

RESILIENCE AND SURVIVABILITY OF 5G NETWORKS

A Dissertation
IN
Computer Networks & Communication Systems
And
Electrical Engineering

Presented to the Faculty of the University
of Missouri–Kansas City in partial fulfillment of
the requirements for the degree

DOCTOR OF PHILOSOPHY

by
ROHIT ABHISHEK

M. S, University of Missouri-Kansas City, USA, 2014

Kansas City, Missouri
2020

© 2020

ROHIT ABHISHEK

ALL RIGHTS RESERVED

RESILIENCE AND SURVIVABILITY OF 5G NETWORKS

Rohit Abhishek, Candidate for the Doctor of Philosophy Degree

University of Missouri–Kansas City, 2020

ABSTRACT

5G is going to be the central force behind the Fourth Industrial Revolution. It is the next-generation wireless technology which is slated to provide a wide range of services. It is geared to provide greater capacity, increased energy efficiency, and lower latency. A critical issue in service delivery is to provide resilience in 5G networks.

In this thesis, we present 5G network architecture with network virtualization with multiple providers for network resilience that uses a self-organizing ad hoc network among the gNBs (macrosites). Thus, the primary provider for a 5G network may use a secondary provider for network resilience when network components fail. We present an optimization formulation and a heuristic for network survivability for our proposed 5G network for the primary network provider. Through simulations, we show our proposed heuristic is very close to optimal. The simulation results on the trade-off between using a provider's own network or rely on auxiliary capacity from another provider allow us to see the trade-off on availability.

We also envision an environment where 5G network resilience is addressed in the presence of unlicensed spectrum and non-terrestrial networks. In this prospect, we present a framework for network survivability with network virtualization with multiple providers, and the use of unlicensed spectrum band and non-terrestrial network (NTN); this is done along with a self-organizing ad hoc network among the gNBs that may use a secondary provider for network resilience when the aggregation network and the backhaul network fails. In this architecture, we present an optimization model for survivability for a 5G networks provider (primary provider) that may also use a secondary provider in the event of a failure along with unlicensed spectrum and NTN. Our simulations show (1) the trade-off between using a primary provider's own network or rely on auxiliary capacity from the secondary provider, and (2) the use of unlicensed spectrum band and NTN enhances the resilience of the network.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Graduate Studies, have examined a dissertation titled “Resilience and Survivability of 5G Networks,” presented by Rohit Abhishek, candidate for the Doctor of Philosophy degree, and certify that in their opinion it is worthy of acceptance.

Supervisory Committee

Deep Medhi, Ph.D., Committee Chair
Department of Computer Science & Electrical Engineering

Baek-Yong Choi, Ph.D.
Department of Computer Science & Electrical Engineering

Sejun Song, Ph.D.
Department of Computer Science & Electrical Engineering

Ghulam Chaudhry, Ph.D.
Department of Computer Science & Electrical Engineering

Masud Chowdhury, Ph.D.
Department of Computer Science & Electrical Engineering

CONTENTS

ABSTRACT	iii
ILLUSTRATIONS	viii
TABLES	xi
ACKNOWLEDGEMENTS	xiv
Chapter	
1 INTRODUCTION	1
1.1 Motivation And Scope	3
1.2 Contributions	4
1.3 Additional Contributions	7
1.4 Organization	17
2 Research Survey	18
3 5G Architecture for Resilience	20
4 Proposed Optimization and Heuristic for 5G Network Resilience	25
4.1 Optimization Problem formulation for Network Resilience	25
4.2 Simulation Setup and Results	28
4.3 A Heuristic for Network Resilience	38
4.4 Simulation Setup and Results	39
5 Unlicensed Spectrum band and Non-Terrestrial Network	72
5.1 Unlicensed Spectrum Band	72

5.2	Non- Terrestrial Network	74
6	An Integrated 5G Architecture for Survivability	75
6.1	An Integrated 5G Architecture for Survivability	76
6.2	Optimization Model	80
6.3	Simulation Setup and Results	83
7	Conclusion	93
Appendix		
A	Optimization model file	96
B	Heuristic code	100
C	Topology Generation	105
REFERENCE LIST		130
VITA		134

ILLUSTRATIONS

Figure		Page
1	SPArTaCuS: Architecture Framework	7
2	Modelling Smart Cities networks in SPArTaCuS	8
3	BuDDI $N + 2$ Reliability	9
4	BuDDI Middlebox Architecture.	10
5	Home Architecture	12
6	Architecture Framework	14
7	Framework of proposed implementation	16
8	Proposed 5G	21
9	5G high level architecture	22
10	Aggregation Router Failure	23
11	Topology 1	29
12	Topology 1:Aggregation Router A failure	31
13	Topology 1:Aggregation Router B failure	31
14	Topology 1:Aggregation Router C failure	32
15	Topology 1:Aggregation Router D failure	32
16	Topology 2	35
17	Topology 2: Simulation Result	36
18	3x3 grid topology	40

19	Non-grid topology	41
20	Legends: 3x3 grid topology and Non-grid topology	48
21	3x3 grid topology: AR A failure	48
22	3x3 grid topology: AR B failure	49
23	3x3 grid topology: AR C failure	49
24	3x3 grid topology: AR E failure	50
25	3x3 grid topology: AR F failure	50
26	3x3 grid topology: AR G failure	51
27	3x3 grid topology: AR H failure	51
28	Non-grid topology: AR A failure	60
29	Non-grid topology: AR B failure	60
30	Non-grid topology: AR C failure	61
31	Non-grid topology: AR E failure	61
32	Non-grid topology: AR F failure	62
33	Non-grid topology: AR G failure	62
34	Non-grid topology: AR H failure	63
35	Use of Non-Terrestrial Network with 5G	74
36	Extended 5G high level architecture	76
37	Proposed Resilient Framework	77
38	NTN integrated simulation topology	85
39	NTN + US: Aggregation Router A failure	86
40	NTN + US: Aggregation Router B failure	87

41	NTN + US: Aggregation Router C failure	87
42	NTN + US: Aggregation Router E failure	88

TABLES

Tables		Page
1	Model Notations: 5G Network Resilience	26
2	Simulation Values: Topology 1	30
3	Abbreviations: Topology 1 and Topology 2	30
4	Topology 1: Percentage of traffic satisfied when AR A fails	33
5	Topology 1:Percentage of traffic satisfied when AR B fails	33
6	Topology 1:Percentage of traffic satisfied when AR C fails	34
7	Topology 1:Percentage of traffic satisfied when AR E fails	34
8	Topology 2:Percentage of traffic satisfied when AR B and AR C fails (one failure at a time)	37
9	Topology 2: Percentage of traffic satisfied when AR D and AR E fails (one failure at a time)	37
10	Simulation Values: 3x3 grid topology	42
11	AR A failure: Model vs Heuristic	43
12	AR B failure: Model vs Heuristic	43
13	AR C failure: Model vs Heuristic	44
14	AR E failure: Model vs Heuristic	44
15	AR F failure: Model vs Heuristic	45
16	AR G failure: Model vs Heuristic	45

17	AR H failure: Model vs Heuristic	46
18	Model vs Heuristic Analysis	46
19	3x3 grid topology: Traffic satisfied and cost incurred when AR A fails . .	52
20	3x3 grid topology: Traffic satisfied and cost incurred when AR B fails . .	53
21	3x3 grid topology: Traffic satisfied and cost incurred when AR C fails . .	54
22	3x3 grid topology: Traffic satisfied and cost incurred when AR E fails . .	55
23	3x3 grid topology: Traffic satisfied and cost incurred when AR F fails . .	56
24	3x3 grid topology: Traffic satisfied and cost incurred when AR G fails . .	57
25	3x3 grid topology: Traffic satisfied and cost incurred when AR H fails . .	58
26	Simulation Values: Non-grid topology	59
27	Non-grid topology: Traffic satisfied and cost incurred when AR A fails . .	63
28	Non-grid topology: Traffic satisfied and cost incurred when AR B fails . .	64
29	Non-grid topology: Traffic satisfied and cost incurred when AR C fails . .	65
30	Non-grid topology: Traffic satisfied and cost incurred when AR D fails . .	66
31	Non-grid topology: Traffic satisfied and cost incurred when AR E fails . .	67
32	Non-grid topology: Traffic satisfied and cost incurred when AR F fails . .	68
33	Non-grid topology: Traffic satisfied and cost incurred when AR G fails . .	69
34	Non-grid topology: Traffic satisfied and cost incurred when AR H fails . .	70
35	Model Notations: 5G Network Resilience in presence of NTN and US . .	84
36	ABBREVIATIONS: NTN + US	89
37	Simulation Values: NTN + US	89
38	NTN + US: AR A failure	90

39	NTN + US: AR B failure	90
40	NTN + US: AR C failure	91
41	NTN + US: AR E failure	91

“Arthur Compton became my graduate advisor. He was the ideal graduate advisor for me: he came into my research room only once during my graduate career and usually had no idea how I was spending my time”

Luis Walter Alvarez

ACKNOWLEDGEMENTS

“The output of your Ph.D. is not your thesis. It’s You!” was the advice Dr. Deep Medhi gave to me when I started my Ph.D. under his supervision. It is always an honor for any student to have an advisor like Dr. Medhi. He has been a constant pillar of support during my Ph.D. study. I want to thank him for his patience, motivation, immense knowledge, and his unique sense of humor. His guidance helped me in all the time of research and writing of this thesis. He has taught me, both consciously and unconsciously, how good research is done. I am thankful for his time, ideas, and funding, which made my Ph.D. experience productive and stimulating. His joy and enthusiasm for research are contagious and were consistently a motivational force for me through the ups and downs of my Ph.D. time. He has set an excellent example for me to be a good researcher and, above all, a great human being. I could not have imagined having a better advisor and mentor for my Ph.D. study.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Baek-Yong Choi, Dr. Sejun Song, Dr. Masud Chowdhury, and Dr. Ghulam Chaudhry, for their insightful comments and encouragement. My sincere thanks go to Dr. David Tipper from the University of Pittsburgh for all his support and guidance in my research. I would also like to thank Dr. Abhishek Roy from Mediatek and Dr. Stephan Wenger from Tencent America for their mentorship during my internships.

Special thanks to my mom Rani Singh and dad Yogendra Narayan Singh for all emotional and financial support over the years. I would like to thank my friends Anurag

Thantharate, Poonam Kankariya, Priyanka Gaikward, Rahul Paropkari, Saravanan Anbazhagan and Vijay Walunj for supporting me spiritually throughout writing this thesis and my life in general. Dr. Ajita Rattani, Dr. Devisha Varma and Dr. Shuai Zhao have always been a big motivational force behind me. Lastly, I wouldn't have survived writing this thesis without the companionship of my dog Molly.

This work is partially supported by NSF Grant # 1526299.

CHAPTER 1

INTRODUCTION

5G networks are expected to be 10-100 times faster with increased energy efficiency and much lower latency than the current 4G LTE networks. Furthermore, 5G is expected to vastly improve support for emerging mobility services. It will use the New Radio (NR) and is expected to be integrated along with the existing 4G LTE architecture and is supposed to use unlicensed frequencies such as 3.5 GHz spectrum besides low spectrum frequencies such as 600 MHz. The key components for the future 5G will be millimeter waves, Network function virtualization (NFV) and Software Defined Networking (SDN). By 2020 the mobile data traffic will explode to more than 200-fold from 2010 [1]. Features like carrier aggregation and self-organizing networks (SON) are expected to be the key factor in the next generation wireless technology. To provide increased data capacity, 5G is expected to use small cells, which will be the key functionality of 5G networks along with key factor features like carries aggregation and self-organizing networks (SON).

In general, the 5G network architecture will enable the creation of mobile networks on cloud-based virtualization platforms and will offer different cloud-based services. Software-defined networking (SDN) will be used in the 5G architecture to slice the sub-networks, which can be used for higher bandwidth requirements [2]. This would help maintain the CAPEX (Capital Expenditure) and OPEX (Operational Expenditure) while reducing latency and making it more energy-efficient for the exploding traffic [3].

For 5G networks, network functions will be software-defined including the core network functions such as MME (Mobile Management Entity), P-GW (Packet Gateway), S-GW (Serving Gateway) and will run as virtualized network functions (VNF) in the cloud, making the network infrastructure more flexible, manageable, and energy-efficient. This would enable the evolved packet core (EPC) to be used/offered as a service (EPCaaS) [4]. The software-defined network controller would automatically allocate resources to the slices to adapt to the demands for different slices.

With exponential growth in wireless data traffic and demand for faster network, network resilience will continue to be an important factor in emerging 5G networks. The issue of network resilience is tied to the 5G network architecture as well. Ensuring high reliability and availability in 5G networks requires ensuring resilient cloud services, which can be affected by the failure of any VNFs for 5G networks running on the VMs in the cloud. A VNF failure in the application layer can have a major impact in the network performance and can result in a network blackout. Thus, network redundancy must be provided for network resilience at minimal cost.

1.1 Motivation And Scope

5G has the potential to transform the lives of the people around the world by supporting millions of devices with its ultrafast speed and more capacity. The mission-critical services affecting the safety and security of the services will be greatly improved with the help of 5G. These include opportunities in smart cities, remote surgery, traffic control, and delay-sensitive traffic [5]. In 5G architecture, the core network functions such as MME (Mobile Management Entity), S-GW, P-GW will run as virtualized network functions (VNF) in the cloud, making the network infrastructure more manageable, flexible, and energy-efficient. However, the biggest challenge is to provide network resilience. Any VNF failure in the application layer can have a significant impact on network performance and can result in a network blackout.

In this regard, our research focuses on providing network resilience and survivability. We present the next-generation 5G architecture and a self-healing ad hoc network among the gNBs (macrosites) when any failure in the backhaul (aggregation network) affects the fronthaul traffic. An optimization-based survivability framework with a partnership with a secondary provider has been presented. A heuristic for large scale network where optimization model is time-consuming has been presented. We then extend our research to consider US and NTN for a more complete network resilience environment.

1.2 Contributions

In this work, we propose a self organizing ad hoc network among gNBs (macrosites) in 5G networks that may use another provider, when aggregation network (AN) fails. Here, when the AN fails, the eNB would form a proactive ad hoc network with its neighboring eNBs, potentially from another provider, to meet the failed demands. We present an optimization based survivability framework with partnership established with secondary providers. Through our approach, the overall capacity of the network would be minimally affected by a failure. Our proposed framework also works for an AN link failure and SDN controller failure.

In the first part of this thesis, we consider 5G services in the presence of a secondary provider in which we refer to the original provider as the primary provider. We assume that there is a business relationship between the primary and the secondary provider. When an aggregation network (AN) fails in the primary provider, we propose to activate a self-organizing ad hoc network among gNBs (macrosites) using an unlicensed spectrum between the two providers to provide continuity of services. In this framework, we present our architecture for resilience and then present an optimization formulation to consider reallocation of flows between the primary and the secondary provider when there is a failure. We also present a heuristic to solve this model, followed by a study. Our proposed work has the following notable contributions:

- We present an architecture for 5G networks for resilience.

- We present an optimization-based survivability framework with partnership established between primary and secondary providers. Our proposed framework works for an AN link failure and SDN controller failure.
- Obtaining solutions for large scale problems using optimization tools like CPLEX [6] is difficult as the number of variables and constraints increases with the increase in the number of nodes in the network. Therefore, we present a heuristic.
- We present a study, especially with a quality-of-service based analysis which can be used for prioritization of emergency services during failures.

The current licensed spectrum band used by the operators for different services could be limiting with growing traffic demand and data-intensive applications. With the vast availability of unlicensed spectrum band available, it can be used to complement network capacity either in aggregation with the licensed spectrum band (LAA) or as standalone. Therefore, the unlicensed spectrum can work as a critical factor in providing resilience and survivability for future networks.

In the second part of this thesis, we consider 5G networks in the presence of unlicensed spectrum (US) and non-terrestrial networks (NTN). While 5G networks will have faster speed and lower latency, we envision that due to its ability to provide network virtualization and coordination with US and NTN, superior services would be possible to provide. In particular, our work considers network resilience for 5G to address for failures that can benefit from having access to US and NTN. US and NTNs will make the 5G network more complete and resilient. For this unified environment, we present

our architectural framework. Here, in case of an aggregation network (AN) failure, (1) the gNB can form a proactive ad hoc network with its neighboring gNBs from same as well as a secondary provider, (2) the UEs (user equipments) belonging to the failed aggregation router (AR) can use unlicensed spectrum band besides the licensed spectrum band to connect to the neighboring gNBs and, (3) the UEs can use NTN to route the failed demands. An optimization based survivability framework with partnership established with secondary providers in the presence of US and NTNs has been presented.

1.3 Additional Contributions

1.3.1 SPArTaCuS: Service Priority Adaptiveness for Emergency Traffic in Smart Cities using Software-defined networking

In this work, we propose SPArTaCuS, a framework for smart cities on how to prioritize services for emergency needs in a stressed situation. Our proposed approach uses SDN to accomplish service prioritization for emergency services in a stressed situation. In particular, it uses the SDN framework with OVX to create virtual SDN networks for different service classes that are mapped to the physical infrastructure. Fig. 1 presents a high-level view of the SPArTaCuS framework. In our approach, the middlebox layer has a priority management layer on top of OVX; that is connected to multiple SDN controllers on the northbound interface.

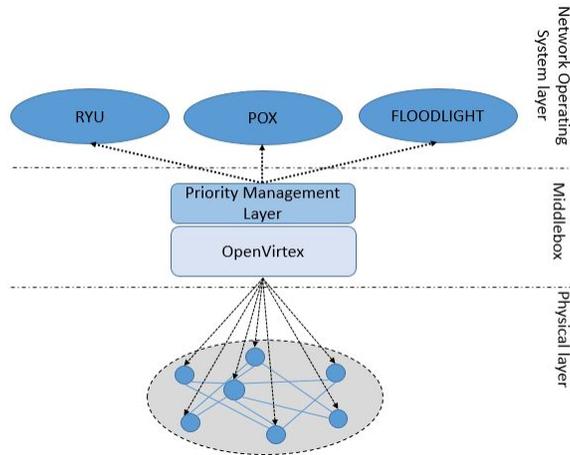


Figure 1: SPArTaCuS: Architecture Framework

Fig. 2 shows internal architecture with virtual networks in SPArTaCuS . The middlebox layer is used to create the VNs and provide priority to them. The OVX creates

the VNs whereas prioritization is done by the priority management layer. In the middlebox, virtual networks are created for different entities or organizations. For example, traffics for government communication can be directed via a specific set of VNs that are responsible for government networks. Similarly, we can classify other virtual networks according to different traffics such as for helpline, general public. We illustrate three VNs in SPArTaCuS shown as VN1, VN2 and VN3; here, VN1 is responsible for government traffic, VN2 for helpline services, and VN3 is categorized for public traffic.

This work was published in IEEE International Smart Cities Conference (ISC2) 2016 [7].

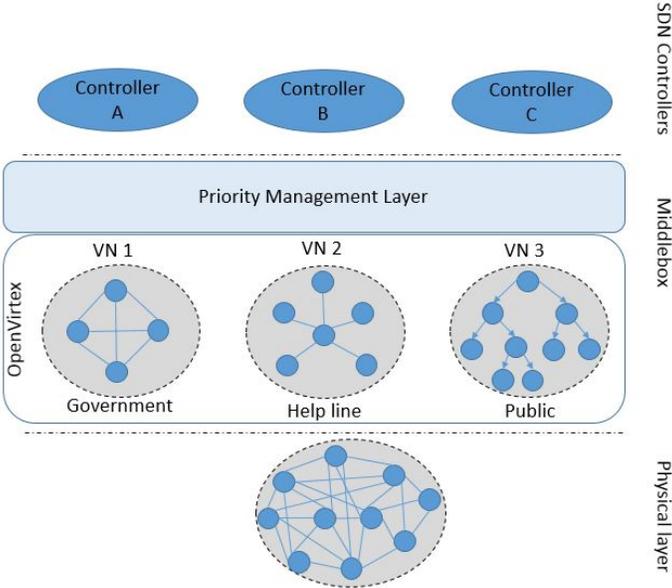


Figure 2: Modelling Smart Cities networks in SPArTaCuS

1.3.2 BuDDI: Bug detection, debugging, and isolation middlebox for software-defined network controllers

In this paper, we propose an online software Bug Detection, Debugging, and Isolation (BuDDI) middlebox architecture for SDN controllers. BuDDI consists of a shadow-controller based online debugging facility and a CMFD mitigation module in support of a seamless heterogeneous controller failover. For on-line bug detection and debugging, unlike a traditional $N + 1$ redundancy cluster system, we propose a $N + 2$ load balancing cluster system where components (N) have at least two independent failover components (+2). As illustrated in Figure 3, BuDDI facilitates a CMFD mitigation module by taking advantage of software diversity of the existing heterogeneous controllers. In addition, BuDDI enables a shadow controller that mirrors the active controller functions and turns on a verbose debugging mode for a specific failure module. Eventually, the two failover components will be converged into one active controller. If the shadow-controller cannot identify a software bug in a given period, it sends a preemption message to the active CMFD module to take over the active role. Otherwise, it will confirm an active role for the CMFD module.

