ null
1  for $d_i$ in $D$ do
2      if $s$ in $d_i$ then
3          $p_i$ = Dijkstra($G_t(V_t, E_t)$, $n$, $d_i$); append $p_i$ to $P_D$
4  if $|P_D| \geq 2$ then
5      sort($P_D$, ascending = true, key = length)
6      $p_0 = P_D[0]$; $d_0 \leftarrow p_0$
7      for $p_i$ in $P_D$ do
8          $d_i \leftarrow p_i$
9          if $d_i \neq d_0$ then $d_b = d_i$; break
10     if $d_0 = d_p$ and $d_b \neq$ null then
11         $G'_t(V_t, E_t) = \text{aux}(G_t(V_t, E_t), \{d_p, d_b\}, d')$
12         $P_B$ = Bhandari($G'_t(V_t, E_t), n, d'$)
13         if $|P_B| = 2$ then
14             $p_p = P_B[0]$; $p_b = P_B[1]$
15             if valid($G_t(V_t, E_t)$, $d_p$, $d_b$, $p_p$, $p_b$, $\Omega$) then
16                 $T_{Prop} = T_{abs}(p_p, p_b)$; $T_{e2e} \leftarrow T_{Prop}$
17                 if $T_{e2e} \leq l$ then
18                     $b'_p = \frac{b_{\max} - b_{\min}}{T_{\max} - T_{\min}} * (T_{e2e} - T_{\min}) + b_{\min}$
19                     $b'_b = m * b'_p$; populate $\Delta$, $\Sigma$, $\Phi$, and $\Psi$
20                     return $\Delta$, $\Sigma$, $\Phi$, $\Psi$
```

In `Algorithm` 3, we represent the pseudo code to provision an ER live stream request in case the stream is active in the closest DC (s in d_p). This scenario implies that the requested ER live stream is also available in a different edge DC. From lines 1 to 3, the `Algorithm` 3 finds the shortest path from the requesting node to each DC with the active ER stream and add the path to P_D. If the ER live stream is active in at least two different DCs (including the closest DC to the requesting node) with valid paths, the algorithm continues to sort the paths in P_D by their total length in ascending order and finds the first and second closest DCs (lines 4-9). If the first and second closest DCs are found, the algorithm constructs an auxiliary graph using these two nodes, a dummy node, and dummy links [Chapter 3] (line 11). In line 12, the algorithm uses the Bhandari's extended algorithm to find a pairs of link-disjoint paths from the requesting node to the first and second closest DCs (with the active ER live stream). If the pair of the link-disjoint paths is found, the algorithm verifies that the network resources are sufficient along the primary and back paths, estimates the absolute propagation delay of the paths, and computes end-to-end latency using the model in (Eqn. 4.1) (lines 13-16). If the end-to-end latency fulfills the ER latency requirement, the ER live request is admitted and the algorithm updates the bandwidth reserved on each link, throughput reserved on each node, computing reserved in each DC, and storage reserved in each DC using the hash maps Δ, Σ, Φ, and Ψ respectively. In case the network has insufficient resources, the ER live stream request is rejected and $\Delta = \Sigma = \Phi = \Psi = \varnothing$ (i.e., no network resources are reserved).

4.2.3.2 Extended reality live stream is available in the data center cluster

In this scenario, the requested ER live stream is not available in the closest DC but available in multiple locations in the DC cluster. `Algorithm` 4 exploits the multicast functionality of optical nodes to make the requested ER live stream available in the closest node and provide it to the requesting node. From lines 1 to 3, the algorithm finds the shortest path from the requesting node to each DC and add it to P_D. Moreover, the DC with the active ER live stream is added to D_0. If there exists multiple paths and the requested ER live stream is available in multiple locations in the DC cluster, the algorithm continues to sort the paths in P_D by their total length in ascending order and find the closest DC (without the requested ER live stream) as the primary DC (lines 4-6). From lines 7 to 13, the algorithm finds the second and third closest DCs with the active ER live stream (which are not the primary DC) to function as the root of the multicast tree and the backup DC, respectively. If the root of the multicast tree and the primary DC are valid, the algorithm computes the primary and multicast paths (p_p and p_m) using the Dijkstra's algorithm (lines 15 and 16). If the primary and multicast paths are found, `Algorithm` 4 uses the Bhandari's extended algorithm to find all link-disjoint paths (P_B) from the backup DC to the requesting node (line 18) and sorts them by their total length in ascending order (line 19). From lines 20 to 22, the algorithm computes the backup path which is the shortest path in P_B and link-disjoint with both primary and multicast paths.

Algorithm 4: Live stream available in the DC cluster.