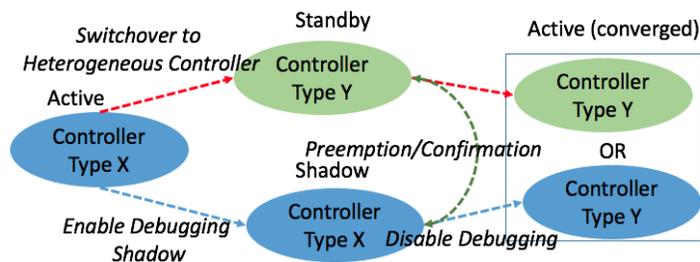


Figure 3: BuDDI $N + 2$ Reliability

Controller switchover algorithms and shadow controllers debugging facilities are built on the top of OpenVirtex, which gives the facility to create virtual network and to map them to the physical network. The middlebox acts as proxy between the physical network and the controllers. As a preliminary part of our experiment, we choose two of the heterogeneous controllers (Ryu and Pox) to verify that both heterogeneous controller switchover and $N+2$ redundancy mechanism supports do not cause any additional performance overhead in the proposed BuDDI mechanism. The proposed architecture is shown in Figure 4. The middlebox is connected to the controllers via northbound OpenFlow whereas with the physical network via southbound OpenFlow.

This work was published in the 12th International Conference on Network and Service Management 2016 [8].

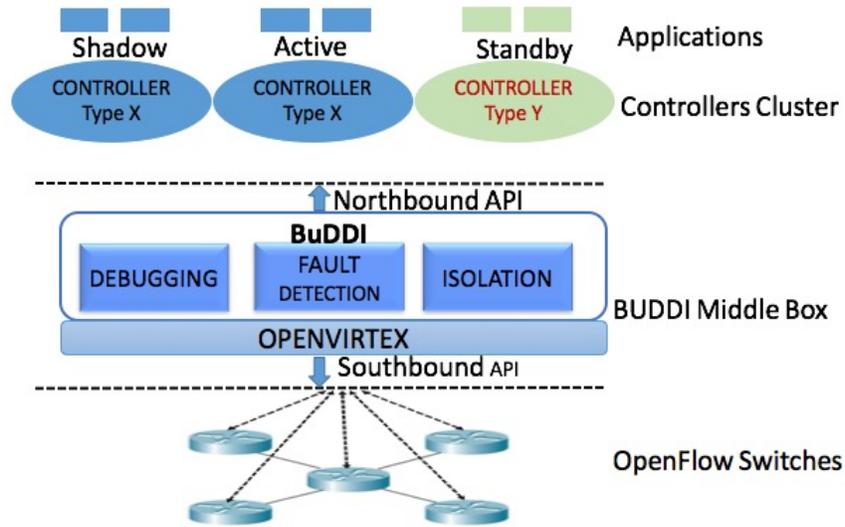


Figure 4: BuDDI Middlebox Architecture.

1.3.3 SeSAMe: Software defined smart home alert management system for smart communities

In this work, we propose SeSAMe as an architectural vision for software defined smart community home alarm management based on software defined networks (SDN). We present the protocol messages and system components for the operation of SeSAMe. With our approach, should any alert/event such as a fire occur, an automated notification is sent to all the homes in the neighborhood and to the fire department and the police department about the fire. At the same time, alerts can also be forwarded to the police and the fire departments.

Fig. 5 shows the high level architecture of a smart home in SeSAMe. It can be categorized into two categories: home gateway and sensors. The sensors include different sensors that are part of the home, e.g., fire sensor, temperature sensor, light sensor, motion sensor, and so on. All sensors send their data to the home gateway. The controller creates a database of the readings from various sensors. As shown in the figure, there are three sensors in the home and the controller creates a database for each of the sensors. The home gateway consists of a database where the reading from different sensors is stored, at least temporarily.

The management layer is the core of a smart home. It continues monitoring the data coming from different sensors. Based on the data collected from the sensors, it creates a triggered event that is sent to the controller along with the type of data and the reading (or a notification that the fire has been detected).

This work was published in IEEE International Symposium on Local and Metropolitan Area Networks 2017 [9].

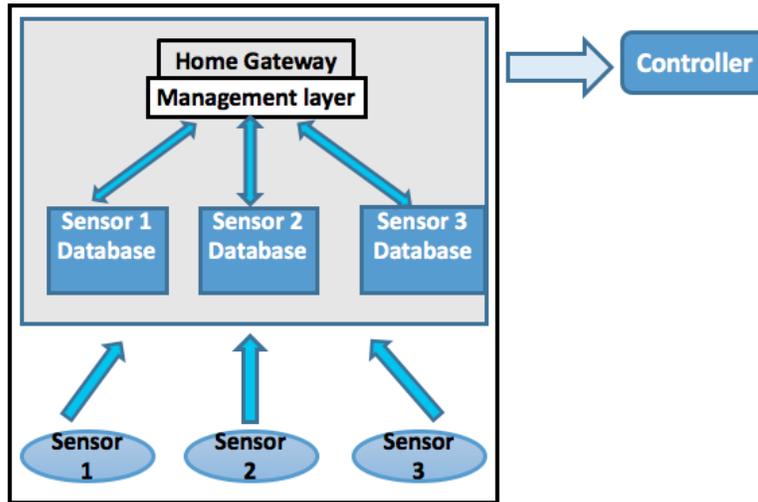


Figure 5: Home Architecture

1.3.4 Disaster recovery power and communications for smart critical infrastructures

In this paper, we propose a framework to address the problem of providing resilient power and ICT to support smart infrastructure applications under natural disaster conditions. Our approach is a combination of a multi-user electrical microgrid to provide power together with cellular based communications dynamically reconfigured into a mesh network and local edge computing resources to support critical smart infrastructure services/applications in a specific geographic area. The goal is to create geographic districts within a city that are safe havens with critical services functioning at a degraded but acceptable level of service in the face of extreme conditions.

We consider a scenario where a natural disaster has resulted in a power outage of size and duration such that commercial cellular networks have outages and the smart infrastructures which depend on a steady supply of electricity as well as cellular communication services are adversely affected. The cellular network outage maybe due to the failures of base stations and/or the backhaul network and/or associated core network services (e.g., authentication, mobility management, synchronization, etc.). Note, that while some cell sites may have backup batteries (typically 4 to 8 hours of power) or diesel gensets, they cannot provide service without backhaul network connectivity and core network services (this was observed in 2012 hurricane Sandy in New Jersey where powered base stations could not provide service due to flooded backhaul equipment resulting in isolation from the core network). Here we propose to use edge computing devices together with dynamic reconfiguring of powered cellular network base stations across operators including pooling the available spectrum to form a multihop ad hoc mesh network which

can provide local disaster communication services to the public, government and smart infrastructures. The components of our framework are illustrated in Figure 6. There are two major pieces: (1) a multi-user microgrid and (2) a disaster recovery cellular based communication network that is organized into a multihop wireless mesh network.

This work was published in IEEE International Conference on Communications 2018 [10].

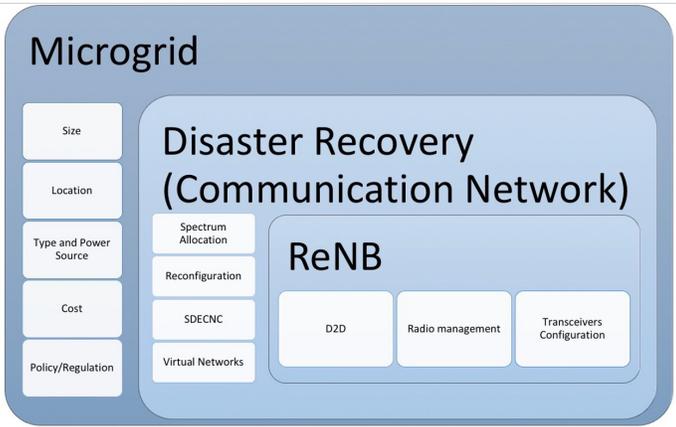


Figure 6: Architecture Framework

1.3.5 Network Optimization for Differentiated QoS Traffic in an SDN Environment for PoP-Data Center Traffic

In this work, we consider a software-defined network (SDN) environment for differentiated QoS traffic classes. An advantage of SDN is that flows or a collection of flows can be controlled. We consider this problem where we assume the different traffic classes to enter through Points-of-Presence (PoPs) of Internet Service Providers (ISPs) to the core network that are to be served from data center (DC) locations that are geographically distributed in the network. We further assume that all data centers are equipped to provide all service requests from any of the data centers. For this problem, we present a novel network optimization formulation that considers differentiated delay requirements for different traffic classes. In our approach, we use a piece-wise linear approximation for non-linear latency, thus allowing us to formulate the problem as a mixed integer linear programming (MILP) problem. our proposed approach can be implemented using the framework as shown in Fig. 7.

This work was published in IEEE Sarnoff Symposium 2018 [11].

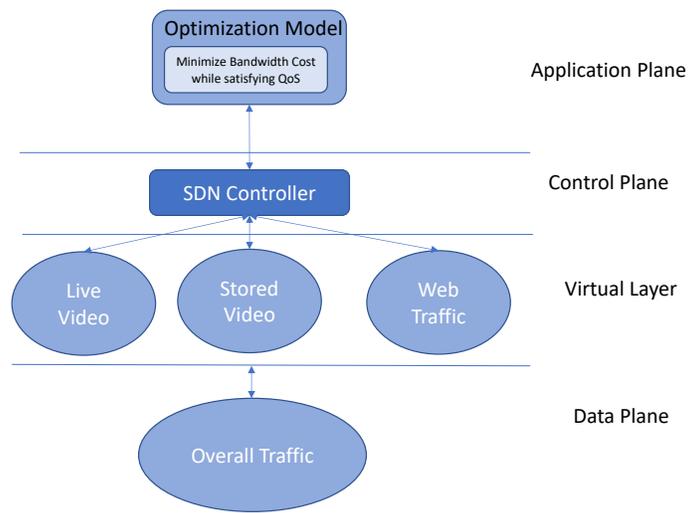


Figure 7: Framework of proposed implementation

1.4 Organization

In chapter 2 we present the research survey. The proposed 5G architecture for resilience has been presented in chapter 3. Chapter 4 presents the optimization formulation for 5G resilience and proposed heuristic. The optimality of the heuristic has been shown by comparing it with the optimization model. In chapter 5 a brief description about the unlicensed spectrum and non-terrestrial network has been shown followed by an integrated 5G architecture for survivability in chapter 6. Finally the conclusion is shown in chapter 7.

CHAPTER 2

RESEARCH SURVEY

Charnsripinyo and Tipper [12] addressed the problem of designing survival 3G wireless packet based backhaul networks. In this work, an optimization model, aimed at finding a wireless backhaul network topology that is able to provide minimum cost survivability with acceptable quality of service, has been proposed for third generation wireless access backhaul networks design by using a mesh topology. This work mainly considered the failure of Node B in 3G network whereas our work focuses on the AN (aggregation network) failure for 5G networks.

Several works has been done on the survivability of the 4G networks. A hybrid iterated local search (ILS) heuristic, named GPP4G-ILS was proposed in [13]. It aimed at solving the global planning problem of survival wireless networks by maximizing its survivability and minimizing the cost. [14] presented a quantitative approach for performance evaluation of network survivability. A framework for the recovery of communication for government, public and critical operations during disasters had been proposed in [7] and [10].

[15] proposed the survivability of an infrastructure based wireless network by utilizing the continuous-time Markov chain, which incorporates the correlated failures

caused by disaster propagation. In [16] a path protection solution for multi-failure network scenarios has been proposed to jointly incorporate traffic engineering and risk minimization objectives. [17] proposed a network resilience evaluation methodology using topology generation, analytical, simulation and experimental emulation techniques. The issue of survivability due to physical attacks has been addressed in [18]. The authors proposed to support critical services in the face of a major attack by minimizing network congestion and maximizing the use of resources. A self healing approach for survivability of an LTE network has been proposed in [19] by using N:M active-standby configuration. However, this approach does not consider virtualization and survivability through a competing provider.

The issue of Mobility Management Entity (MME) failure has been addressed in [20]. The authors proposes a set of solutions to ensure service resiliency in EPS. A proactive MME restoration approach has been proposed in this work. As soon as an MME is offline, the MME relocation for the affected UEs is done. Here the MME relocation and restoration is proactively triggered so as to avoid disruption of service at failure stage. This work focuses on restoring the network during MME failure by selecting a new MME whereas our work focuses on AN failure by creating an adhoc network among the gNBs.

CHAPTER 3

5G ARCHITECTURE FOR RESILIENCE

The next generation 5G network core is expected to be different from the current LTE while it will be based on the present LTE architecture and will be flexible enough to support all lower generations. 5G is expected to be more reliable as compared to the present LTE network and will provide lower latency in the order of few milliseconds. It will have a cloud based core [21] and will be comprised of different NFV/SDN domains, multi-layer control & orchestration, multi-tenancy NFV/SDN, multi-vendor NFV/SDN [22]. Note that the LTE architecture is based on the principles which includes [23]: (1) Common access point and gateway (GW) nodes for all access technologies; (2) Optimized user plane; (3) IP based protocol on all interfaces; (4) Split in the control/user plane between the core network and the gateway; (5) Non-3GPP access technology integration using client-as well as network based IP. We envision that the 5G architecture will utilize the present LTE architecture with NFV/SDN features added to it.

In such a 5G network context, we propose a resilient network framework to achieve network redundancy and load balancing in the network by forming an ad hoc network between the gNBs. At the same time, some of the core network functionalities also would need to be pushed out from the application layer to the gNBs so that security functions and services are available. These functionalities include Policy and Charging Rules Function (PCRF) and HSS functionality for managing the service policy, QoS and performing

authentication and authorization of the user [24]. We present our proposed framework with virtualized mobile functions in Fig. 8. Here the core network functions would run as applications on different VMs or the same VM.

In Fig. 9, we present a general view of our proposed cloud based 5G network architecture when multiple providers that form a partnership are considered as shown using three providers. Each aggregation router (AR) would be mapped to its virtual instance in the virtual layer as shown in the figure. The virtual instance is connected to the SDN controller (control plane), which in turn would be mapped to the application layer where the VNFs run on different virtual machines (VMs). The control plane and the virtual layer are the orchestration layer where multiple providers can coordinate resource and/or service orchestration with each other.

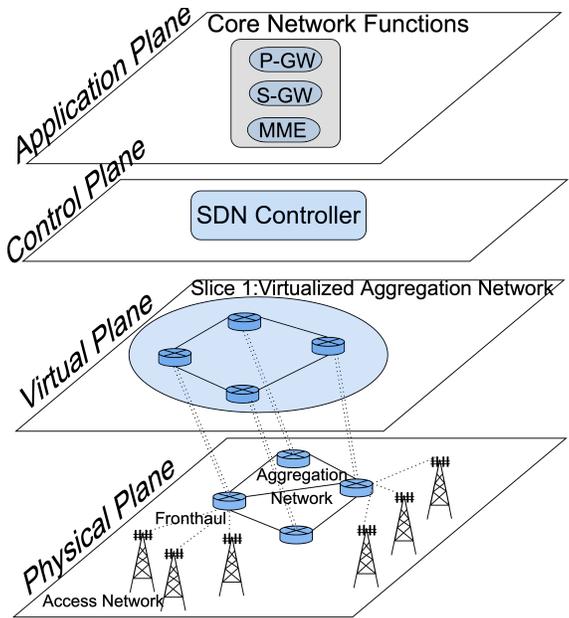


Figure 8: Proposed 5G

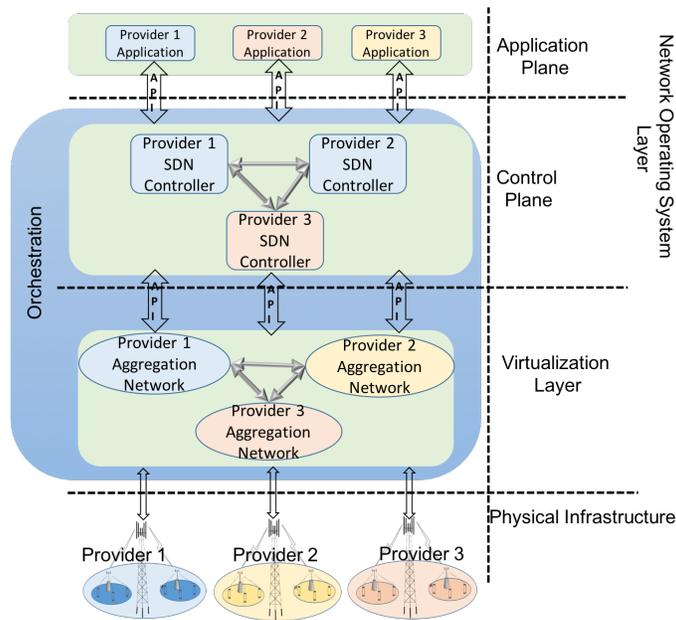


Figure 9: 5G high level architecture

Now consider the scenario where an aggregation router fails or is unreachable due to a physical connectivity failure. The gNBs can form the network with gNBs from the same provider as well as with other providers in the same service area by using unlicensed spectrum band. However, the rerouted traffic should not overload the traffic in a cell sector of another provider of the backup routes within the same provider. So when an aggregation router fails, the gNBs attempt to automatically reroute the traffic using backup paths and provide network redundancy and load balancing.

Fig. 10 represents a failure scenario handling in our framework. For our illustration, we consider Provider-1 as the primary provider and Provider-2 is the secondary provider. Our goal is to provide resilience for the primary provider. We assume that a business agreement exists between these providers. When a aggregation router fails for

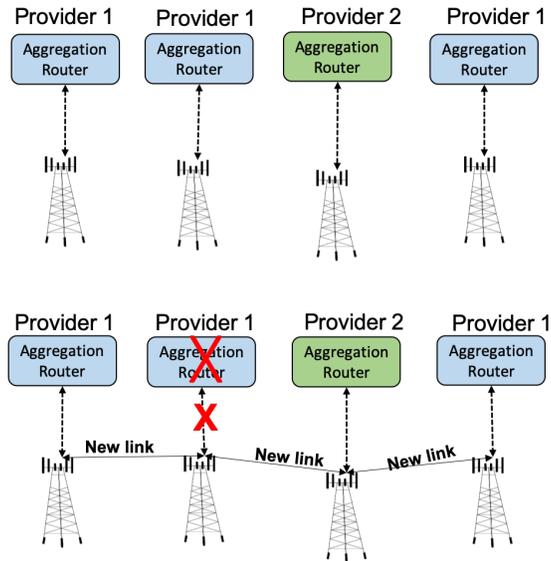


Figure 10: Aggregation Router Failure

the primary provider, the gNBs notifies its neighboring gNBs by sending an announce message via the unlicensed spectrum band. On receiving the announce message, the neighboring gNBs send a reply to the affected gNBs about its available capacity, policies and cost. After the reply has been received, the affected gNBs set up wireless links, by using the unlicensed spectrum band, with the neighboring gNB by sending a hello message. At the same time, if the neighboring gNB belongs to the secondary provider, then it tries to form a link with the nearest gNB belonging to the primary provider (of the failed core) to transmit the demand back into the network. Once the link between the gNBs have been established, the neighboring gNBs from the secondary provider behave simply like a Relay Node to push the demand back into the core network. Note that a gNB belonging to the primary provider through which traffic gets back into the core (destination node) serves as the Donor gNB [25]. The Donor node also provides the S-GW/P-GW

functionalities for the gNBs.

There are two possible options for traffic routing: (1) when all affected traffic is routed using the primary provider, (2) when affected traffic is routed by splitting via the primary provider and a secondary provider. The cost comparison may be a decision factor for choosing one of these options. The steps followed to form the ad hoc network in case of a failure are:

- Each gNB is configured with a backup path, which is calculated based on its distance from the neighboring gNBs (that could be the primary provider or multiple providers).
- Once the aggregation router fails, the gNBs send requests to connect to the neighboring gNBs within the specified radius.
- The neighboring gNBs, based on the available capacity, allocate some capacity to form a wireless link to the gNB requesting the connection.
- After the links between the gNBs are established, some of the backend features of PCRF and HSS are pushed to the gNBs and the links are activated.

CHAPTER 4

PROPOSED OPTIMIZATION AND HEURISTIC FOR 5G NETWORK RESILIENCE

4.1 Optimization Problem formulation for Network Resilience

In this section, we present our optimization model formulation. We address the case when the primary provider has an aggregation router failure. Then the gNBs homed to this router can establish a link to other gNBs (within the range) in its network, or it can establish a link with the secondary provider's gNB(s). Thus, there are two forms of (residual) capacity:

- c_ℓ : link capacity in the primary provider's network
- c'_ℓ : link capacity in the secondary provider's network

It is possible that there is more than one entry point from the secondary provider back to the primary provider's network with the failed aggregation router. However, in this work, we consider only the nearest entry point.

The notations are summarized in Table 1. There are a number of constraints as discussed below. First, the total traffic demand that is carried in the primary provider's network, in the secondary provider's network, and the traffic demand which cannot be carried equals the total traffic demand for each source to destination:

$$x_k + x'_k + w_k = h_k, \quad k \in \mathcal{K} \quad (4.1)$$

Table 1: Model Notations: 5G Network Resilience

<p>Given</p> <p>\mathcal{N} := set of nodes (all types) in the primary provider's network \mathcal{K} := Set of demands identifiers that connects gNBs/ARs between the primary provider's network and all destinations (internal or external) \mathcal{L} := Set of links (all types) in the primary provider's network \mathcal{L}' := Set of links in the secondary provider's network h_k := Traffic demand for identifier k that connect a gNBs/AR to a destination s_k := Source node of identifier k t_k := Destination node of identifier k c_ℓ := Capacity on a link ℓ in the primary provider's network $c'_{\ell'}$:= Capacity on a link ℓ' in the secondary provider's network $\delta_{v\ell}$:= 1 if link ℓ originates at node v; 0, otherwise $\gamma_{v\ell}$:= 1 if link ℓ terminates at node v; 0, otherwise ξ_ℓ := Unit cost on a link ℓ in the primary provider's network $\xi'_{\ell'}$:= Unit cost incurred on a link ℓ' in the secondary provider's network α := Penalty cost incurred for not carrying remaining traffic w_k</p> <p>Variables</p> <p>$x_k (\geq 0)$:= Part of the traffic demand that's carried in the primary provider's network $x'_k (\geq 0)$:= Part of the traffic demand that's carried in the secondary provider's network $w_k (\geq 0)$:= Part of the traffic demand not carried $z_{\ell k} (\geq 0)$:= Link flow on link ℓ for demand identifier k $y_\ell (\geq 0)$:= Link flow on link $\ell \in \mathcal{L}$ (in the primary provider's network) $y'_{\ell'} (\geq 0)$:= Link flow on link $\ell \in \mathcal{L}'$ (in the secondary provider's network)</p>

The allocated traffic flows, x_k , in the primary provider's network are carried on the links in its network:

$$\sum_{\ell \in \mathcal{L}} \delta_{v\ell} z_{\ell k} - \sum_{\ell \in \mathcal{L}} \gamma_{v\ell} z_{\ell k} = \begin{cases} x_k, & \text{if } v = s_k \\ 0, & \text{if } v \neq s_k, t_k, \\ & v \in \mathcal{N} \\ -x_k, & \text{if } v = t_k \end{cases} \quad (4.2)$$

The traffic flows, x'_k , overflowed to the secondary provider's network, are carried on the

links in the secondary provider's network:

$$\sum_{\ell \in \mathcal{L}'} \delta_{v\ell} z_{\ell k} - \sum_{\ell \in \mathcal{L}'} \gamma_{v\ell} z_{\ell k} = \begin{cases} x'_k, & \text{if } v = s_k \\ 0, & \text{if } v \neq s_k, t_k, \\ & v \in \mathcal{N} \\ -x'_k, & \text{if } v = t_k \end{cases} \quad (4.3)$$

Next, (4.4) and (4.5) denote the traffic flows on the links for all demand pairs in each of the primary provider's network and in the secondary provider's network, respectively:

$$\sum_{k \in \mathcal{K}} z_{\ell k} = y_\ell, \quad \ell \in \mathcal{L} \quad (4.4)$$

$$\sum_{k \in \mathcal{K}} z_{\ell k} = y'_\ell, \quad \ell \in \mathcal{L}' \quad (4.5)$$

The link flows in each network must satisfy the capacity available.

$$y_\ell \leq c_\ell, \quad \ell \in \mathcal{L} \quad (4.6)$$

$$y'_\ell \leq c'_\ell, \quad \ell \in \mathcal{L}' \quad (4.7)$$

Note that in the case of the secondary provider, the available capacity means the capacity the secondary provider is willing to share with the primary provider in the case of a failure based on their mutual business agreement.

There are three main goals: to minimize the total link flow cost in the primary provider's network, to minimize the total link flow cost incurred in the secondary provider's network, and also to consider minimizing the total penalty cost incurred for the traffic flow that could not be carried. This is given by:

$$\min_{\{z, y, x\}} \sum_{\ell \in \mathcal{L}} \sum_{k \in \mathcal{K}} \xi_\ell y_\ell + \sum_{\ell \in \mathcal{L}'} \sum_{k \in \mathcal{K}} \xi'_\ell y'_\ell + \alpha \sum_{k \in \mathcal{K}} w_k \quad (4.8)$$

4.2 Simulation Setup and Results

We present results on survivability and the tradeoff between a provider and a partnering provider through two different topologies. Topology 1 considers demand between each of the gNB while in topology 2, we consider video traffic where the demands are between each gNB and the cloud datacenter.

4.2.1 Topology 1

In the first topology, we consider 5 ARs in a 1000×1000 grid. Each AR has 10 gNBs connected to it. The topology is shown in Fig. 11. We considered three scenarios:

- When an AR fails and no partner provider is available to route the traffic
- When an AR fails and one partner provider is available to route the traffic
- When an AR fails and there are two partner providers is available to route the traffic

The location of the AR and the gNB connected to them were randomly generated. The simulation was run for two sets of link capacity values. The radius to connect to the neighboring gNB varied from 0 to 1500 units, where the radius indicates the range within which the affected gNB can form a link with the neighboring gNBs (same provider or multiple provider). The link capacity values used for the simulation are shown in Table 2. In our study, we considered AR A, AR B, AR C and AR D (Fig. 11) failures, one AR failure at a time. From Fig. 11, we can see that each AR failure affects the topology differently, while the radius gives us a metric on the spread of the failure.

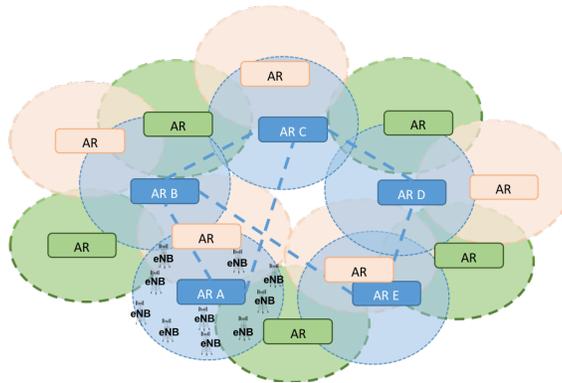


Figure 11: Topology 1

The simulation results are shown in Fig. 12, Fig. 13, Fig. 14, and Fig. 15. The results are also shown in Table 4, Table 5, Table 6, and Table 7, where the values represent the percentage of failed traffic values that are satisfied. The graphs show the percentage of the required capacity, which is met using our proposed architecture. The x-axis shows the percentage of capacity that is met, whereas the y-axis shows the radius for connecting to other macro sites. Table 3 explains the legends used in the graphs. As can be seen from the graphs, more the link capacity is, the more will be the failed demand which would be recovered.

For instance, when AR A fails, all the gNBs connected to it fails. Then the affected gNBs connect to the neighboring gNB from the same provider and from the partner provider to route its traffic. Fig. 12 shows the percentage of failed traffic, which is satisfied using our proposed framework. Initially, when the node fails, we can see from the graph that the percentage of failed demand met is 0. As the radius to connect to the neighboring gNB is increased, the percentage of satisfied traffic increases. Also from the graph we can observe that as the number of partner providers increases, the percentage of

satisfied traffic increases as well. In Table 6 for AR C failure, we can see that the values for capacity 40 and radius 1500 are all same for 0, 1, or 2 providers since AR C has three links connected to it.

Table 2: Simulation Values: Topology 1

Link	Scenario 1		Scenario 2	
	Capacity	Cost	Capacity	Cost
AR-AR	12000	1	5000	1
AR-gNB	2000	1	500	1
gNB -gNB (same provider)	40/80	8	40	8
gNB -gNB (different provider)	40/80	16	40	16

Table 3: Abbreviations: Topology 1 and Topology 2

	#Partner Providers	Wireless link Capacity
0 Provider Cap 40	0	40
1 Provider Cap 40	1	40
2 Provider Cap 40	2	40
0 Provider Cap 80	0	80
1 Provider Cap 80	1	80
2 Provider Cap 80	2	80

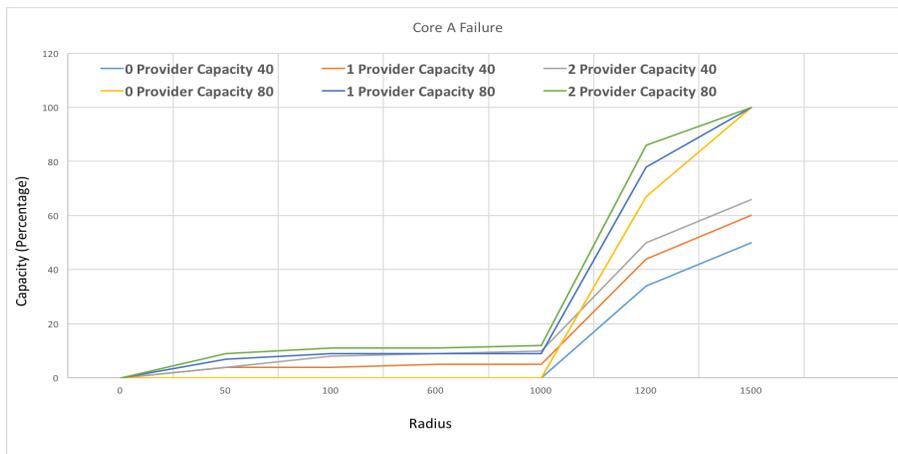


Figure 12: Topology 1:Aggregation Router A failure

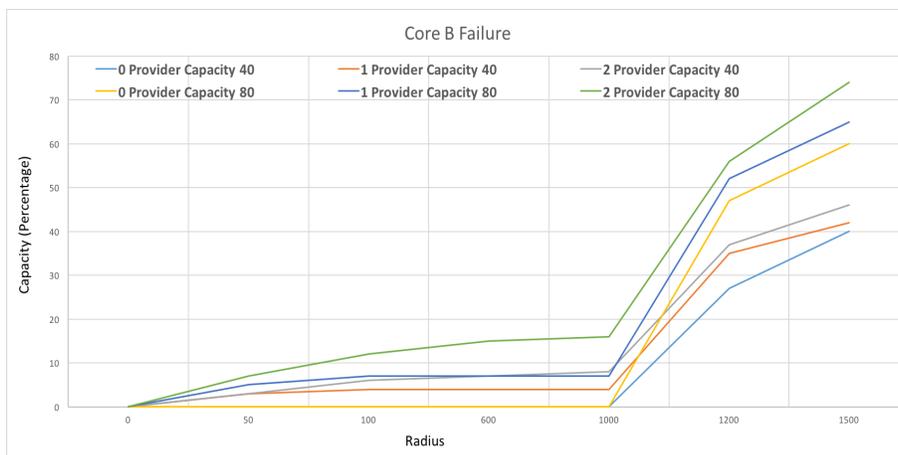


Figure 13: Topology 1:Aggregation Router B failure

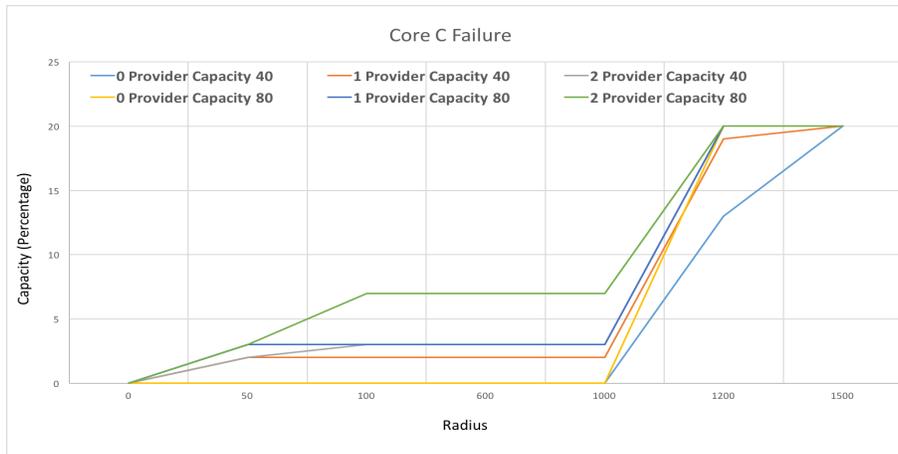


Figure 14: Topology 1:Aggregation Router C failure

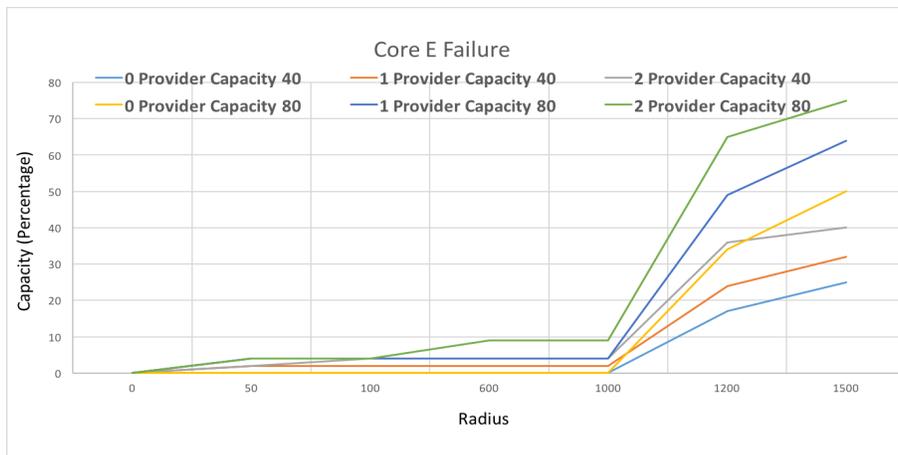


Figure 15: Topology 1:Aggregation Router D failure

Table 4: Topology 1: Percentage of traffic satisfied when AR A fails

Radius	Capacity 40			Capacity 80		
	0 Provider	1 Provider	2 Provider	0 Provider	1 Provider	2 Provider
0	0	0	0	0	0	0
50	0	4	4	0	7	9
100	0	4	8	0	9	11
600	0	5	9	0	9	11
1000	0	5	10	0	9	12
1200	34	44	50	67	78	86
1500	50	60	66	100	100	100

Table 5: Topology 1:Percentage of traffic satisfied when AR B fails

Radius	Capacity 40			Capacity 80		
	0 Provider	1 Provider	2 Provider	0 Provider	1 Provider	2 Provider
0	0	0	0	0	0	0
50	0	3	3	0	5	7
100	0	4	6	0	7	12
600	0	4	7	0	7	15
1000	0	4	8	0	7	16
1200	27	35	37	47	52	56
1500	40	42	46	60	65	74

Table 6: Topology 1:Percentage of traffic satisfied when AR C fails

Radius	Capacity 40			Capacity 80		
	0 Provider	1 Provider	2 Provider	0 Provider	1 Provider	2 Provider
0	0	0	0	0	0	0
50	0	2	2	0	3	3
100	0	2	3	0	3	7
600	0	2	3	0	3	7
1000	0	2	3	0	3	7
1200	13	19	20	20	20	20
1500	20	20	20	20	20	20

Table 7: Topology 1:Percentage of traffic satisfied when AR E fails

Radius	Capacity 40			Capacity 80		
	0 Provider	1 Provider	2 Provider	0 Provider	1 Provider	2 Provider
0	0	0	0	0	0	0
50	0	2	2	0	4	4
100	0	2	4	0	4	4
600	0	2	4	0	4	9
1000	0	2	4	0	4	9
1200	17	24	36	34	49	65
1500	25	32	40	50	64	75

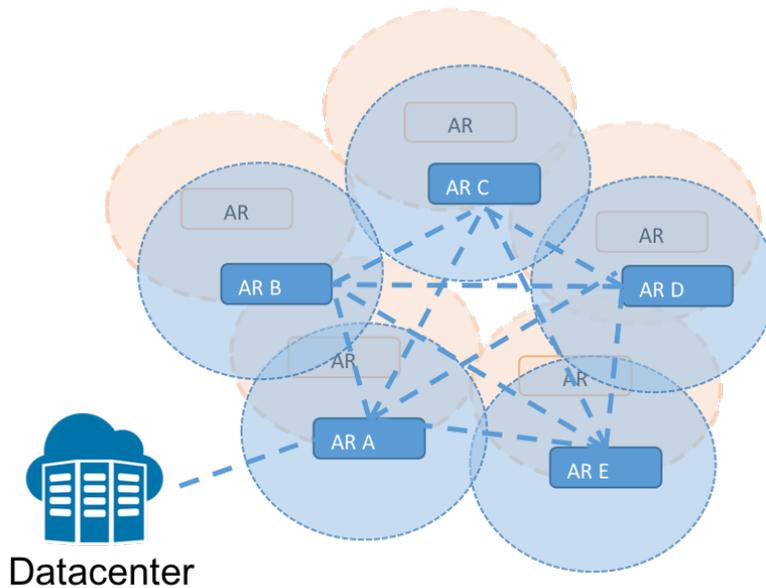


Figure 16: Topology 2

4.2.2 Topology 2

We next consider a cloud data center serving traffic to users. The topology is a full mesh network consisting of five core nodes and a datacenter spread in a 1000X1000 grid. Traffic demand is considered between each of the gNB and the datacenter. We performed the simulation for two different scenarios:

- When an AR fails and no partner provider is available to route the traffic
- When an AR fails and one partner provider is available to route the traffic

The topology is shown in Fig. 16. For this simulation, we have considered the failure of AR B, AR C, AR D and AR E, one failure at a time.

The location of the AR and gNB were randomly generated. The radius for connecting to the neighboring gNB varied from 0 to 600 units. The link capacity used in this

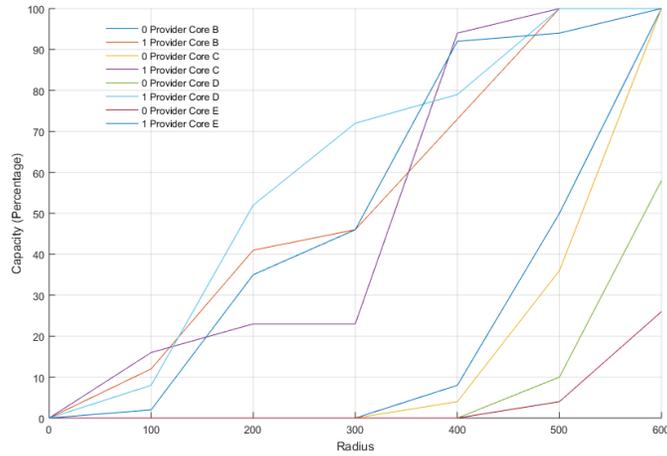


Figure 17: Topology 2: Simulation Result

scenario is shown in Table 2. The simulation results are shown in Fig. 17 and in Table 8 and Table 9, which shows the percentage of the required capacity that is met using the proposed architecture. As can be seen from the graph, the greater the radius to connect to the neighboring gNBs, the more is the failed demand which is recovered. In this simulation, the radius required to recover the failed demands is less because the number of gNB connected to each AR is large, so there are more number of neighboring gNB available as the radius to connect to the neighboring gNB increases.