```
   Input: $G_t(V_t, E_t)$, $D$, $S$, $R$, $\Omega = (b, h, l, m, n, r, s, t)$, $d_p$, $s$ in $D$
   Output: $\Delta$, $\Sigma$, $\Phi$, $\Psi$
   Data: $\Delta = \Sigma = \Phi = \Psi = P_D = D_0 = \varnothing$, $d_m = d_b =$ null
 1 for $d_i$ in $D$ do
 2     $p_i$ = Dijkstra($G_t(V_t, E_t)$, $n$, $d_i$); append $p_i$ to $P_D$
 3     if $s$ in $d_i$ then add $d_i$ to $D_0$
 4 if $|P_D| \geq 2$ and $|D_0| \geq 2$ then
 5     sort($P_D$, ascending = true, key = length)
 6     $p_0 = P_D[0]$; $d_0 \leftarrow p_0$
 7     if $d_0 \notin D_0$ and $d_0 = d_p$ then
 8         for $p_i$ in $P_D$ do
 9             $d_i \leftarrow p_i$
10             if $d_i \neq d_0$ and $d_i \in D_0$ then $d_m = d_i$; break
11         for $p_i$ in $P_D$ do
12             $d_i \leftarrow p_i$
13             if $d_i \neq d_0$ and $d_i \neq d_m$ and $d_i \in D_0$ then $d_b = d_i$; break
14         if $d_m \neq$ null and $d_b \neq$ null then
15             $p_p$ = Dijkstra($G_t(V_t, E_t)$, $d_p$, $n$)
16             $p_m$ = Dijkstra($G_t(V_t, E_t)$, $d_m$, $d_p$)
17             if $p_p$ and $p_m$ then
18                 $P_B$ = Bhandari($G_t(V_t, E_t)$, $d_b$, $n$)
19                 sort($P_B$, ascending = true, key = length)
20                 for $p_i$ in $P_B$ do
21                     if $p_i \cap p_p = \varnothing$ and $p_i \cap p_m = \varnothing$ then
22                         $p_b = p_i$; break
23             if $p_p$ and $p_m$ and $p_b$ and valid($G_t(V_t, E_t), d_p, d_m, d_b, p_p, p_m, p_b, \Omega$) then
24                 $T_{Prop} = T_{abs}(p_p,\ p_b)$; $T_{e2e} \leftarrow T_{Prop}$
25                 if $T_{e2e} \leq l$ then
26                     $b'_p = \frac{b_{\max} - b_{\min}}{T_{\max} - T_{\min}} * (T_{e2e} - T_{\min}) + b_{\min}$
27                     $b'_b = m * b'_p$; populate $\Delta$, $\Sigma$, $\Phi$, and $\Psi$
28                     return $\Delta$, $\Sigma$, $\Phi$, $\Psi$
```

If the primary, multicast, and backup paths are found, the algorithm verifies that the network resources are sufficient along the paths, estimates the absolute propagation delay of the paths, and computes end-to-end latency using the model in (Eqn. 4.1) (lines 23 and 24). If the end-to-end latency fulfills the ER latency requirement, the ER live request is admitted and the algorithm updates the bandwidth reserved on each link, throughput reserved on each node, computing reserved in each DC, and storage reserved in each DC using the hash maps Δ, Σ, Φ, and Ψ respectively (line 28). In case the network has insufficient resources, the ER live stream request is rejected and $\Delta = \Sigma = \Phi = \Psi = \varnothing$ (i.e., no network resources are reserved).

4.2.3.3 Extended reality live stream is only available in the location of the live event

Algorithm 5: Live stream available in the location of live event.

```
   Input: $G_t(V_t, E_t)$, $D$, $S$, $R$, $\Omega = (b, h, l, m, n, r, s, t)$, $d_p$, $s$ in $d_r$
   Output: $\Delta$, $\Sigma$, $\Phi$, $\Psi$
   Data: $\Delta = \Sigma = \Phi = \Psi = P_D = D_0 = \varnothing$, $d_b =$ null
 1 for $d_i$ in $D$ do
 2     $p_i =$ Dijkstra($G_t(V_t, E_t)$, $n$, $d_i$); append $p_i$ to $P_D$
 3     if $s$ in $d_i$ then add $d_i$ to $D_0$
 4 if $|P_D| \geq 2$ and $|D_0| = \varnothing$ then
 5     sort($P_D$, ascending = true, key = length)
 6     $p_0 = P_D[0]$; $d_0 \leftarrow p_0$; $p_1 = P_D[1]$; $d_b \leftarrow p_1$
 7     if $d_0 = d_p$ then
 8         $p_p^r =$ Dijkstra($G_t(V_t, E_t)$, $r$, $d_p$)
 9         $p_p =$ Dijkstra($G_t(V_t, E_t)$, $d_p$, $n$)
10     if $p_p$ and $p_p^r$ then
11         $P_B^R =$ Bhandari($G_t(V_t, E_t)$, $r$, $d_b$)
12         $P_B =$ Bhandari($G_t(V_t, E_t)$, $d_b$, $n$)
13         for $p_i$ in $P_B$ do
14             if $p_i \cap p_p = \varnothing$ and $p_i \cap p_p^r = \varnothing$ then
15                 $p_b = p_i$; break
16         for $p_i$ in $P_B^R$ do
17             if $p_i \cap p_p = \varnothing$ and $p_i \cap p_p^r = \varnothing$ then
18                 $p_r'' = p_i$; break
19         if valid($G_t(V_t, E_t), d_p, d_b, p_p, p_b,\ p_p^r,\ p_b^r$) then
20             $T_{Prop} = T_{abs}(p_p,\ p_b)$; $T_{e2e} \leftarrow T_{Prop}$
21             if $T_{e2e} \leq l$ then
22                 $b_p' = \frac{b_{\max} - b_{\min}}{T_{\max} - T_{\min}} * (T_{e2e} - T_{\min}) + b_{\min}$
23                 $b_b' = m * b_p'$; populate $\Delta$, $\Sigma$, $\Phi$, and $\Psi$
24                 return $\Delta$, $\Sigma$, $\Phi$, $\Psi$, $t_d$
```

In `Algorithm` 5, we represent the pseudo code to retrieve the requested ER live stream from the location of the live event when it is not available in the DC cluster. From lines 1 to 3, the algorithm finds the shortest path from the requesting node to each DC in the cluster and adds them to a list (P_D). Moreover, the DC with the active ER live stream is added to D_0. If there exist multiple paths (i.e., there are multiple survivable

DCs in the cluster), the algorithm continues to sort the paths in P_D by their total length in ascending order and selects the first and second closest DCs to the requesting node as the primary and backup DCs, respectively (lines 4-6). In case the primary and backup DCs are valid, the algorithm uses the Dijkstra's algorithm to compute the primary paths from the location of the live event to the primary DC and from the primary DC to the requesting node (lines 7-9). If the primary paths are found, the algorithm uses the Bhandari's extended algorithm to find all link-disjoint paths between the location of the live event and the backup DC and between the backup DC and the requesting node (lines 11 and 12). From lines 13 to 18, the algorithm computes the backup paths from the location of the live event to the backup DC and from the backup DC to the requesting node. To guarantee the survivability of the backup paths in case a failure occurs on the primary paths, the algorithm finds the backup paths such that they are link-disjoint with the primary paths. If the primary and backup DCs, primary and backup paths are found, the algorithm verifies that the network resources are sufficient in each DC and along the paths, estimates the absolute propagation delay of the paths, and computes end-to-end latency using the model in (4.1) (lines 19 and 20). If the end-to-end latency fulfills the ER latency requirement, the ER live request is admitted and the algorithm updates the bandwidth reserved on each link, throughput reserved on each node, computing reserved in each DC, and storage reserved in each DC using the hash maps Δ, Σ, Φ, and Ψ respectively (line 24). In case the network has insufficient resources, the ER live stream request is rejected and $\Delta = \Sigma = \Phi = \Psi = \varnothing$ (i.e., no network resources are reserved).

4.3 Illustrative Numerical Results

4.3.1 Physical Networks and Simulation Setup

4.3.1.1 Physical Networks