This work was published in 2018 IEEE Globecom Workshops (GC Wkshps) [26]

Table 8: Topology 2: Percentage of traffic satisfied when AR B and AR C fails (one failure at a time)

Radius	AR B Failure		AR C Failure	
	0 Provider	1 Provider	0 Provider	1 Provider
0	0	0	0	0
100	0	12	0	16
200	0	41	0	23
300	0	46	0	23
400	8	73	4	94
500	50	100	36	100
600	100	100	100	100

Table 9: Topology 2: Percentage of traffic satisfied when AR D and AR E fails (one failure at a time)

Radius	AR D Failure		AR E Failure	
	0 Provider	1 Provider	0 Provider	1 Provider
0	0	0	0	0
100	0	8	0	2
200	0	52	0	35
300	0	72	0	46
400	0	79	0	92
500	10	100	4	94
600	58	100	26	100

4.3 A Heuristic for Network Resilience

Obtaining solutions for large scale problems using optimization tools like CPLEX [6] for the optimization formulation presented in Section 4.1 can be time-consuming as the number of variables and constraints increases with the increase in the number of nodes in a network. Therefore, in this section, we present our heuristic to solve the formulated problem.

The heuristic first tries to find the set of available nodes (\mathcal{N}) in the primary provider's network as well as the secondary provider's network for a failure. Based on the nodes available within the specified radius, the set of links (ℓ, ℓ') are allocated for both the primary provider's network as well as for the secondary provider's network. After that, the capacity constraints c_ℓ and c'_ℓ and the cost constraints ξ_ℓ and ξ'_ℓ are assigned to the links. We then consider demands, h_k , to be satisfied under failure. Our proposed heuristic then works by finding the maximal flow [27] that can be sent at a minimum cost over the network for a specific demand. If the maximal flow is greater than the demand that needs to be sent, the heuristic tries to find the flows for the minimum cost routing. After that, the available capacities of the links are updated. We then do this process for the next demand, one after another, in a sequential manner while updating the available capacities.

In our approach, we used three different rules to select the order of demands to be considered to find the maximal flow. In the first rule, the demands were selected based on the product of maximal flow and the cost incurred for each pair of demand. In the second rule, the product of the maximal flow, cost, and a number of links for each demand pairs were used to determine the sequence of demands to be considered. In the third and final

Algorithm 1 Heuristic

for each h_k selected using one of the three rules **do**
 find all $\mathcal{L}, \mathcal{L}'$
 find maximal flow f_k , which can be sent for h_k via all $\mathcal{L}, \mathcal{L}'$
 if $f_k > h_k$ **then**
 find minimum cost flows f_l, f'_l for each link ℓ, ℓ' for h_k
 $c_\ell \leftarrow c_\ell - f_l$
 $c'_\ell \leftarrow c'_\ell - f'_l$
 $w_k \leftarrow 0$
 else
 $h_k \leftarrow f_k$
 find minimum cost flows f_l, f'_l for each link ℓ, ℓ'
 $c_\ell \leftarrow c_\ell - f_l$
 $c'_\ell \leftarrow c'_\ell - f'_l$
 $w_k \leftarrow h_k - f_k$
 end if
end for

rule, the demands were sent in a sequential manner based on the normal pair numbering. In the end, the solution is selected based on the rule that produced the best result. The heuristic is shown in Algorithm 1.

4.4 Simulation Setup and Results

The scope of the simulation study is to understand the following issues:

- comparison of the optimization model and the heuristic
- present results on survivability and the trade-off between the primary provider and a secondary provider
- present results on quality-of-service based traffic for mission-critical traffic services.

We considered two core topologies, 3x3 grid topology and non-grid topology, as shown in Fig. 18 and Fig. 19, respectively. For each of the topology, 9 ARs in a 2000×2000 grid have been considered. Each AR has 49 gNBs connected to it (not shown in figures). Demands from ARs A, B, C, D, E to ARs F, G, H, I have been considered for each of the topologies. We considered two scenarios:

- When an AR fails, and no secondary provider is available to route the traffic
- When an AR fails, and a secondary provider is available to route the traffic

The radius to connect to the neighboring gNB was varied from 0 to 600 units, where the radius indicates the range within which the affected gNB can form a link with the neighboring gNBs (the primary provider or the secondary provider).

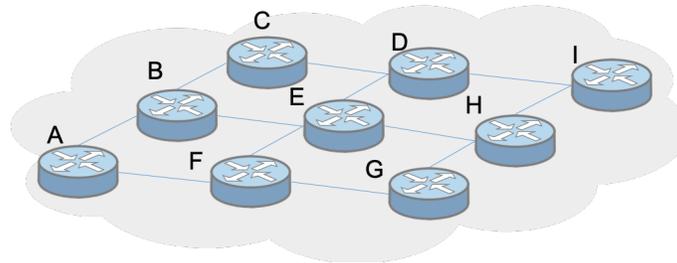


Figure 18: 3x3 grid topology

4.4.1 Comparison between Model and Heuristic

In the first set of experiments, we validate the performance of our heuristic over the proposed optimization model using the topology shown in Fig. 18. To show the effectiveness of our heuristic, we compared the simulation results using both CPLEX to solve the optimization formulation and using our heuristic.

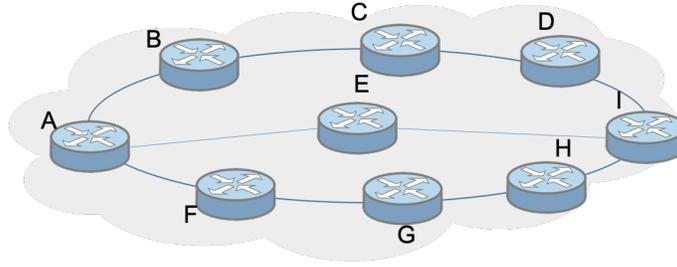


Figure 19: Non-grid topology

The link capacity values used for the simulation are shown in Table 10. In our study, we considered the failure of AR A, AR B, AR C, AR E, AR F, AR G and AR H (Fig. 18), one AR failure at a time. From the topologies, we can see that each AR failure affects the topology differently, while the radius gives us a metric on the spread of the failure. We compare the objective cost, and the total failed demand flow recovered for the network between the model (using CPLEX) and the heuristic. Comparisons are shown in Table 11 to Table 17.

In the tables shown, 0 provider refers to using only the primary provider for recovery of failed demands and 1 provider indicates using a secondary provider besides the primary provider for failure recovery. For AR A failure, the maximum deviation between CPLEX and heuristic results is less than 0.08% for 0 provider and 0.1% for 1 provider for objective cost, and no deviation in the failed demand flow recovered, as can be seen in Table 11. When AR B fails, the maximum deviation is 0.05% for 0 provider and 1.16% for 1 provider for objective cost, and no deviation in the total failed flow satisfied, as shown in Table 12. Similarly, for AR C, AR F, AR G and AR H the heuristic gives an ideal solution with the deviation between the results obtained from CPLEX and the heuristic

being considerably small.

From this set of results, we observed that average deviation between the CPLEX and heuristic results when using no secondary provider is 0.59% for the objective cost and 0.13% for the failed demand flow satisfied, and when using other provider, the average deviation for the objective cost is 0.45% and less than 0.01% for the total failed demand flow observed. Table 18 presents a summary of the CPLEX and heuristic comparison results. We also observed that CPLEX is not able to solve some of the optimization formulation as shown in Table 12, Table 14 which the heuristic can. Therefore, we can infer from the results that our heuristic gives close to the optimal solution when compared with the solution from the optimization model.

Table 10: Simulation Values: 3x3 grid topology

Link	Capacity
AR - AR	4000
AR - gNB	2000
gNB - gNB (same providers)	20
gNB - gNB (different providers)	20

Table 11: AR A failure: Model vs Heuristic

	0 Provider		1 Provider	
	Objective	Total flow satisfied	Objective	Total flow satisfied
100	0.0000%	0.0000%	0.0000%	0.0000%
200	0.0030%	0.0000%	0.0039%	0.0000%
300	0.0035%	0.0000%	0.0095%	0.0000%
500	0.0158%	0.0000%	0.0824%	0.0000%
600	0.0716%	0.0000%	0.1021%	0.0000%

Table 12: AR B failure: Model vs Heuristic

	0 Provider		1 Provider	
	Objective	Total flow satisfied	Objective	Total flow satisfied
100	0.0000%	0.0000%	0.0004%	0.0000%
200	0.0022%	0.0000%	0.0058%	0.0000%
300	0.0032%	0.0000%	0.0113%	0.0000%
500	0.0151%	0.0000%	1.1581%	0.0000%
600	0.0492%	0.0000%	Model Not Solvable	Model Not Solvable

Table 13: AR C failure: Model vs Heuristic

	0 Provider		1 Provider	
	Objective	Total flow satisfied	Objective	Total flow satisfied
100	0.0000%	0.0000%	0.0000%	0.0000%
200	0.0025%	0.0000%	0.0040%	0.0000%
300	0.0037%	0.0000%	0.0084%	0.0000%
500	2.2864%	2.2857%	0.0475%	0.0000%
600	0.0362%	0.0000%	1.6169%	0.0000%

Table 14: AR E failure: Model vs Heuristic

	0 Provider		1 Provider	
	Objective	Total flow satisfied	Objective	Total flow satisfied
100	0.0000%	0.0000%	0.0050%	0.0000%
200	0.0150%	0.0000%	0.0406%	0.0000%
300	0.0229%	0.0000%	0.2551%	0.0000%
500	4.2200%	0.0000%	Model Not Solvable	Model Not Solvable
600	4.8378%	0.0000%	Model Not Solvable	Model Not Solvable

Table 15: AR F failure: Model vs Heuristic

	0 Provider		1 Provider	
	Objective	Total flow satisfied	Objective	Total flow satisfied
100	0.0022%	0.0000%	0.0031%	0.0000%
200	0.0054%	0.0000%	0.0177%	0.0417%
300	0.0099%	0.0000%	0.0242%	0.0000%
500	0.0130%	0.0000%	0.0926%	0.0000%
600	0.0385%	0.0000%	3.8460%	0.0000%

Table 16: AR G failure: Model vs Heuristic

	0 Provider		1 Provider	
	Objective	Total flow satisfied	Objective	Total flow satisfied
100	0.0000%	0.0000%	0.0000%	0.0000%
200	0.0014%	0.0000%	0.0092%	0.0000%
300	0.0021%	0.0000%	0.0225%	0.0000%
500	0.0071%	0.0000%	0.5002%	0.0000%
600	0.0403%	0.0000%	0.4545%	0.0000%

Table 17: AR H failure: Model vs Heuristic

	0 Provider		1 Provider	
	Objective	Total flow satisfied	Objective	Total flow satisfied
100	0.0000%	0.0000%	0.0001%	0.0000%
200	0.0014%	0.0000%	0.0065%	0.0000%
300	0.0019%	0.0000%	0.0153%	0.0000%
500	0.0115%	0.0000%	1.5645%	0.0000%
600	0.0453%	0.0000%	0.1477%	0.0000%

Table 18: Model vs Heuristic Analysis

	0 Provider		1 Provider	
	Objective Cost	Total flow satisfied	Objective Cost	Total flow satisfied
Max Deviation	4.84%	2.29%	3.85%	0.04%
Min Deviation	0.00%	0.00%	0.00%	0.00%
Std Deviation	1.12%	0.39%	0.77%	0.01%
Median	0.01%	0.00%	0.02%	0.00%
Average Deviation	0.59%	0.13%	0.45%	0.00%

4.4.2 Result on Survivability and Tradeoff

This section presents results on the network survivability and tradeoff between using the primary provider and the secondary provider. The simulation topologies used are 3x3 grid topology and non-grid topology.

4.4.2.1 3x3 grid topology

The simulation results are shown from Fig. 21 to Fig. 27, while Fig. 20 denotes the legends for these graphs. The results are also shown from Table 19 to Table 25. Here,

the flow represents the percentage of failed traffic values that are satisfied, and the cost represents the percentage change in the total cost when the traffic is re-routed because of the failure. As can be seen from the tables, the larger the radius to connect to the neighboring gNBs, the more will be the failed demand, which would be recovered. For instance, when AR A fails, all gNBs connected to it fails. Then the affected gNBs connect to the neighboring gNB from the primary provider and from the secondary provider to route its traffic. Fig. 21 shows the percentage of failed traffic, which is satisfied using our proposed framework when AR A fails. Initially, when the node fails, we can see from the table that the percentage of failed demand met is 0%. As the radius to connect to the neighboring gNBs is increased, the percentage of satisfied traffic increases.

The additional cost represents the fluctuation in the cost when the traffic is re-routed. From the table, we can see that when the radius is 0, the added cost is 0%, as there are no available routes for the affected gNBs to satisfy the failed demands. However, with the increase in the radius, the percentage of satisfied demand increases as the number of links to connect to the neighboring gNBs increase resulting in the increase of the additional cost incurred to send the demands. As can be seen in Fig. 21, when the radius increases from 0 to 100, using no secondary provider, the additional cost incurred is 0% as the traffic can be re-routed using the same network and there is no extra cost. However, when the secondary provider is used, the cost increases by 7% as an external provider is routing the flow. As the radius to connect to the neighboring gNBs increase, the cost might decrease instead of increasing as it is possible that with the increase in radius, the traffic is routed using a more cost-effective route. The same can be seen in the table when

a secondary provider is used, and the radius increases from 500 to 600. The variation in the cost is dependent on the failure location of the node in the topology. As can be seen from the results shown from Table 19 to Table 25, the satisfied demands, and the cost varies with different node failures as different node failure affects the topology in a different way. In Fig. 24, for AR E failure, we can observe that the cost is comparatively higher as compared to other nodes in the topology as E is the core node in the topology. So, when E fails, the traffic which is supposed to pass through node E (when there is no failure) is re-routed via other nodes resulting in additional flow cost.