To evaluate our proposed solutions, we use the Tokyo23 metro network covering an urban area up to tens of kilometers in diameter as in Fig. 2.5 [27]. The Tokyo23 metro network has been designed using regional characteristics such as population distribution, locations of local government offices, and railway lines with the number of passengers getting on/off each station. It consists of 43 bi-directional links and 23 nodes, with each node located at each ward office building in the Tokyo metropolitan area. We also use the Japan Photonic Network with 25 nodes and 43 bi-directional links as the backbone network (JPN25 network) for the metro network [70]. In other words, the combined network (including metro and backbone networks) consists of 48 nodes (23 metro nodes, 1-23, and 25 backbone nodes, 24-48) and 86 bi-directional links (43 100-Gbps, bi-directional metro links and 43 400-Gbps, bi-directional backbone links). The metro network is connected to the backbone network through nodes 12 (Setagaya to Kanagawa), 18 (Arakawa to Saitama), and 23 (Edogawa to Chiba). We consider a service provider with 1000 ER live streams (i.e, $|S| = 1000$) in the catalog whose popularity is modeled as a Zipf distribution $[32, 55, 66]$. The sets $D = \{1, 7, 12, 13, 16\}$ and $R = \{24, 32, 34, 40, 42\}$ denote the set of edge DCs in a geographical cluster in the metro network and the set of nodes as the potential locations of live events (remote DCs in the backbone network).

4.3.1.2 Simulation Setup

In this study, each ER live stream request, $\Omega = (b, h, l, m, n, r, s, t)$, is generated whose characteristics follow the analysis of live streaming workloads on the Internet [71]. The data reported in [71] were collected over a 3 month period and contain over 70 million requests for 5,000 distinct URLs from clients in over 200 countries at Akamai Technologies, Inc. (a major CDN). Even though the analysis is for generic live streams (at the time ER was not technologically ready), we believe that it is highly suitable to characterize ER live stream requests. In each ER live stream request, the requested bandwidth, b, is uniformly selected from 4 values, 25 Mbps, 100 Mbps, 400 Mbps, and 1000 Mbps corresponding to 4 levels of resolution of the ER stream, early-stage ER, entry-level ER, advanced ER, and extreme ER, respectively [65]. The holding time of the ER live stream, h, is modeled using a truncated Pareto distribution with the minimum and maximum duration of 15 and 120 minutes. The maximum latency, l, for the required bandwidths mentioned above is 20 ms. Note that, in case computed latency is smaller the maximum latency, BDD-MF is flexible to offer less bandwidth (than the requested bandwidth) to an ER live stream

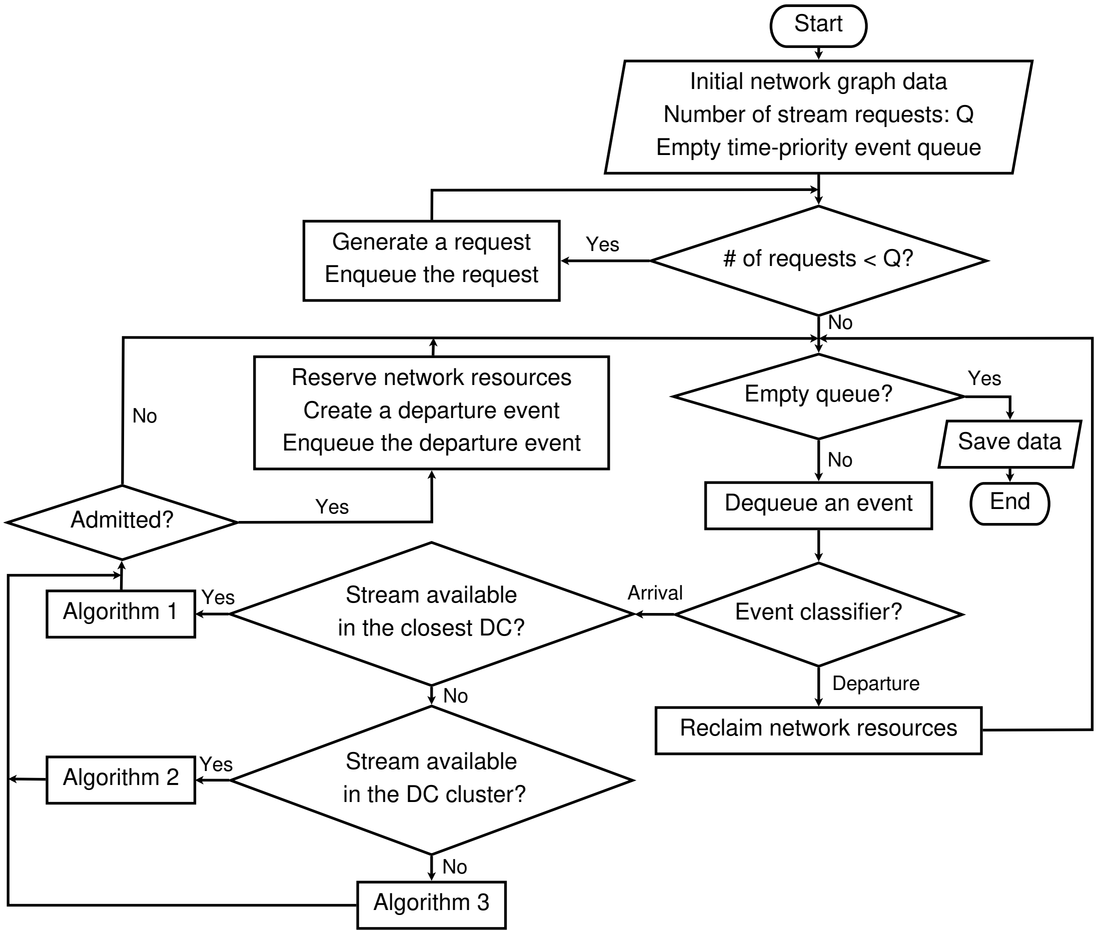

Start
Initial network graph data
Number of stream requests: Q
Empty time-priority event queue
# of requests < Q?
Yes
Generate a request
Enqueue the request
No
Empty queue?
Yes
Save data
End
No
Dequeue an event
Event classifier?
Arrival
Departure
Reclaim network resources
Stream available in the closest DC?
Yes
Algorithm 1
No
Stream available in the DC cluster?
Yes
Algorithm 2
No
Algorithm 3
Admitted?
Yes
Reserve network resources
Create a departure event
Enqueue the departure event
No

Figure 4.2: Simulation framework.

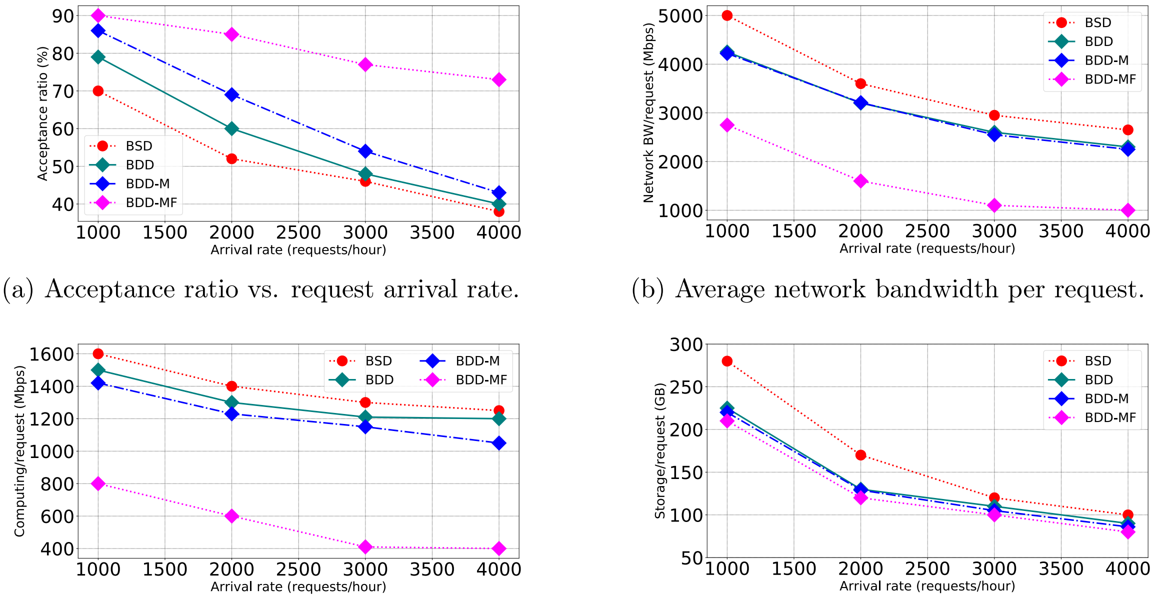
request. The degraded service level, m, is uniformly selected from $\{0.5,\ 0.7,\ 1.0\}$ where $m = 1.0$ denotes full protection in case the primary path is disrupted (i.e., the ratio of survivable bandwidth to requested bandwidth in case the primary path is disrupted is 1.0). The requesting node, n, is uniformly selected among metro nodes which are not edge DCs (i.e, $n \notin D$). The location of the live event, r, is uniformly selected from the set R (i.e., $r \in R$). The ER live stream ID, s, is selected from the catalog of the service provider following the popularity of the ER live streams. Finally, ER live stream arrival time, t, is modeled as a discrete Poisson process.