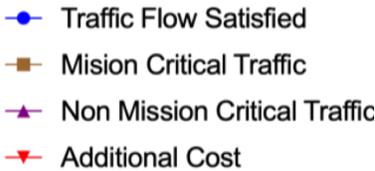


Figure 20: Legends: 3x3 grid topology and Non-grid topology

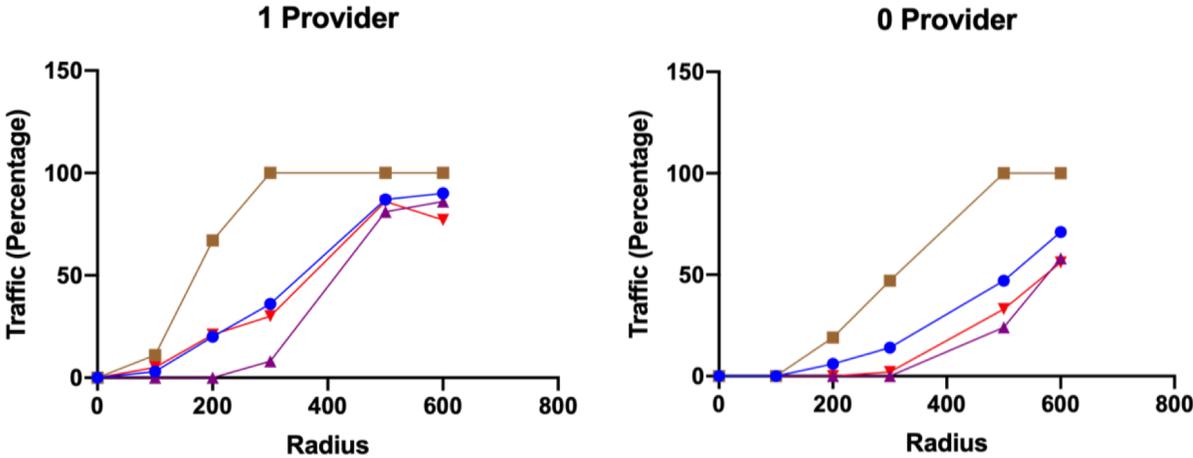


Figure 21: 3x3 grid topology: AR A failure

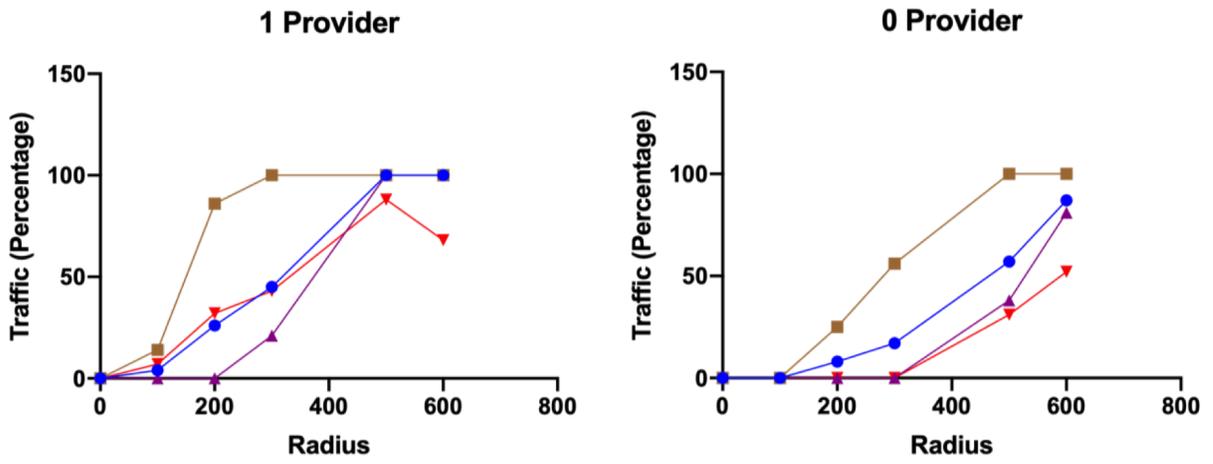


Figure 22: 3x3 grid topology: AR B failure

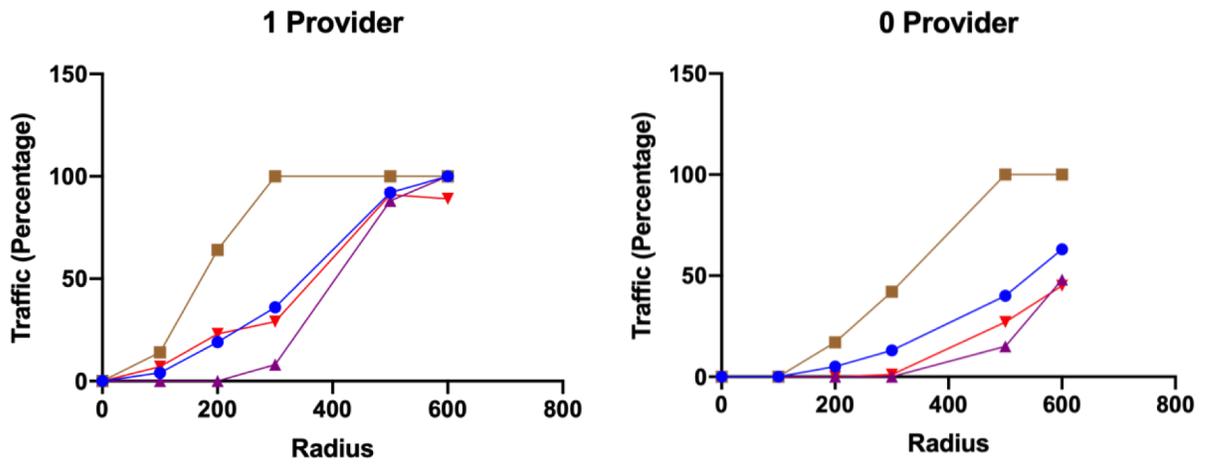


Figure 23: 3x3 grid topology: AR C failure

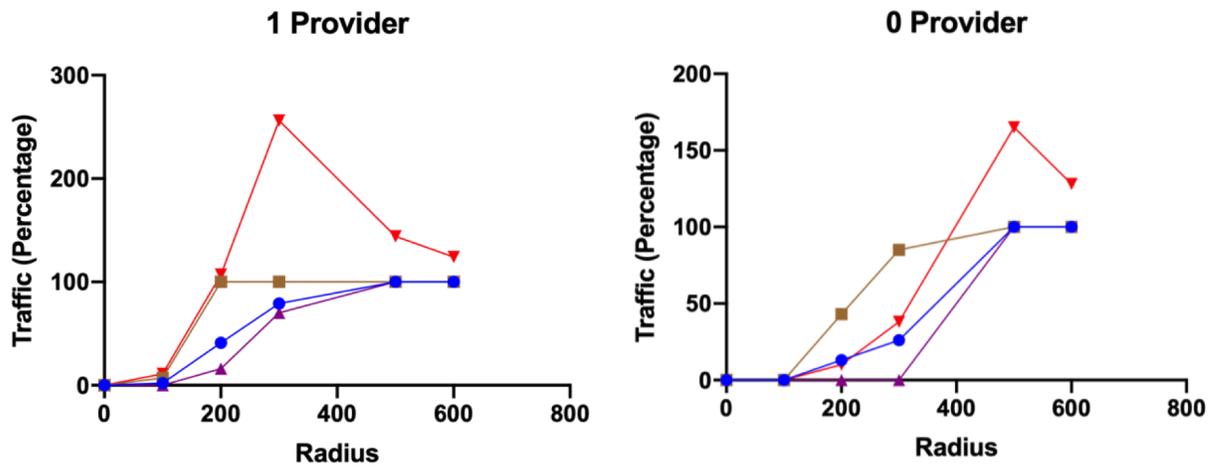


Figure 24: 3x3 grid topology: AR E failure

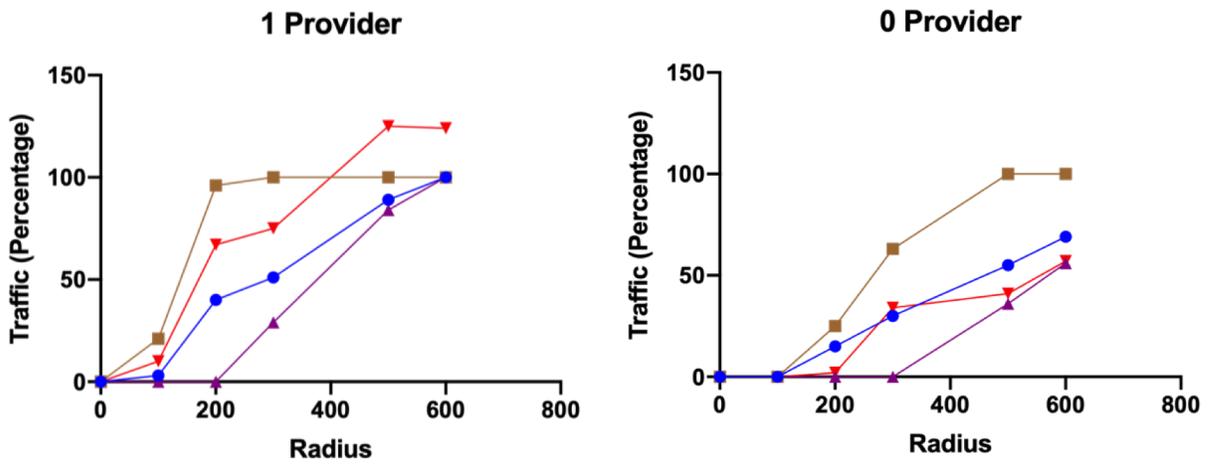


Figure 25: 3x3 grid topology: AR F failure

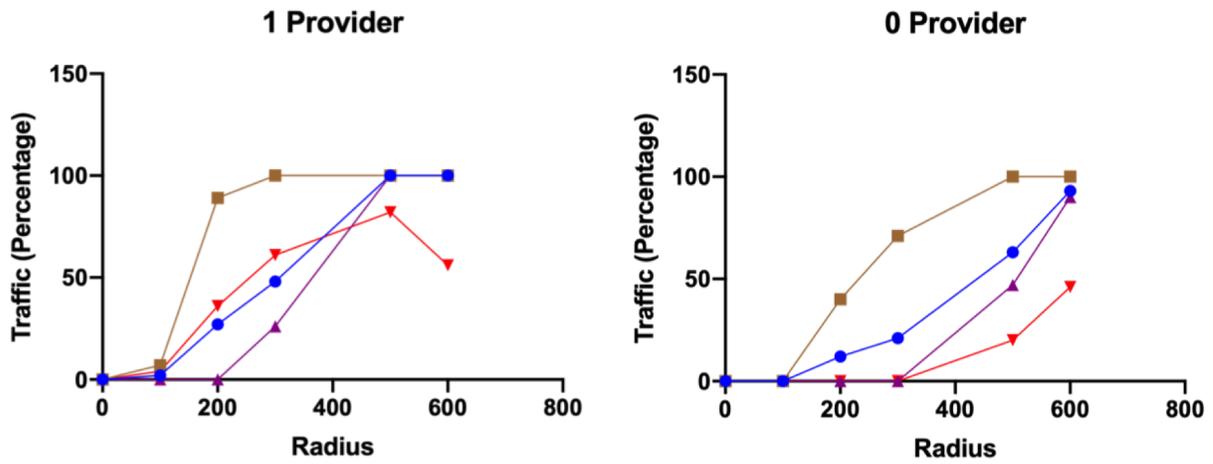


Figure 26: 3x3 grid topology: AR G failure

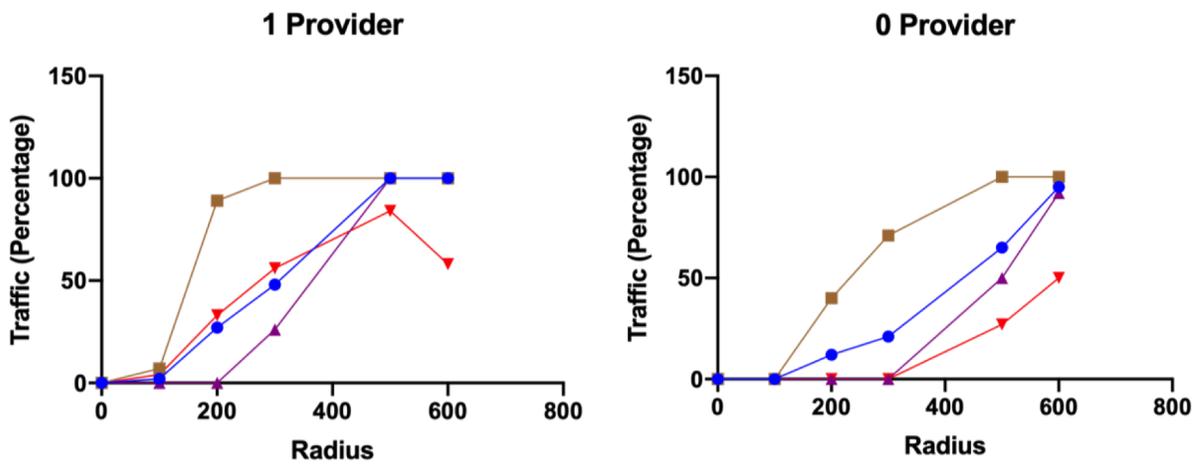


Figure 27: 3x3 grid topology: AR H failure

Table 19: 3x3 grid topology: Traffic satisfied and cost incurred when AR A fails

0 Provider				
Radius	Total Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	0%	0%	0%	0%
200	6%	19%	0%	0%
300	14%	47%	0%	2%
500	47%	100%	24%	33%
600	71%	100%	58%	56%

1 Provider				
Radius	Total Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	3%	11%	0%	5%
200	20%	67%	0%	21%
300	36%	100%	8%	30%
500	87%	100%	81%	86%
600	90%	100%	86%	77%

Table 20: 3x3 grid topology: Traffic satisfied and cost incurred when AR B fails

0 Provider				
Radius	Total Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	0%	0%	0%	0%
200	8%	25%	0%	0%
300	17%	56%	0%	0%
500	57%	100%	38%	31%
600	87%	100%	81%	52%

1 Provider				
Radius	Total Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	4%	14%	0%	7%
200	26%	86%	0%	32%
300	45%	100%	21%	43%
500	100%	100%	100%	88%
600	100%	100%	100%	68%

Table 21: 3x3 grid topology: Traffic satisfied and cost incurred when AR C fails

0 Provider				
Radius	Total Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	0%	0%	0%	0%
200	5%	17%	0%	0%
300	13%	42%	0%	1%
500	40%	100%	15%	27%
600	63%	100%	48%	45%

1 Provider				
Radius	Total Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	4%	14%	0%	7%
200	19%	64%	0%	23%
300	36%	100%	8%	29%
500	92%	100%	88%	91%
600	100%	100%	100%	89%

Table 22: 3x3 grid topology: Traffic satisfied and cost incurred when AR E fails

0 Provider				
Radius	Total Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	0%	0%	0%	0%
200	13%	43%	0%	10%
300	26%	85%	0%	38%
500	100%	100%	100%	165%
600	100%	100%	100%	128%

1 Provider				
Radius	Total Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	2%	7%	0%	11%
200	41%	100%	16%	107%
300	79%	100%	70%	256%
500	100%	100%	100%	144%
600	100%	100%	100%	124%

Table 23: 3x3 grid topology: Traffic satisfied and cost incurred when AR F fails

0 Provider				
Radius	Total Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	0%	0%	0%	0%
200	15%	25%	0%	2%
300	30%	63%	0%	34%
500	55%	100%	36%	41%
600	69%	100%	56%	57%

1 Provider				
Radius	Total Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	3%	21%	0%	10%
200	40%	96%	0%	67%
300	51%	100%	29%	75%
500	89%	100%	84%	125%
600	100%	100%	100%	124%

Table 24: 3x3 grid topology: Traffic satisfied and cost incurred when AR G fails

0 Provider				
Radius	Total Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	0%	0%	0%	0%
200	12%	40%	0%	0%
300	21%	71%	0%	0%
500	63%	100%	47%	20%
600	93%	100%	90%	46%

1 Provider				
Radius	Total Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	2%	7%	0%	4%
200	27%	89%	0%	36%
300	48%	100%	26%	61%
500	100%	100%	100%	82%
600	100%	100%	100%	56%

4.4.2.2 Non-grid topology

The second topology we used to show the effectiveness of our proposed algorithm is shown in Fig. 19. The link capacities for the topology are shown in Table 26. The results are shown from Fig. 28 to Fig. 34 (refer to Fig. 20 for the legends used in the graphs). The results are also shown in a tabular form from Table 27 to Table 34. The percentage of failed traffic satisfied is represented by the flow, and the cost represents the percentage change in the total cost when the traffic is re-routed because of the failure. From the results shown in the tables, we can see how the satisfied demand increases with

Table 25: 3x3 grid topology: Traffic satisfied and cost incurred when AR H fails

0 Provider				
Radius	Total Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	0%	0%	0%	0%
200	12%	40%	0%	0%
300	21%	71%	0%	0%
500	65%	100%	50%	27%
600	95%	100%	92%	50%

1 Provider				
Radius	Total Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	2%	7%	0%	4%
200	27%	89%	0%	33%
300	48%	100%	26%	56%
500	100%	100%	100%	84%
600	100%	100%	100%	58%

an increase in the radius and when using other provider to re-route the traffic. The results also demonstrate the fluctuation in the cost when the radius is increased, and an additional provider is used.

Compared to the 3x3 grid topology, the non-grid topology has less links as can be seen from Fig. 18 and Fig. 19. Although the links used in non-grid topology have greater capacity, having fewer links connected to the core node (E) overloads other links in the network where the capacity may not be sufficient to route the failed demands. This may affect the total flow satisfied and the additional cost incurred depending on the location of the failure node in the topology. When AR A fails, the total flow satisfied in

the 3x3 grid topology is more compared to non-grid topology as can be seen in Fig. 21 and Fig. 28. When using only primary provider, the maximal flow satisfied in case of 3x3 grid topology topology is 71% accounting for 100% mission-critical traffic and 58% non-mission critical traffic with 56% additional cost. In the case of non-grid topology, the maximal flow restored is only 23%, with 77% mission-critical traffic and no non-mission traffic being satisfied. The additional cost incurred is 38%. In case of AR B failure (Fig. 22, Fig. 29), we notice that for both the topologies, as the radius is increased to 500 and 600, the total flow satisfied and the cost incurred are near same. However, when AR E fails, we see that non-grid topology is more cost-effective than 3x3 grid topology. This is because when AR E fails, both the topologies behave like a ring network with non-grid topology having more link capacity as compare to 3x3 grid topology. Therefore, the gNBs in non-grid topology will require fewer links to re-route the failed demands as compared to 3x3 grid topology.

Table 26: Simulation Values: Non-grid topology

Link	Capacity
AR - AR	7000
AR - gNB	2000
gNB - gNB (same providers)	20
gNB - gNB (different providers)	20