For our experiment setup, we use the dynamic framework in Fig. 4.2 to simulate the network with 105000, 205000, 305000, 405000, and 505000 ER live stream requests. We first generate all ER live stream requests and enqueue them in a time-priority queue and start with an empty network (i.e., a network with no active traffic). During simulation, each ER live stream request is dequeued from the queue and provisioned using a) `Algorithm` 3 if the requested ER live stream is geographically available in the closest DC (i.e., s in d_p); b) `Algorithm` 4 if the requested ER live stream is geographically available in the DC cluster (i.e., s in D); and c) `Algorithm` 5 if the requested ER live stream is available in the remote DC (i.e., s in d_r). If an ER live stream request is admitted, required network resources are reserved for it, and a departure event with departure time equal to the content request arrival time plus the holding time is enqueued to the queue. If the event is a departure, reserved network resources are released. Note that the simulator processes one event at a time (either an arrival or a departure), and it proceeds with the next event in queue as soon as it finishes the current one. We also found that the network requires approximate 5000 ER live stream requests to reach a steady state, and numerical results obtained for simulating 105000 and 505000 content requests are comparable. To reduce experiment time, below we report the numerical results for simulating 105000 ER live stream requests (i.e., first 5000 ER live stream requests are discarded and the acceptance ratio is the ratio of the number of admitted requests to the number of simulated requests, e.g., 100000 in this study). In the next section, we will use the congestion point (η_0) and acceptance ratio (η) defined in Chapter 3 to evaluate our proposed service-provisioning schemes in `Algorithm` 3, `Algorithm` 4, and `Algorithm` 5, and compare their performance to those of reference schemes (BSD, BDD, and BDD-M).

4.3.2 Simulation Results

In Fig. 4.3, we report the numerical results of our proposed service-provisioning scheme namely BDD-MF vs. the reference schemes, i.e., BSD, BDD, and BDD-M. In this study, we compare the performance of BDD-MF to the performance of the reference schemes in four aspects including acceptance ratio, network bandwidth, computing capacity in DCs, and storage capacity in DCs.

In Fig. 4.3.a, we report the acceptance ratio as a function of request arrival rate for all service-provisioning schemes. As expected, the acceptance ratio is increased from



(a) Acceptance ratio vs. request arrival rate.

(b) Average network bandwidth per request.

(c) Average computing capacity per request.

(d) Average storage capacity per request.

Figure 4.3: Backup from different data centers with multicast and flexible bandwidth vs. reference schemes.

BSD, BDD, BDD-M, and BDD-MF. In other words, BDD outperforms BSD such that it can admit more incoming ER live stream requests. In detail, the introduction of the multicast functionality in optical nodes improves the utilization of network resources and increases the acceptance ratio. The acceptance ratio is further improved in case the offered bandwidth to each ER live stream request is flexible based on end-to-end latency between the requesting node and edge DCs. It is worthy to mention that, compared to BSD, BDD, BDD-M, and BDD-MF also provide protection against the failures of computing and storage in DCs. Figure 4.3.b shows the average network bandwidth (defined in Section 3.1) per request in Mbps. It shows that the multicast in the optical layer and the flexibility of offered bandwidth significantly reduce the average network bandwidth. This can be explained as a few ER live stream requests are made available to users without establishing high-bandwidth connections from the location of the live event to edge DCs. Similarly, since no additional high-bandwidth connections are established from the location of the live event to edge DCs, our proposed scheme uses less computing resources and storage capacity in DCs (Figs. 4.3.c and d).

4.4 Conclusion

In this chapter, we considered the scenario where the contents are extended reality live streams. Compared to contents investigated in Chapter 3, extended reality live streams are not cacheable and require high bandwidth, low latency, and reliable connections. To meet these stringent requirements (especially on latency which should be less then a few ms), we proposed backup from different data centers with multicast and flexible offered bandwidth to fulfill the extended reality live streams. Our service-provisioning scheme provides protection not only against the failures on links in the physical network but also against the failures of computing and storage in DCs. Numerical results show that our service-provisioning scheme efficiently utilizes networks resources, hence, it can admit more incoming extended reality live stream requests.