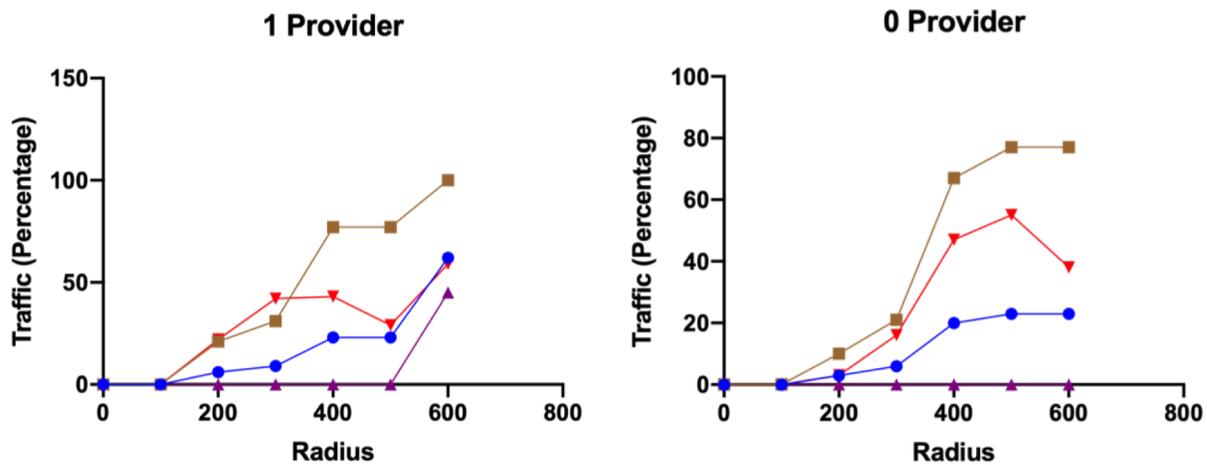


Figure 28: Non-grid topology: AR A failure

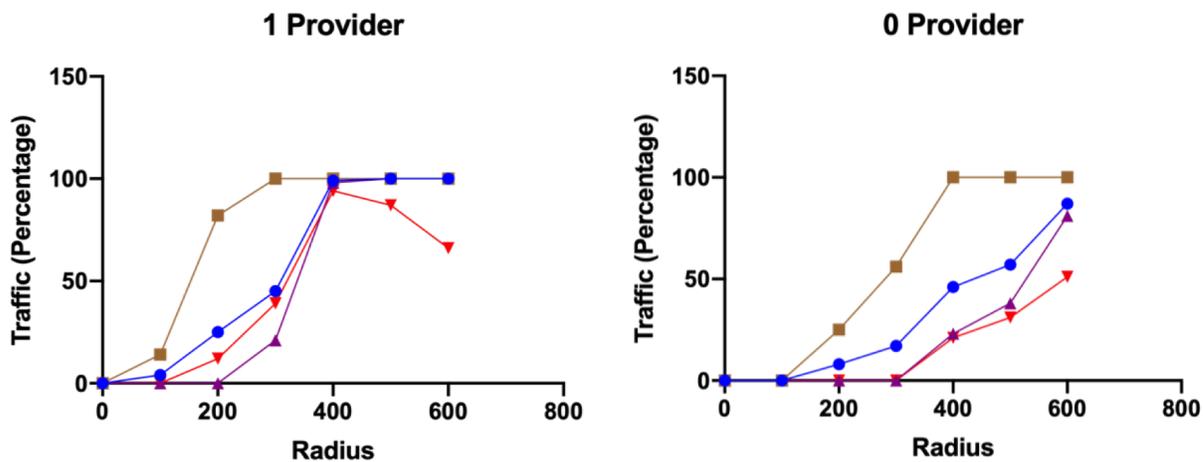


Figure 29: Non-grid topology: AR B failure

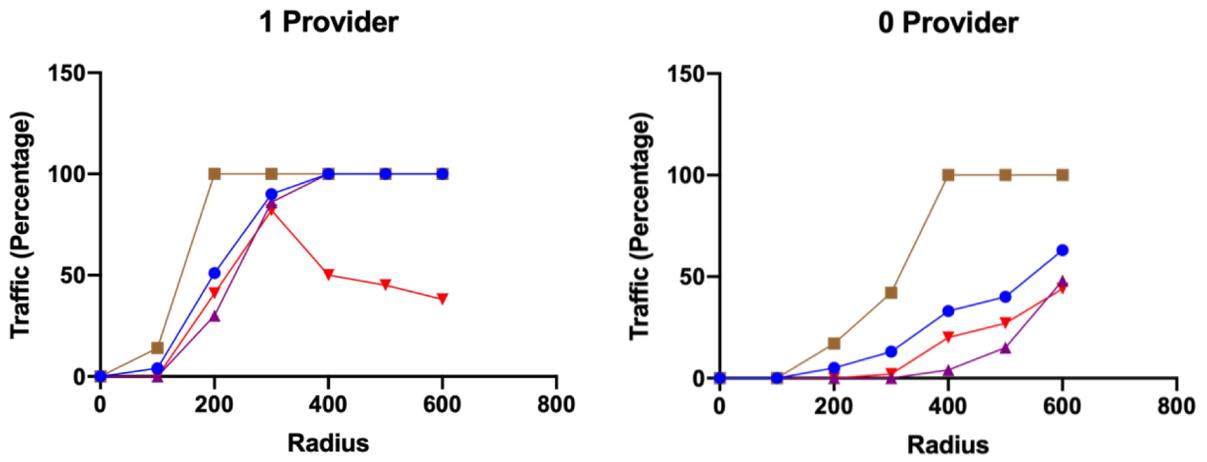


Figure 30: Non-grid topology: AR C failure

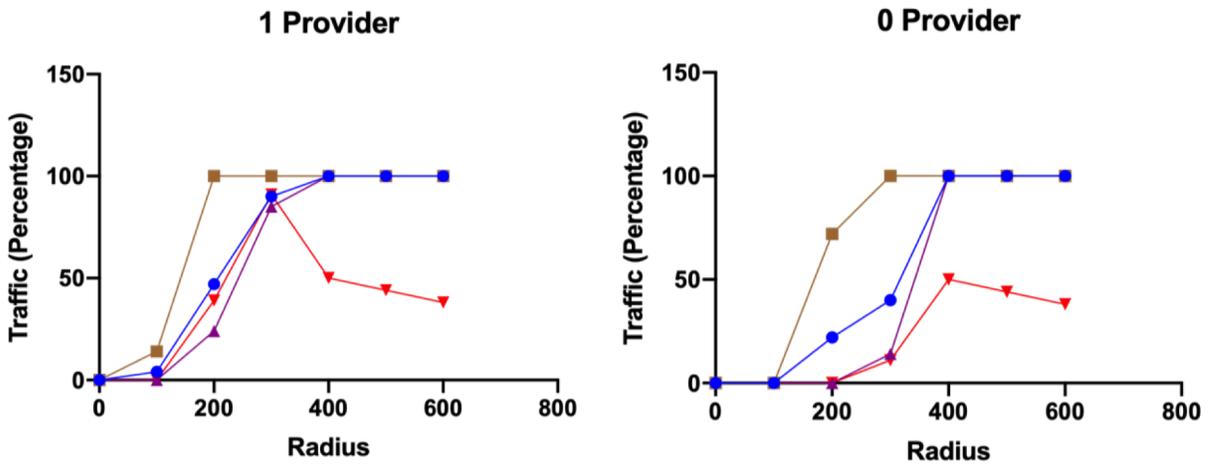


Figure 31: Non-grid topology: AR E failure

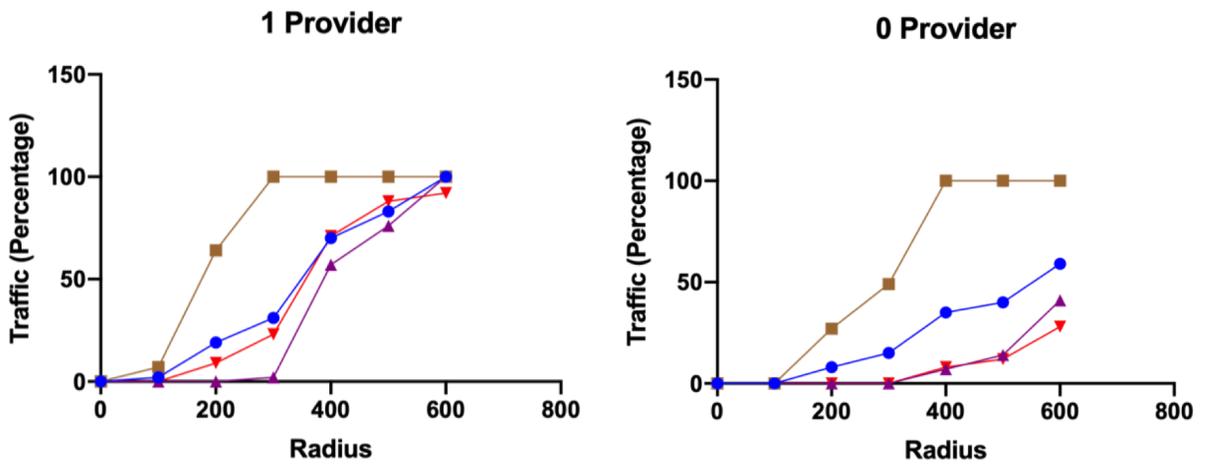


Figure 32: Non-grid topology: AR F failure

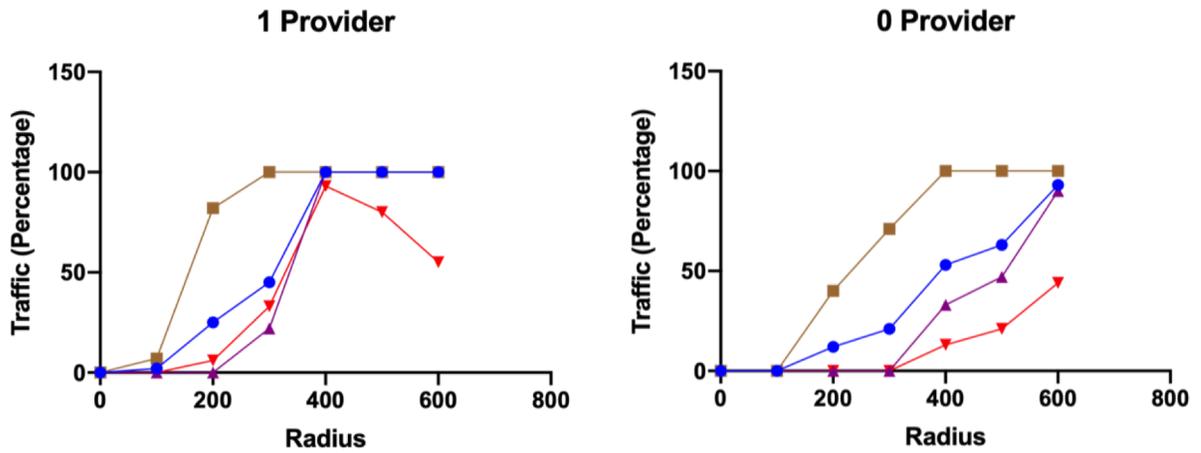


Figure 33: Non-grid topology: AR G failure

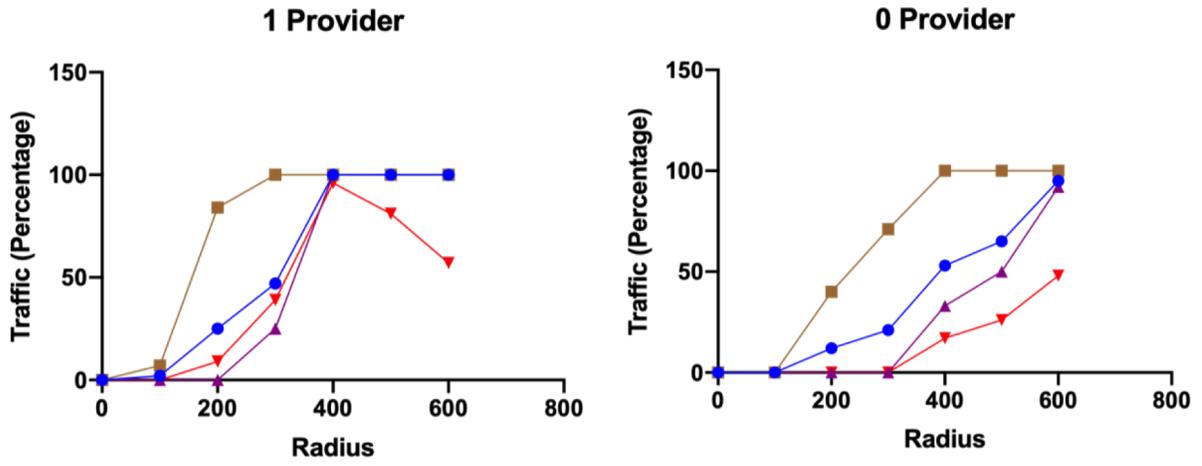


Figure 34: Non-grid topology: AR H failure

Table 27: Non-grid topology: Traffic satisfied and cost incurred when AR A fails

0 Provider				
Radius	Traffic Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	0%	0%	0%	0%
200	3%	10%	0%	3%
300	6%	21%	0%	16%
400	20%	67%	0%	47%
500	23%	77%	0%	55%
600	23%	77%	0%	38%

1 Provider				
Radius	Traffic Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	0%	0%	0%	0%
200	6%	21%	0%	22%
300	9%	31%	0%	42%
400	23%	77%	0%	43%
500	23%	77%	0%	29%
600	62%	100%	45%	59%

Table 28: Non-grid topology: Traffic satisfied and cost incurred when AR B fails

0 Provider				
Radius	Traffic Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	0%	0%	0%	0%
200	8%	25%	0%	0%
300	17%	56%	0%	0%
400	46%	100%	23%	21%
500	57%	100%	38%	31%
600	87%	100%	81%	51%

1 Provider				
Radius	Traffic Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	4%	14%	0%	0%
200	25%	82%	0%	12%
300	45%	100%	21%	39%
400	99%	100%	98%	94%
500	100%	100%	100%	87%
600	100%	100%	100%	66%

Table 29: Non-grid topology: Traffic satisfied and cost incurred when AR C fails

0 Provider				
Radius	Traffic Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	0%	0%	0%	0%
200	5%	17%	0%	0%
300	13%	42%	0%	2%
400	33%	100%	4%	20%
500	40%	100%	15%	27%
600	63%	100%	48%	44%

1 Provider				
Radius	Traffic Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	4%	14%	0%	0%
200	19%	64%	0%	7%
300	36%	100%	8%	31%
400	83%	100%	75%	83%
500	92%	100%	88%	90%
600	100%	100%	100%	85%

Table 30: Non-grid topology: Traffic satisfied and cost incurred when AR D fails

0 Provider				
Radius	Traffic Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	0%	0%	0%	0%
200	22%	72%	0%	0%
300	40%	100%	14%	10%
400	98%	100%	98%	49%
500	100%	100%	100%	45%
600	100%	100%	100%	38%

1 Provider				
Radius	Traffic Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	4%	14%	0%	0%
200	51%	100%	30%	41%
300	90%	100%	86%	82%
400	100%	100%	100%	50%
500	100%	100%	100%	45%
600	100%	100%	100%	38%

Table 31: Non-grid topology: Traffic satisfied and cost incurred when AR E fails

0 Provider				
Radius	Traffic Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	0%	0%	0%	0%
200	22%	72%	0%	0%
300	40%	100%	14%	11%
400	100%	100%	100%	50%
500	100%	100%	100%	44%
600	100%	100%	100%	38%

1 Provider				
Radius	Traffic Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	4%	14%	0%	0%
200	47%	100%	24%	39%
300	90%	100%	85%	91%
400	100%	100%	100%	50%
500	100%	100%	100%	44%
600	100%	100%	100%	38%

Table 32: Non-grid topology: Traffic satisfied and cost incurred when AR F fails

0 Provider				
Radius	Traffic Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	0%	0%	0%	0%
200	8%	27%	0%	0%
300	15%	49%	0%	0%
400	35%	100%	7%	8%
500	40%	100%	14%	12%
600	59%	100%	41%	28%

1 Provider				
Radius	Traffic Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	2%	7%	0%	0%
200	19%	64%	0%	9%
300	31%	100%	2%	23%
400	70%	100%	57%	71%
500	83%	100%	76%	88%
600	100%	100%	100%	92%

Table 33: Non-grid topology: Traffic satisfied and cost incurred when AR G fails

0 Provider				
Radius	Traffic Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	0%	0%	0%	0%
200	12%	40%	0%	0%
300	21%	71%	0%	0%
400	53%	100%	33%	13%
500	63%	100%	47%	21%
600	93%	100%	90%	44%

1 Provider				
Radius	Traffic Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	2%	7%	0%	0%
200	25%	82%	0%	6%
300	45%	100%	22%	33%
400	100%	100%	100%	93%
500	100%	100%	100%	80%
600	100%	100%	100%	55%

Table 34: Non-grid topology: Traffic satisfied and cost incurred when AR H fails

0 Provider				
Radius	Traffic Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	0%	0%	0%	0%
200	12%	40%	0%	0%
300	21%	71%	0%	0%
400	53%	100%	33%	17%
500	65%	100%	50%	26%
600	95%	100%	92%	48%

1 Provider				
Radius	Traffic Flow Satisfied	Mission Critical Traffic	Non Mission Critical Traffic	Additional Cost
0	0%	0%	0%	0%
100	2%	7%	0%	0%
200	25%	84%	0%	9%
300	47%	100%	25%	39%
400	100%	100%	100%	96%
500	100%	100%	100%	81%
600	100%	100%	100%	57%

4.4.3 Results on quality-of-service based traffic for mission-critical traffic services

Mission-critical traffic can be defined as traffic that is essential for the survival of a business, an organization, or society. The failure of such systems can have a significant effect on the operations of the system. Therefore, these traffic demands need to be supported in any circumstance such as helpline communication, traffic system, remote surgery, etc. In this analysis, we assume that the network has 30% mission-critical traffic and 70% non-mission-critical traffic. The mission-critical traffic in the table represents

the percentage of the total mission traffic whereas the non-mission-critical traffic is the percentage of the total non-mission-critical traffic which is satisfied.

Our study results for this scenario for 3x3 grid topology are shown from Table 19 to Table 34. From the results shown in the tables, when a failure occurs, the mission-critical traffic is prioritized over the other traditional non-mission-critical traffics. Hence, the non-mission-critical traffic is only addressed after all the mission-critical traffic demands have been satisfied.

This work has been accepted in the Journal of Network and Systems Management [28].

CHAPTER 5

UNLICENSED SPECTRUM BAND AND NON-TERRESTRIAL NETWORK

5.1 Unlicensed Spectrum Band

With the exponentially increasing communication demand, denser network deployment and multi-antenna systems may not be sufficient to satisfy the network capacity and throughput demand [29]. Therefore, additional spectrum will be required for the ever growing data traffic in future. With large amount of under-utilized spectrum available in the unlicensed band, it can be used to augment the network capacity to meet the increasing data demand. Besides, unlicensed spectrum can be used for use cases where using licensed spectrum band is a limitation [29].

Unlicensed frequencies are widely being used for various wireless communication such as broadband, mMTC, and URLLC applications [29]. However, there are major issues which need to be addressed to make such applications viable such as interference due to absence of LBT procedure. The NR is the 5G new radio designed to support key features like ultra-lean design, low latency and spectrum flexibility [30]. The use of NR in the unlicensed spectrum band can exploit the existing key mechanisms to enhance the QoS requirements, besides enabling new use cases. The existing and new licensed spectrum provides seamless coverage, high spectral efficiency, and high reliability as they are licensed for exclusive use by the IMT. However, using unlicensed spectrum will degrade the quality of service as they might suffer from interference and collision due to channel

load.

5.1.1 LBT

Listen Before Talk (LBT) is a mechanism to sense the channel for reducing interference and collision probability before actual transmission. In NR-U, the unlicensed spectrum available is limited and different (heterogenous) devices share the same channel to utilize it efficiently. Now, if more than one device try to access the channel at the same time, interference may arise and would limit the systems capacity. Therefore, to enable constructive coexistence between the devices operating in the unlicensed spectrum, any NR-U UE or gNB needs to perform an LBT procedure to avoid collision with any other nodes before initiating data transmission. If the LBT succeeds, the unlicensed band is occupied, and the data is transmitted, else the transmitter needs to wait and try to sense the channel again. To perform an LBT procedure, the device having data to transmit must perform Clear Channel Assessment (CCA) check to detect if the channel is idle or occupied. It should listen for at-least 15 μ sec on its operating channel. If the channel is sensed idle, the maximum contiguous transmission time should be less than 5 msec. The channel occupancy can be determined by the energy level in the channel. If the energy level in the channel exceeds the threshold, which is proportional to the power transmitted by the transmitter, the channel should be considered occupied, and the device should wait for random factor duration 1-20 times of the CCA observation time, before attempting further to access the channel again.

5.2 Non- Terrestrial Network

A non-terrestrial network (NTN) can be described as a network or a segment of networks which extends the usage of the existing RF resources to a satellite or UAS platform [31]. NTN can substantially add service benefits and resilience to the 5G networks services. Fig. 35 shows Non-Terrestrial Network. A non-terrestrial network typically consists of the following elements:

- A satellite (or UAS platform) which generates beams over a given service area which is bounded by its field of view
- Satellite gateway which connects the Non-Terrestrial Network to a public data network
- A Feeder link between the satellite- gateway and the satellite (or UAS platform).
- A Service link between the UE and the satellite (or UAS platform).
- An optional Inter-satellite links (ISL) inc are of a constellation of satellites

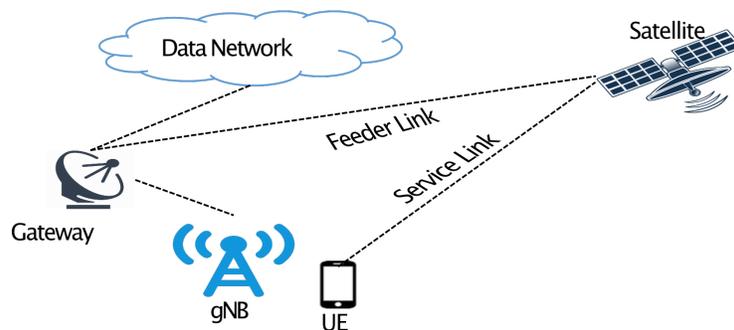


Figure 35: Use of Non-Terrestrial Network with 5G

CHAPTER 6

AN INTEGRATED 5G ARCHITECTURE FOR SURVIVABILITY

In our extended framework, we consider unlicensed spectrum and non-terrestrial networks with 5G. It may be noted that licensed spectrum in 5G in aggregation with unlicensed spectrum (2.4 GHz, 5 GHz, and 60 GHz) can increase the network capacity to a greater extent, thereby service the capacity needs in the event of a major failure. Use of the unlicensed band is a challenge since more than one user may be competing for the same channel simultaneously. Therefore, an interference mitigation technique is required. In this regard, the use of Listen Before Talk (LBT) mechanism can reduce the collision probability to a greater extent. LBT is a procedure to reduce interference by sensing the channel before transmission. If the LBT is successful, the data is transmitted, else the transmitter waits and after a predefined time senses the channel again. While choosing an unlicensed channel, the UE needs to measure the RSRP (Reference Signal Received Power) as well as the channel occupancy of the spectrum. High channel occupancy can result in high channel interference leading to LBT failure. Channel Access Priority Class (CAPC) has also been defined for use in unlicensed spectrum so as to prevent non-critical traffic getting prioritized over the critical ones.

A non-terrestrial network (NTN) can be described as a network or a segment of networks which extends the usage of the existing RF resources to a satellite or UAS platform [31]. NTN can substantially add service benefits and resilience to the 5G networks

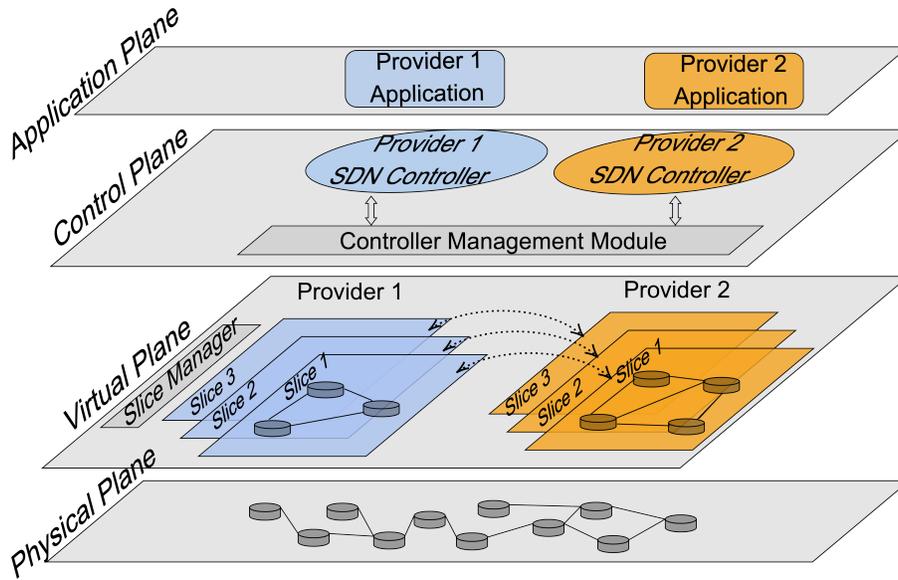


Figure 36: Extended 5G high level architecture

services. Fig. 35 shows Non-Terrestrial Network. It consists of one or more satellite gateway (terrestrial) that connects the NTN to the data network. The feeder link connects the satellite gateway and the satellite, and the service link is responsible for communication between the UE and the satellite. The gateway can be connected to the data network or to the gNB.

6.1 An Integrated 5G Architecture for Survivability

Our earlier framework for 5G networks [26] aimed to achieve network redundancy and load balancing in the network by forming an ad hoc network between the gNBs. Fig. 8 presents our proposed 5G architecture with virtualized mobile functions. Network slicing would allow the network providers to portion the network for specific use cases such as vehicular traffic and mission critical traffic and prioritize them based on the traffic

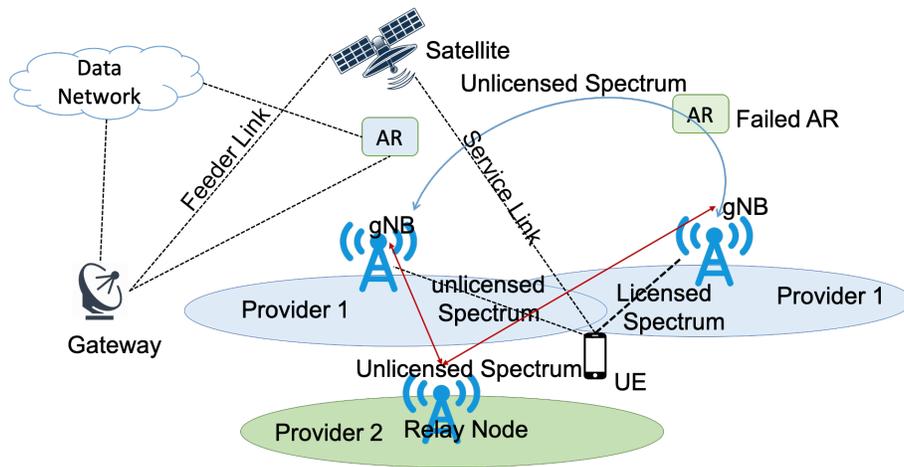


Figure 37: Proposed Resilient Framework

entity [32]. Multiple providers are considered that form a partnership. As shown in Fig. 36, the virtual layer contains the slices, which comprise of the virtual instance of each aggregation router (AR). All ARs would be mapped to its virtual instance. One AR can be mapped to more than one slice; however, each slice would be isolated from each other. A slice manager will be responsible for resource allocation to the slices as well for the SLA between different providers.

Each slice is connected to the SDN controller (control plane) via a controller management module (CMM). The CMM will work as a proxy layer between the virtual plane and the control plane. Besides, it would be responsible for traffic load balancing among the controllers from the same provider as well as different providers based on the SLAs. The SDN controller would then map each slice to the application layer where the VNF will be running on different virtual machines (VMs). The virtual layer and the control plane are the orchestration layer where multiple providers can coordinate resource or/and service orchestration with each other.

Now consider the scenario where an aggregation router fails or is unreachable due to a physical connectivity failure. During such situations, the gNBs can form a wireless network with neighboring gNBs from one provider as well as with other providers in the same service area by utilizing unlicensed spectrum band. Simultaneously, the UEs connected to the failed AR can use NTN to route the failed demands. It can also use licensed and unlicensed spectrum band to connect to any neighboring gNB within its range. We are proposing the use of unlicensed spectrum band as the licensed spectrum band is limited and traffic will not be overloaded in the cell sector of the gNB to which the UE is getting connected. So when an aggregation router fails, the gNBs attempt to automatically reroute the traffic using backup paths and provide network redundancy and load balancing.

Fig. 37 represents a failure scenario handling in our architecture. In case of an AR failure, an announce message is sent by the affected gNBs to its neighboring gNBs to notify them of the failure. Once the announce message is received, the neighboring gNBs send a reply to the affected gNBs about its policies, available capacity and cost. After the reply is received, the affected gNBs initiate Listen Before Talk (LBT) procedure to sense the channel so as to avoid collision with other operators using the same unlicensed spectrum band. If the LBT procedure is successful, the spectrum band is occupied and the connection is established with the neighboring gNBs. In case of LBT failure, the gNB has to wait for the next LBT timing. If the neighboring gNB belongs to the competitor provider, it tries to form a link with the nearest gNB belonging to the provider (of the failed core) to transmit the demand back into the network. After the link between the

gNBs is established, gNBs from the secondary provider behave simply like a Relay Node to push the traffic back into the core network. The gNB belonging to the same provider through which traffic enters back into the core (destination node) serves as the Donor gNB [25]. At the same time, the UE can connect to (1) neighboring gNBs within its range (using licensed and unlicensed spectrum), (2) NTN via the service link. The traffic routed via the NTN can then be pushed back to the core network as shown in Fig. 37. Our proposed framework provides three options for traffic routing:

1. when the same provider is used to route the affected traffic,
2. when affected traffic is routed by splitting via the same provider and a partner network provider that has an agreement with the first provider,
3. when the NTN is used to route the affected traffic.

The cost comparison may be a decision factor for choosing these options. The steps followed to form the ad hoc network in case of a failure are:

1. Form ad-hoc network among the gNBs
 - Each gNB is configured with a backup path. This backup path is calculated based on its distance from the neighboring gNBs (that could be the same provider or multiple providers).
 - In case of aggregation router failure, the gNBs tries to connect to the neighboring gNBs within the specified radius by sending a request to connect.
 - Based on the available capacity, the neighboring gNBs allocate some capacity to form a wireless link to the gNB requesting the connection.

- Once the links between the gNBs are established, backend features of PCRF and HSS are pushed to the gNBs and the links are activated.
2. The UE connects to the neighboring cell via the unlicensed spectrum band. LBT procedure is performed. If successful, the data is transmitted, else it needs to wait for next LBT timing.
 3. The UE connects to the NTN via the service link.

6.2 Optimization Model

We now present an optimization model to assess the impact of an aggregation router failure when the 5G network (the primary provider) is supplemented by unlicensed spectrum and NTNs as well as the presence of a secondary provider. We also consider establishing wireless links among the gNBs from the primary provider as well as secondary providers in this context. Table 35 summarizes the notations used in the model formulation.

The total traffic demand from the source to the destination is the sum of the demand carried in the primary provider's network, the secondary provider's network, NTN and the traffic demand which is not realized (w_k). This is given by:

$$x_k + x'_k + x''_k + w_k = h_k, \quad k \in \mathcal{K} \quad (6.1)$$

For the allocation of demand that falls on each network type, i.e., x_k in the primary

provider's network, x'_k in the secondary provider's network, x''_k in NTN, the flow conservation must be satisfied, which can be specified as follows:

$$\sum_{\ell \in \mathcal{L}} \delta_{v\ell} z_{\ell k} - \sum_{\ell \in \mathcal{L}} \gamma_{v\ell} z_{\ell k} = \begin{cases} x_k, & \& \text{if } v = s_k \\ 0, & \& \text{if } v \neq s_k, t_k, \\ \& v \in \mathcal{N} \\ -x_k, & \& \text{if } v = t_k \end{cases} \quad (6.2)$$

$$\sum_{\ell \in \mathcal{L} \cup \mathcal{L}'} \delta_{v\ell} z_{\ell k} - \sum_{\ell \in \mathcal{L} \cup \mathcal{L}'} \gamma_{v\ell} z_{\ell k} = \begin{cases} x'_k, & \& \text{if } v = s_k \\ 0, & \& \text{if } v \neq s_k, t_k, \\ \& v \in \mathcal{N} \\ -x'_k, & \& \text{if } v = t_k \end{cases} \quad (6.3)$$

$$\sum_{\ell \in \mathcal{L} \cup \mathcal{L}''} \delta_{v\ell} z_{\ell k} - \sum_{\ell \in \mathcal{L} \cup \mathcal{L}''} \gamma_{v\ell} z_{\ell k} = \begin{cases} x''_k, & \& \text{if } v = s_k \\ 0, & \& \text{if } v \neq s_k, t_k, \\ \& v \in \mathcal{N} \\ -x''_k, & \& \text{if } v = t_k \end{cases} \quad (6.4)$$

The traffic flows for all links in the primary provider's network, the secondary provider's network would be separated for flows on licensed spectrum and unlicensed spectrum:

$$\sum_{k \in \mathcal{K}} z_{\ell k} = f_{lc/\ell} + f_{ulc/\ell}, \quad \ell \in \mathcal{L} \quad (6.5)$$

$$\sum_{k \in \mathcal{K}} z_{\ell k} = f'_{lc/\ell} + f'_{ulc/\ell}, \quad \ell \in \mathcal{L}' \quad (6.6)$$

The remaining link flows are for links on the NTN:

$$\sum_{k \in \mathcal{K}} z_{\ell k} = f''_{\ell}, \quad \ell \in \mathcal{L}'' \quad (6.7)$$

The flows in (6.5) and (6.6) are then limited by the capacity constraints for licensed spectrum for the primary provider's network, and the secondary provider's network:

$$f_{lc/\ell} \leq c_\ell, \quad \ell \in \mathcal{L} \quad (6.8)$$

$$f'_{lc/\ell} \leq c'_\ell, \quad \ell \in \mathcal{L}' \quad (6.9)$$

In regard to unlicensed spectrum, the UEs are allowed to only use the unlicensed spectrum if the traffic flow is less than the channel occupancy thresholds (τ, τ') subject to available capacity:

$$f_{ulc/\ell} \leq \tau c_{ulc/\ell} \quad (6.10)$$

$$f'_{ulc/\ell} \leq \tau' c'_{ulc/\ell} \quad (6.11)$$

The last constraint is the capacity constraint in the NTN network:

$$f''_\ell \leq c''_\ell, \quad \ell \in \mathcal{L}' \quad (6.12)$$

The objective for the problem is given in terms of cost optimization for flows in different networks while accounting for a penalty for any unrealized demand.

$$\begin{aligned} \min_{\{x,z,f\}} \quad & \& \sum_{\ell \in \mathcal{L}} \sum_{k \in \mathcal{K}} (\xi_{lc/\ell} f_{lc/\ell} + \xi_{ulc/\ell} f_{ulc/\ell}) \\ & \& + \sum_{\ell \in \mathcal{L}} \sum_{k \in \mathcal{K}} (\xi'_{lc/\ell} f'_{lc/\ell} + \xi'_{ulc/\ell} f'_{ulc/\ell}) \\ & \& + \sum_{\ell \in \mathcal{L}} \sum_{k \in \mathcal{K}} T_\ell f''_\ell + \alpha \sum_{k \in \mathcal{K}} w_k \end{aligned} \quad (6.13)$$

We assume that capacity provisioned by the secondary provider to the primary provider will be based on the contractual requirements. Therefore, it is possible that some of the failed traffic demands may not be satisfied, thereby requiring us to consider w_k for unsatisfied demands via a penalty cost α in the objective function.

6.3 Simulation Setup and Results

In this section, we present results on survivability and discuss the tradeoff between using a single provider (primary) or using a secondary provider. We show how the use of unlicensed spectrum and NTN can greatly help recover the failed demands. The simulation topology is shown in Fig. 38. It consists of 5 ARs in a 1000×1000 grid with 5 gNBs connected to each of the AR. We assume that only UEs connected to AR A and AR B can connect to the NTN and traffic is routed back into the terrestrial network through AR D, as can be seen from the topology. We considered following simulation scenarios:

- When a failed AR traffic is routed with no parter provider
 - When only unlicensed spectrum band is available to route the traffic
 - When unlicensed spectrum band and NTN is available to route the traffic
- When a failed AR traffic is routed with one parter provider
 - When only unlicensed spectrum band is available to route the traffic
 - When unlicensed spectrum band and NTN is available to route the traffic

Table 35: Model Notations: 5G Network Resilience in presence of NTN and US

<p>Given</p> <p>\mathcal{N} := set of nodes (all types) in the provider's network \mathcal{K} := Set of demands identifiers that connects gNBs/ARs from provider's network to all destinations (internal or external) \mathcal{L} := Set of links (all types) in the primary provider's network \mathcal{L}' := Set of links in the secondary provider's network \mathcal{L}'' := Set of links in NTN h_k := Traffic demand for identifier k that connect a gNBs/AR to a destination s_k := Source node of identifier k t_k := Destination node of identifier k c_ℓ := Capacity on a link ℓ in the primary provider's network for licensed spectrum c'_ℓ := Capacity on a link ℓ in the secondary provider's network for licensed spectrum c''_ℓ := Capacity on a link ℓ in NTN $c_{ulc/\ell}$:= Capacity on a link ℓ in the primary provider's network for unlicensed spectrum $c'_{ulc/\ell}$:= Capacity on a link ℓ in the secondary provider's network for unlicensed spectrum $\delta_{v\ell}$:= 1 if link ℓ originates at node v; 0, otherwise $\gamma_{v\ell}$:= 1 if link ℓ terminates at node v; 0, otherwise $\xi_{lc/\ell}$:= Unit cost on a link ℓ in the primary provider's network for licensed spectrum $\xi_{ulc/\ell}$:= Unit cost on a link ℓ in the primary provider's network for unlicensed spectrum $\xi'_{lc/\ell}$:= Unit cost incurred on a link ℓ' in the secondary provider's network for licensed spectrum $\xi'_{ulc/\ell}$:= Unit cost incurred on a link ℓ' in the secondary provider's network for unlicensed spectrum T_ℓ := Unit cost incurred on a link ℓ'' in NTN α := Penalty cost incurred for not carrying load w_k τ := Channel occupancy threshold for unlicensed spectrum in primary provider's network τ' := Channel occupancy threshold for unlicensed spectrum in secondary provider's network</p> <p>Variables</p> <p>$x_k(\geq 0)$:= Part of the traffic demand that's carried in the primary provider's network $x'_k(\geq 0)$:= Part of the traffic demand that's carried in the secondary provider's network $x''_k(\geq 0)$:= Part of the traffic demand carried in NTN $w_k(\geq 0)$:= Part of the traffic demand not carried $z_{\ell k}(\geq 0)$:= Link flow on link ℓ for demand identifier k $f_{lc/\ell}(\geq 0)$:= Link flow on link $\ell \in \mathcal{L}$ (in primary provider's network) for licensed spectrum $f_{ulc/\ell}(\geq 0)$:= Link flow on link $\ell \in \mathcal{L}$ (in primary provider's network) for unlicensed spectrum $f'_{lc/\ell}(\geq 0)$:= Link flow on link $\ell \in \mathcal{L}'$ (in secondary provider's network) for licensed spectrum $f'_{ulc/\ell}(\geq 0)$:= Link flow on link $\ell \in \mathcal{L}'$ (in secondary provider's network) for unlicensed spectrum $f''_\ell(\geq 0)$:= Link flow on link $\ell \in \mathcal{L}''$ (in NTN)</p>

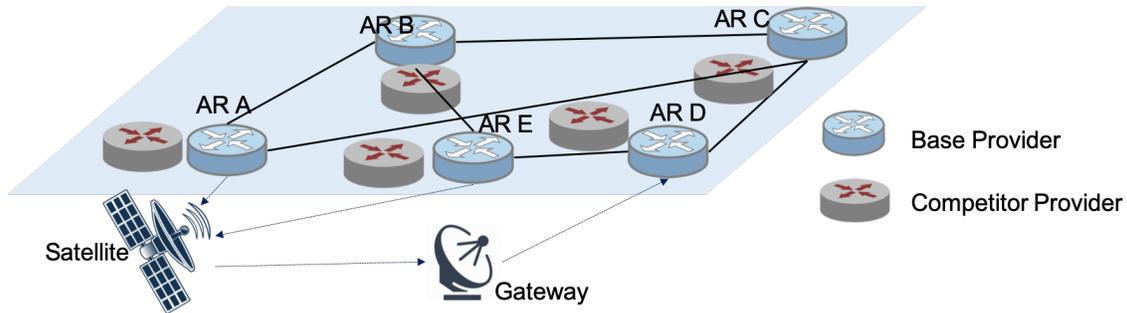


Figure 38: NTN integrated simulation topology

The radius to connect to the neighboring gNBs varied from 0 to 2000 units. Here, radius can be defined as the range within which the affected gNB can connect to neighboring gNBs via the unlicensed spectrum band. The gNB can either belong to the same provider or other provider. The simulation parameter values are shown in Table 37.

We have used normalized unit cost for links in our study by using one unit cost for links between AR-AR and AR-gNB. The unit cost for unlicensed links formed between the gNBs from the same provider has been considered to be higher (8) as the traffic is being routed via another gNB and might affect the traffic flow of the gNB to which the failed gNB gets connected. The unit cost for routing the demands is considered to be 16 for links between gNBs of different providers as routing traffic via an external provider will be significantly more expensive as compared to routing the traffic via gNB in the same provider's network. We also use unit cost for the unlicensed spectrum band (for same provider as well as other provider) since this band can be used by any operator or system without any prior licensing. We assume that in future the cost to access the NTN will be substantially low while having notable capacity. Therefore, in this simulation we

have considered the cost of routing traffic via NTN to be less than routing traffic via a competitor provider. In our study, AR A, AR B, AR C, and AR E failures have been considered with one failure at a time. From the simulation topology shown in Fig. 38, it is evident that each AR failure has different effect on the topology.