Chapter 5

Conclusion

5.1 Summary

In this dissertation, we considered content connectivity as a network survivability metric against failures and investigated how to ensure content connectivity in various applications under different scenarios of failures in the physical network. We studied content connectivity in three distinct applications: survivable virtual network mapping with content connectivity against multiple link failures, reliable provisioning with degraded service using multipath routing from multiple data centers, and reliable provisioning of low-latency and high-bandwidth extended reality live streams. We proposed solutions for each of the above applications and simulated their performance under various network settings. Numerical results show that, in all the above-mentioned applications, content connectivity has higher survivability, particularly against a large-scale disaster, saves network bandwidth, significantly improves service availability, and hence, it is a highly-suitable network survivability metric.

5.2 Future Works

The problems in Chapters 3 and 4 are dynamic problems in which each time a request is admitted, resources are reserved and the network moves to another state. We are interested in formulating these problems as reinforcement learning problems using ideas from dynamical systems theory, specifically, as the optimal control of incompletely-known

Markov decision processes [72,73]. The whole dynamic simulation framework with service-provisioning algorithms can function as a learning agent interacting over time with its network to admit as many requests as possible (goal). The learning agent must be able to sense the state of its network to some extent and must be able to take actions that affect the state. The principle of this learning problem is straightforward. However, the application in practice is limited since it is very hard to characterize the network state. For future works, we are interested in the problem of deep reinforcement learning to learn the correct online service-provisioning policies by parameterizing the policies with deep neural networks that can sense a complex network state.

Bibliography

[1] Cisco, "Cisco visual networking index (VNI): Forecast and trends 2017 – 2022," *White Paper*, 2019.

[2] M. F. Habib, M. Tornatore, and B. Mukherjee, "Fault-tolerant virtual network mapping to provide content connectivity in optical networks," in *Optical Fiber Communication Conference (OFC)*, March 2013.

[3] G. Le, A. Marotta, S. Ferdousi, S. Xu, Y. Hirota, Y. Awaji, M. Tornatore, and B. Mukherjee, "Logical network mapping with content connectivity against multiple link failures in optical metro networks," in *Proc. of IEEE ANTS*, 2019.

[4] G. Le, S. Ferdousi, A. Marotta, S. Xu, Y. Hirota, Y. Awaji, M. Tornatore, and B. Mukherjee, "Survivable virtual network mapping with content connectivity against multiple link failures in optical metro networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 12, no. 11, pp. 301–311, 2020.

[5] ——, "Reliable provisioning for dynamic content requests in optical metro networks," in *Proc. of IEEE/OSA OFC*, 2021.

[6] G. Le, S. Ferdousi, A. Marotta, S. Xu, Y. Hirota, Y. Awaji, S. Savas, M. Tornatore, and B. Mukherjee, "Reliable provisioning with degraded service using multipath routing from multiple data centers in optical metro networks," *IEEE Transactions on Network and Service Management (submitted)*, 2022.

[7] S. P. Mohanty, U. Choppali, and E. Kougianos, "Everything you wanted to know about smart cities: The Internet of Things is the backbone," *IEEE Consumer Electronics Magazine*, vol. 5, no. 3, pp. 60–70, July 2016.

[8] S. Parkvall, E. Dahlman, A. Furuskar, and M. Frenne, "NR: The new 5G radio access technology," *IEEE Communications Standards Magazine*, vol. 1, no. 4, pp. 24–30, Dec. 2017.

[9] A. Marotta, D. Cassioli, M. Tornatore, Y. Hirota, Y. Awaji, and B. Mukherjee, "Reliable slicing with isolation in optical metro-aggregation networks," in *Optical Fiber Communication Conference (OFC)*, 2020.

[10] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, 2017.

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

[12] E. Modiano and A. Narula-Tam, "Survivable lightpath routing: a new approach to the design of WDM-based networks," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, pp. 800–809, May 2002.

[13] E. J. Dávalos and B. Barán, “A survey on algorithmic aspects of virtual optical network embedding for cloud networks,” *IEEE Access*, vol. 6, 2018.

[14] M. R. Rahman and R. Boutaba, “SVNE: Survivable virtual network embedding algorithms for network virtualization,” *IEEE Transactions on Network and Service Management*, vol. 10, no. 2, pp. 105–118, 2013.

[15] H. Oliveira, I. Katib, N. da Fonseca, and D. Medhi, “Comparison of network protection in three-layer IP/MPLS-over-OTN-over-DWDM networks,” in *IEEE Global Communications Conference (GLOBECOM)*, Dec. 2015.

[16] P. Gill, N. Jain, and N. Nagappan, “Understanding network failures in data centers: Measurement, analysis, and implications,” in *ACM SIGCOMM Conference*, 2011.

[17] DC Dynamics, “Google cloud US-EAST1 data centers disrupted due to physical damage to multiple fiber bundles,” *Technical Report*, 2019.

[18] BBC, “Google goes offline after fibre cables cut,” *Technical Report*, 2019.

[19] A. Hmaity, F. Musumeci, and M. Tornatore, “Survivable virtual network mapping to provide content connectivity against double-link failures,” in *Design of Reliable Communication Networks Conference (DRCN)*, 2016.

[20] X. Li, S. Huang, S. Yin, B. Guo, Y. Zhao, J. Zhang, and W. Gu, “Shared end-to-content backup path protection in k-node (edge) content connected elastic optical datacenter networks,” *Optics Express*, vol. 24, no. 9, pp. 9446–9464, May 2016.

[21] X. Li, S. Yin, X. Wang, Y. Zhou, Y. Zhao, S. Huang, and J. Zhang, “Content placement with maximum number of end-to-content paths in k-node (edge) content connected optical datacenter networks,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 1, pp. 53–66, Jan. 2017.

[22] X. Li, T. Gao, L. Zhang, Y. Tang, Y. Zhang, and S. Huang, “Survivable k-node (edge) content connected virtual optical network (KC-VON) embedding over elastic optical data center networks,” *IEEE Access*, vol. 6, pp. 38 780–38 793, 2018.

[23] S. Ferdousi, F. Dikbiyik, M. F. Habib, M. Tornatore, and B. Mukherjee, “Disaster-aware datacenter placement and dynamic content management in cloud networks,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 7, pp. 681–694, July 2015.

[24] R. Diestel, *Graph Theory*, 4th ed., ser. Graduate Texts in Mathematics. Springer, 2012.

[25] L. Peterson, A. Al-Shabibi, T. Anshutz, S. Baker, A. Bavier, S. Das, J. Hart, G. Palukar, and W. Snow, “Central office re-architected as a data center,” *IEEE Communications Magazine*, vol. 54, no. 10, pp. 96–101, October 2016.

[26] L. Askari, F. Musumeci, and M. Tornatore, "Latency-aware traffic grooming for dynamic service chaining in metro networks," in *Proc. of IEEE ICC*, May 2019.

[27] T. Tachibana, Y. Hirota, K. Suzuki, T. Tsuritani, and H. Hasegawa, "Construction algorithm of metropolitan area network based on regional characteristics in Japan: A case of Tokyo metropolitan area," *IEICE Technical Report*, 2019.

[28] M. Rahnamay-Naeini, J. E. Pezoa, G. Azar, N. Ghani, and M. M. Hayat, "Modeling stochastic correlated failures and their effects on network reliability," in *Int. Conference on Computer Commununications and Networks (ICCCN)*, July 2011.

[29] Japan Government Headquarters for Earthquake Research Promotion, "Japan earthquake forecast map," *Technical Report*, 2018.

[30] R. Nagai, T. Takabatake, M. Esteban, H. Ishii, and T. Shibayama, "Tsunami risk hazard in Tokyo Bay: The challenge of future sea level rise," *International Journal of Disaster Risk Reduction*, vol. 45, 2020.

[31] M. Clouqueur and W. D. Grover, "Availability analysis of span-restorable mesh networks," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 4, pp. 810–821, 2002.

[32] Metro-Haul, "Definition of use cases, service requirements and KPIs," *Deliverable D 2.1*, 2018.

[33] C. Natalino, A. de Sousa, L. Wosinska, and M. Furdek, "Content placement in 5G-enabled edge/core datacenter networks resilient to link cut attacks," *Wiley Networks Journal*, 2020.

[34] Qualcomm, "Augmented and virtual reality: The first wave of 5G killer apps," *White Paper*, 2017.

[35] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[36] W. Zhang, J. Tang, C. Wang, and S. de Soysa, "Reliable adaptive multipath provisioning with bandwidth and differential delay constraints," in *Proc. of IEEE INFOCOM*, 2010.

[37] C. S. K. Vadrevu, R. Wang, M. Tornatore, C. U. Martel, and B. Mukherjee, "Degraded service provisioning in mixed-line-rate WDM backbone networks using multipath routing," *IEEE/ACM Transactions on Networking*, vol. 22, no. 3, pp. 840–849, 2014.

[38] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: Measurement, analysis, and implications," in *Proc. of ACM SIGCOMM*, 2011.

[39] J. Zhang, K. Zhu, and B. Mukherjee, "Backup reprovisioning to remedy the effect of multiple link failures in WDM mesh networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 57–67, 2006.

[40] L. Ruan and N. Xiao, "Survivable multipath routing and spectrum allocation in OFDM-based flexible optical networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 3, pp. 172–182, 2013.

[41] L. Ruan and Y. Zheng, "Dynamic survivable multipath routing and spectrum allocation in OFDM-based flexible optical networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 6, no. 1, pp. 77–85, 2014.

[42] S. Huang, C. U. Martel, and B. Mukherjee, "Survivable multipath provisioning with differential delay constraint in telecom mesh networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 657–669, 2011.

[43] N. Charbonneau and V. M. Vokkarane, "Routing and wavelength assignment of static manycast demands over all-optical wavelength-routed WDM networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 2, no. 7, pp. 442–455, 2010.

[44] L. Choy, "Virtual concatenation tutorial: enhancing SONET/SDH networks for data transport," *Journal of Optical Networking*, vol. 1, no. 1, pp. 18–29, Jan. 2002.

[45] G. Bernstein, D. Caviglia, R. Rabbat, and H. Van Helvoort, "VCAT-LCAS in a clamshell," *IEEE Communications Magazine*, vol. 44, no. 5, pp. 34–36, 2006.

[46] Y. Sone, A. Watanabe, W. Imajuku, Y. Tsukishima, B. Kozicki, H. Takara, and M. Jinno, "Highly survivable restoration scheme employing optical bandwidth squeezing in spectrum-sliced elastic optical path (slice) network," in *Proc. of IEEE/OSA OFC*, 2009.

[47] X. Li, S. Yin, X. Wang, Y. Zhou, Y. Zhao, S. Huang, and J. Zhang, "Content placement with maximum number of end-to-content paths in K-node (edge) content connected optical datacenter networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 9, no. 1, pp. 53–66, 2017.

[48] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *Journal of Lightwave Technology*, vol. 31, no. 1, pp. 15–22, 2013.

[49] S. S. Savas, F. Dikbiyik, M. F. Habib, M. Tornatore, and B. Mukherjee, "Disaster-aware service provisioning with manycasting in cloud networks," *Photonic Network Communications*, vol. 28, no. 2, p. 123–134, 2014.

[50] R. Goścień, K. Walkowiak, and M. Tornatore, "Survivable multipath routing of anycast and unicast traffic in elastic optical networks," *Journal of Optical Communications and Networking*, vol. 8, no. 6, pp. 343–355, June 2016.

[51] P. M. Moura and N. L. S. da Fonseca, “Multipath routing in elastic optical networks with space-division multiplexing,” *IEEE Communications Magazine*, vol. 59, no. 10, pp. 64–69, 2021.

[52] L. R. Ford and D. R. Fulkerson, “Maximal flow through a network,” *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956.

[53] J. Kleinberg and E. Tardos, *Algorithm Design.* Pearson Education Inc., 2006.

[54] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[55] Metro-Haul, “Selection of metro node architectures and optical technology options,” *Deliverable D 3.1*, 2018.

[56] Qualcomm, “Augmented and virtual reality: The first wave of 5G killer apps,” *White Paper*, 2017.

[57] D. Chatzopoulos, C. Bermejo, Z. Huang, and P. Hui, “Mobile augmented reality survey: From where we are to where we go,” *IEEE Access*, vol. 5, pp. 6917–6950, 2017.

[58] J. Park, S. Samarakoon, H. Shiri, and M. K. Abdel-Aziz, “Extreme ultra-reliable and low-latency communication,” *Nature Electronics*, vol. 5, pp. 133–141, 2022.

[59] B. M. Maggs and R. K. Sitaraman, “Algorithmic nuggets in content delivery,” *ACM SIGCOMM Computer Communication Review*, vol. 45, pp. 52–66, 2015.

[60] L. Kontothanassis, R. Sitaraman, J. Wein, D. Hong, R. Kleinberg, B. Mancuso, D. Shaw, and D. Stodolsky, “A transport layer for live streaming in a content delivery network,” *Proc. of the IEEE*, vol. 92, no. 9, pp. 1408–1419, 2004.

[61] M. Pathan, R. K. Sitaraman, and D. Robinson, “Advanced content delivery, streaming, and cloud services: Overlay networks - an Akamai perspective,” *Wiley-IEEE Press*, pp. 305–328, 2014.

[62] M. Tornatore, “The challenges of end-to-end network resilience,” in *Proc. of ECOC*, 2021.

[63] L. Gifre, F. Paolucci, O. G. de Dios, L. Velasco, L. M. Contreras, F. Cugini, P. Castoldi, and V. López, “Experimental assessment of abno-driven multicast connectivity in flexgrid networks,” *Journal of Lightwave Technology*, vol. 33, no. 8, pp. 1549–1556, 2015.

[64] Qualcomm, “VR and AR pushing connectivity limits,” *White Paper*, 2018.

[65] S. Mangiante, G. Klas, A. Navon, Z. GuanHua, J. Ran, and M. D. Silva, “VR is on the edge: How to deliver 360° videos in mobile networks,” in *Proc. of the Workshop on Virtual Reality and Augmented Reality Network*, 2017.

[66] Metro-Haul, “Functional architecture specifications and functional definition,” *Deliverable D 2.2*, 2018.

[67] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, “A survey on low latency towards 5G: RAN, core network and caching solutions,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3098–3130, 2018.

[68] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. A. Ashraf, B. Almeroth, J. Voigt, I. Riedel, A. Puschmann, A. Mitschele-Thiel, M. Muller, T. Elste, and M. Windisch, “Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture,” *IEEE Communications Magazine*, vol. 55, no. 2, pp. 70–78, 2017.

[69] R. Bhandari, “Optimal physical diversity algorithms and survivable networks,” in *Proc. of IEEE Symposium on Computer and Communications*, 1997.

[70] S. Arakawa, T. Sakano, Y. Tsukishima, H. Hasegawa, T. Tsuritani, Y. Hirota, and H. Tode, “Topological characteristic of Japan photonic network model,” *IEICE Technical Report*, 2013.

[71] K. Sripanidkulchai, B. Maggs, and H. Zhang, “An analysis of live streaming workloads on the Internet,” in *Proc. of the 4th ACM SIGCOMM Conference on Internet Measurement*, 2004.

[72] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html

[73] X. Chen, B. Li, R. Proietti, H. Lu, Z. Zhu, and S. J. B. Yoo, “DeepRMSA: A deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks,” *Journal of Lightwave Technology*, vol. 37, no. 16, pp. 4155–4163, 2019.