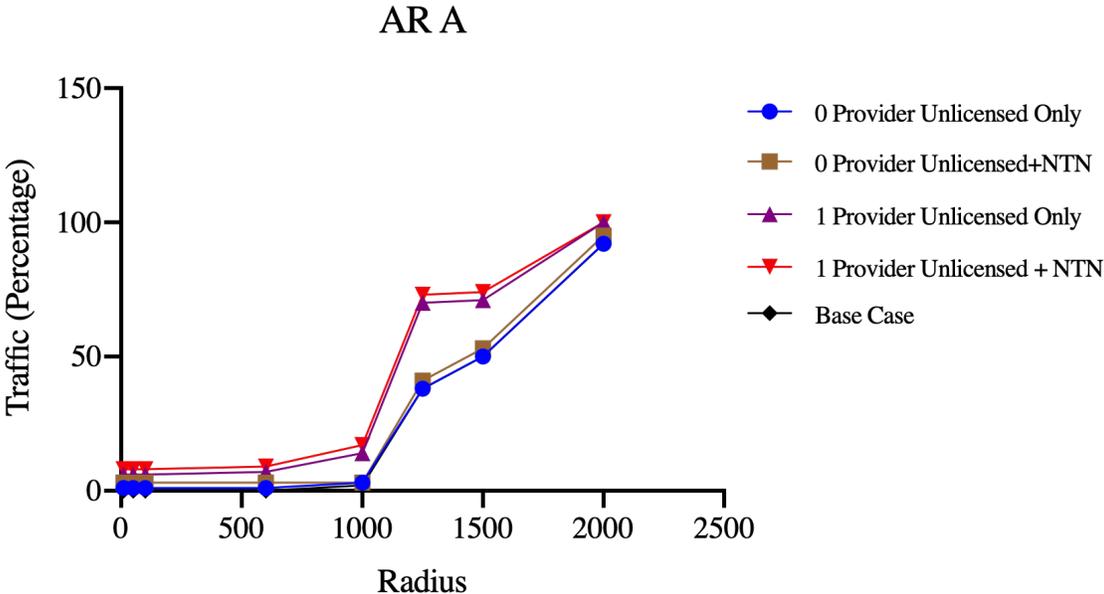


Figure 39: NTN + US: Aggregation Router A failure

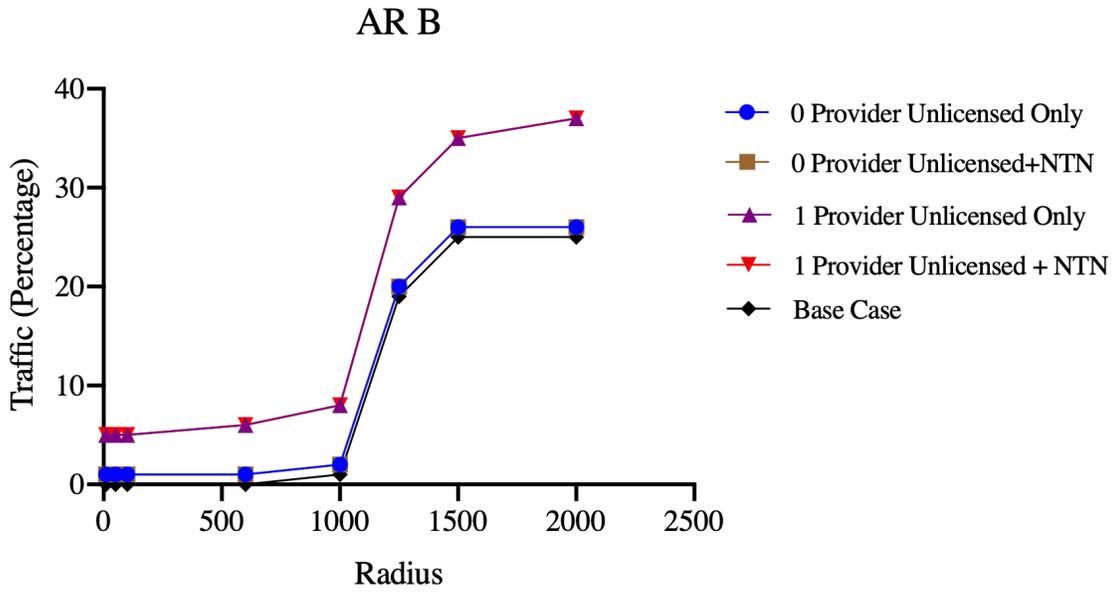


Figure 40: NTN + US: Aggregation Router B failure

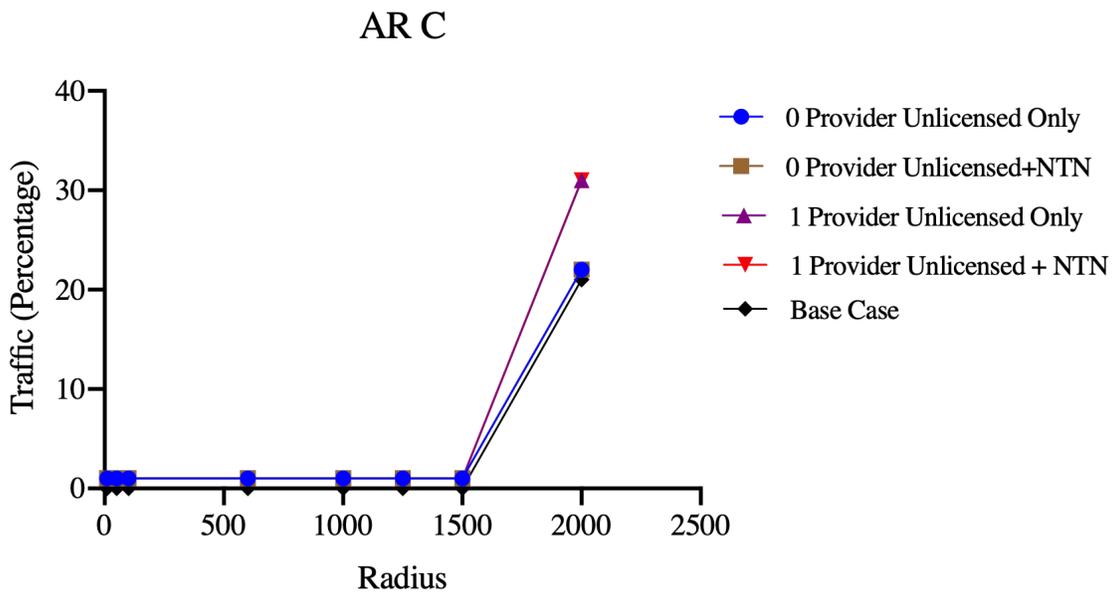


Figure 41: NTN + US: Aggregation Router C failure

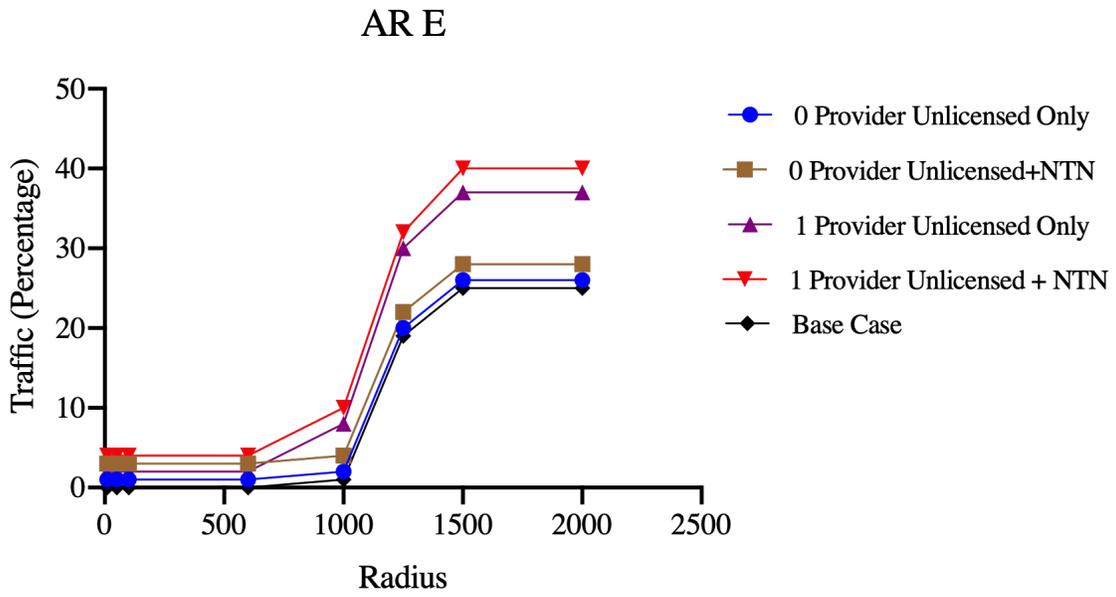


Figure 42: NTN + US: Aggregation Router E failure

Table 36: ABBREVIATIONS: NTN + US

	No of partner Provider	Unlicensed Spectrum availability for UEs	NTN availability
Base Case	0	No	No
0 Provider Unlicensed Only	0	Yes	No
0 Provider Unlicensed+NTN	0	Yes	Yes
1 Provider Unlicensed Only	1	Yes	No
1 Provider Unlicensed Only	1	Yes	Yes

Table 37: Simulation Values: NTN + US

Link	Capacity	Cost
AR-AR	12000	1
AR- gNB	2000	1
gNB-gNB (Same Provider,Unlicensed Spectrum)	40	8
gNB-gNB (Different Provider, Unlicensed Spectrum)	40	16
UE- NTN	200	12
NTN-gNB	200	12
UE-gNB (Licensed Spectrum)	4	1
UE-gNB (Unlicensed Spectrum)	5	1

Table 38: NTN + US: AR A failure

Radius	0 Provider				1 Provider			
	Unlicensed Only		Unlicensed+NTN		Unlicensed Only		Unlicensed+NTN	
	Objective	Virtual Flow	Objective	Virtual Flow	Objective	Virtual Flow	Objective	Virtual Flow
10	1%	1%	3%	3%	5%	6%	8%	8%
50	1%	1%	3%	3%	5%	6%	8%	8%
100	1%	1%	3%	3%	5%	6%	8%	8%
600	1%	1%	3%	3%	6%	7%	9%	9%
1000	3%	3%	3%	3%	14%	14%	16%	17%
1250	37%	38%	40%	41%	68%	70%	71%	73%
1500	49%	50%	52%	53%	69%	71%	72%	74%
2000	91%	92%	93%	95%	98%	100%	98%	100%

Table 39: NTN + US: AR B failure

Radius	0 Provider				1 Provider			
	Unlicensed Only		Unlicensed+NTN		Unlicensed Only		Unlicensed+NTN	
	Objective	Virtual Flow	Objective	Virtual Flow	Objective	Virtual Flow	Objective	Virtual Flow
10	1%	1%	1%	1%	5%	5%	5%	5%
50	1%	1%	1%	1%	5%	5%	5%	5%
100	1%	1%	1%	1%	5%	5%	5%	5%
600	1%	1%	1%	1%	6%	6%	6%	6%
1000	2%	2%	2%	2%	7%	8%	7%	8%
1250	20%	20%	20%	20%	29%	29%	29%	29%
1500	25%	26%	25%	26%	35%	35%	35%	35%
2000	25%	26%	25%	26%	36%	37%	36%	37%

Table 40: NTN + US: AR C failure

Radius	0 Provider				1 Provider			
	Unlicensed Only		Unlicensed+NTN		Unlicensed Only		Unlicensed+NTN	
	Objective	Virtual Flow	Objective	Virtual Flow	Objective	Virtual Flow	Objective	Virtual Flow
10	1%	1%	1%	1%	1%	1%	1%	1%
50	1%	1%	1%	1%	1%	1%	1%	1%
100	1%	1%	1%	1%	1%	1%	1%	1%
600	1%	1%	1%	1%	1%	1%	1%	1%
1000	1%	1%	1%	1%	1%	1%	1%	1%
1250	1%	1%	1%	1%	1%	1%	1%	1%
1500	1%	1%	1%	1%	1%	1%	1%	1%
2000	22%	22%	22%	22%	30%	31%	30%	31%

Table 41: NTN + US: AR E failure

Radius	0 Provider				1 Provider			
	Unlicensed Only		Unlicensed+NTN		Unlicensed Only		Unlicensed+NTN	
	Objective	Virtual Flow	Objective	Virtual Flow	Objective	Virtual Flow	Objective	Virtual Flow
10	1%	1%	3%	3%	2%	2%	4%	4%
50	1%	1%	3%	3%	2%	2%	4%	4%
100	1%	1%	3%	3%	2%	2%	4%	4%
600	1%	1%	3%	3%	2%	2%	4%	4%
1000	2%	2%	4%	4%	7%	8%	10%	10%
1250	19%	20%	22%	22%	41%	42%	43%	45%
1500	25%	26%	28%	28%	36%	37%	39%	40%
2000	25%	26%	28%	28%	36%	37%	39%	40%

The simulation results are shown in Fig. 39, Fig. 40, Fig. 41, and Fig. 42. The results are also shown in Table 38, Table 39, Table 40 and Table 41. The figures show the percentage of failed traffic demand which is recovered using our proposed framework. The x-axis shows the percentage of failed traffic demand which is recovered and the radius to connect to the neighboring gNBs is shown in y-axis. The radius for the UEs to connect to the neighboring gNBs when a failure occurs, is fixed. The legends used in the graphs are shown in Table 36. Besides, radius being a deciding factor for demand recovery, the use of unlicensed spectrum band and NTN by the UEs can further contribute to the route the failed demand. The same is shown in the graphs. For instance, incase of AR A failure,

all gNBs connected to it fail. Then, the affected gNBs connect to the neighboring gNBs from the same provider and from the secondary provider to route its traffic. At the same time the UEs, use NTN and unlicensed spectrum to connect to neighboring gNBs to meet the failed demand.

From the results shown in Fig. 39, Fig. 40, Fig. 41, and Fig. 42, we can see that the satisfied demands varies with different node failures as different node failure affects the topology in different way. In Fig. 39 and Fig. 42, the satisfied demand increases with the increase of radius. Here, both the nodes are connected to NTN, therefore, the satisfied demand increases with the use of unlicensed spectrum and NTN as compared to the base case. The base case here represents the satisfied demands when no other provider, unlicensed spectrum or NTN is available for traffic re-routing. Hence, Base Case < 0 Provider (Unlicensed only) < 0 Provider (Unlicensed + NTN) < 1 Provider (Unlicensed only) < 1 Provider (Unlicensed + NTN). In Fig. 40, Fig. 41, we can see that the flow satisfied using unlicensed spectrum when compared to unlicensed spectrum and NTN is same as the NTN is not connected to the UEs in AR B and AR C.

This work has been accepted in International Conference on Design of Reliable Communication Networks 2020 [33].

CHAPTER 7

CONCLUSION

In this research, we present a 5G network architecture with network virtualization and multiple providers for network resilience exploiting the unlicensed spectrum band and Non Terrestrial network.

We first propose a self-healing ad hoc network that may use a secondary provider when the backhaul network/ Aggregation network fails. The gNBs of the primary as well as the secondary provider forms a dynamic ad hoc network with the neighboring gNBs to route the failed demands effecting the capacity of the network minimally.

We then present an optimization model for survivability to show how a primary provider can coordinate with a secondary provider to recover any failed traffic demands. Our optimization formulation for network survivability captures the critical failure scenarios. We simulate two different network topology scenarios to demonstrate the effectiveness of our proposed model. Our results allow us to understand the trade off between using a provider's own network or rely on auxiliary capacity from another provider, depending on the spread of the failure indicated by a radius from the core node failure.

Solving large scale problems with optimization tools like CPLEX is difficult as the number of variables and constraints increases with the increase in the number of nodes in the network. Therefore, we propose a heuristic for network survivability that captures the critical failure scenarios by allowing a self-organizing ad hoc network among gNBs

for restoration. The optimality of the heuristic is shown by comparing the results with the optimization formulation. Through our simulation results, we show the trade off between using a provider's own network or relying on auxiliary capacity from a secondary provider, depending on the spread of the failure indicated by a radius from the core node failure. Two different topologies have been used to show the effectiveness of the proposed algorithm. We then present an analysis of how the mission-critical traffic can be recovered by using our proposed design. When a failure occurs, the mission-critical traffic is prioritized over the other traditional non- mission-critical traffics.

The last part of our research extends our 5G resilient network architecture. We demonstrate the use of unlicensed spectrum band and NTN for adding resilience to the network. Through our simulation results we show (i) the trade off between using a provider's own network or relying on auxiliary capacity from another provider, depending on the spread of the failure indicated by a radius from the core node failure and (ii) the use of unlicensed spectrum band and NTN in providing resilience to the network. Our results allow us to understand the critical role of unlicensed spectrum band and NTN in future networks.

Appendices

APPENDIX A

OPTIMIZATION MODEL FILE

```
# In this fifth example will add general link weights to the
third model.

# Network optimization model based a single type of line in the
# network. Unidirectional model with bi-directional
# constraints.
#
# Note that the "check" statements kick in when the solver is
run and not when the data is built.

# The nodes in the network will be listed in the data file.
set NODES;

# The set of source-destination pair traffic that we want to
route.

set SD_PAIRS within {NODES, NODES}; # will explicitly list the
pairs we want to route for

# The traffic demand between the source and destination
param demand {SD_PAIRS};
```

```

check {(s,d) in SD_PAIRS }: demand[s,d] = demand[d,s];

# The set of unidirectional links. This must be a symmetric
set.
set LINKS within {NODES, NODES};
check {(i,j) in LINKS }: ( (j,i) in LINKS); # (i,j) in LINKS
and

# The set of flow variables, i.e., the amount of traffic on
a particular link dedicated
# to a particular source-destination communication path.
var flow {LINKS, SD_PAIRS} >= 0;
# Bi-directional constraint, the flow between s and d over link
(i,j) must be the same as
# the flow between d and s over link (j,i).
subject to Bidirectional {(s,d) in SD_PAIRS, (i,j) in LINKS }:
flow[i,j,s,d] = flow[j,i,d,s];

# May need some extra variables for piecewise linear formulations
var total_flow {LINKS} >= 0;
var virtual_flow {SD_PAIRS} >= 0;
# Total flow variable is just the flow over the link. This

```

```

seems to be needed due to restrictions on
# argument to piecewise linear
subject to TotalFlow { (i, j) in LINKS }:
total_flow[i,j] = sum { (s, d) in SD_PAIRS } flow[i,j,s,d];

# For right now minimize over the sum of all flows:
param usage_cost{LINKS} >= 0; # Normal cost of using the link
param link_size {LINKS}>= 0; # Size of the links, e.g., 48 for
OC-48
minimize UsageCost: sum {(i,j) in LINKS}
usage_cost[i,j]*total_flow[i,j]+ sum {(i,j) in SD_PAIRS} 10000
*virtual_flow [i,j];

# Node balance equations. We need these at every node for
every source-destination pair.
# May be easiest to break into three cases: (i) where the node
the equation being written for
# is neither the source or destination of the flow, (ii) where
the node for the equation is
# the source for the flow, (iii) where the node for the equation

```

is the destination for the flow.

subject to BalanceNull {n in NODES, (s,d) in SD_PAIRS: s<>n and
d<>n }:

$$\sum \{(n, i) \text{ in LINKS}\} \text{flow}[n,i,s,d] - \sum \{(i,n) \text{ in LINKS}\} \text{flow}[i,n,s,d] = 0;$$

subject to BalanceSource {n in NODES, (n,d) in SD_PAIRS}:

$$\sum \{(n, i) \text{ in LINKS}\} \text{flow}[n,i,n,d] - \sum \{(i,n) \text{ in LINKS}\} \text{flow}[i,n,n,d] + \text{virtual_flow}[n,d] = \text{demand}[n,d];$$

subject to BalanceDest {n in NODES, (s, n) in SD_PAIRS}:

$$\sum \{(n, i) \text{ in LINKS}\} \text{flow}[n,i,s,n] - \sum \{(i,n) \text{ in LINKS}\} \text{flow}[i,n,s,n] [s,n] = -\text{demand}[s,n];$$

subject to LinkConstraint {(i, j) in LINKS }:

$$\text{total_flow}[i,j] \leq \text{link_size}[i,j] ;$$

APPENDIX B

HEURISTIC CODE

```
demand_value_default=[]
demand_from_list=[]
demand_to_list=[]
dct_demand_seq={}

for i in demand_from_list:
    for j in demand_to_list:
        #print("Demand ",i,j)
        G.add_node(i, demand=-demand_value_default)
        G.add_node(j, demand=demand_value_default)
        p=0
        try:
            flow_value, flow_dict = nx.maximum_flow(G, i, j)
            flowCost, flowDict = nx.capacity_scaling(G)
            for k1,v1 in flow_dict.items():
                for k2,v2 in v1.items():
                    if v2>0:
```

```

        p=p+1
        #flowCost, flowDict = nx.capacity_scaling(G)
    except:
        flow_value=0
        flowCost=0
        p=0
    dct_demand_seq[i, j]=flow_value*p*flowCost
    G.add_node(i, demand=0)
    G.add_node(j, demand=0)

items = [(v, k) for k, v in dct_demand_seq.items()]
items.sort()
#items.reverse()
items = [(k, v) for v, k in items]
#print(items)
#print(len(items))

demand_from_list=[]
demand_to_list=[]
for i in range (len(items)):
    demand_from_list.append(items[i][0][0])

```

```

        demand_to_list.append(items[i][0][1])
#print (demand_from_list)
#print (demand_to_list)

total_virtual_flow=0
total_cost=0
for i in range(len(demand_from_list)):
    #print ("demand start ",demand_start)
    flow_value=0
    flow_dict={}
    demand_start=demand_from_list[i]
    demand_end=demand_to_list[i]
    if demand_start!=demand_end:
        demand_value=demand_value_default
        try:
            flow_value, flow_dict = nx.maximum_flow(G, demand_start,
demand_end)
        except:
            flow_value=0
            flow_dict={}
        if flow_value<=demand_value:

```

```

        virtual_flow=demand_value-flow_value
        demand_value=flow_value
    else:virtual_flow=0
#     print("flow_value",flow_value)
#     print("virtual flow :", virtual_flow)
total_virtual_flow=total_virtual_flow+virtual_flow
G.add_node(demand_start, demand=-demand_value)
G.add_node(demand_end, demand=demand_value)
flowCost, flowDict = nx.capacity_scaling(G)
#     print("Total Cost ",flowCost)
total_cost=flowCost+total_cost
    #print(flowDict)
for key_1, value_1 in flowDict.items():
    for key_2, value_2 in value_1.items():
        #print (key_1,key_2)
        for key_3, value_3 in d_link_no.items():
            if value_3 == (key_1, key_2) or value_3 ==
(key_2, key_1):
                # print("Link no ",key_3)
                #print (value_2)
                #print("Bandwidth ",d_bandwidth[key_3])
                #print("New Bandwidth ", d_bandwidth[key_3]-

```

```

        # print ("Cost ",d_cost[key_3])
        d_bandwidth[key_3] = d_bandwidth[key_3]
- value_2

        G.remove_edge(key_1,key_2)
        G.remove_edge(key_2, key_1)
        G.add_edge(key_1, key_2, weight=d_cost[key_3],
capacity=d_bandwidth[key_3])
        G.add_edge(key_2, key_1, weight=d_cost[key_3],
capacity=d_bandwidth[key_3])

        # print (key_1,key_2,d_cost,d_bandwidth)

        G.add_node(demand_start,demand=0)
        G.add_node(demand_end,demand=0)

#print(d_bandwidth)
print("total virtual flow =", total_virtual_flow)
print("total cost =", total_cost+(total_virtual_flow*10000))
b = datetime.datetime.now()
print ("Simulation Time ", b-a)

```

APPENDIX C

TOPOLOGY GENERATION

```
from __future__ import print_function
from collections import defaultdict
import random
import math
import time

failed_switch_no=[ 1001]
demand_from_list=[1,2,3,4,5,6,7,8,9]
#print(len(demand_from_list))
demand_to_list=[439]
#print(len(demand_to_list))

bs_radius=100
bs_radius_o= 00

failed_bs_no=[ 0 ]
switches= 9
base_station= 49
```

```
switches_o= 9
base_station_o= 49

switches_o2= 9
base_station_o2= 4
bs_radius_o2= 00
#random.seed(99)
#demand to and from

log_demand_from=open("demand_from_list.txt","w")
print(demand_from_list,file=log_demand_from)

#demand value
demand=200
#link capacity
switch_bs_cap=2000
bs_bs_failed=20
```

```
bs_bs_cap=20
bs_other_bs_cap=20
switch_switch_cap=4000
#demand value
demand_service1=40
demand_service2=40
demand_service3=20
for i in range (len(failed_switch_no)):
    failed_switch=failed_switch_no[i]
    if failed_switch==1001:
        st_bs=1
        end_bs=(1*base_station)+1
    if failed_switch==1002:
        st_bs=1*base_station
        end_bs=(2*base_station)+1
    if failed_switch==1003:
        st_bs=2*base_station
        end_bs=(3*base_station)+1
    if failed_switch==1004:
        st_bs=3*base_station
        end_bs=(4*base_station)+1
    if failed_switch==1005:
```

```

        st_bs=4*base_station
        end_bs=(5*base_station)+1
if failed_switch==1006:
        st_bs=5*base_station
        end_bs=(6*base_station)+1
if failed_switch==1007:
        st_bs=6*base_station
        end_bs=(7*base_station)+1
if failed_switch==1008:
        st_bs=7*base_station
        end_bs=(8*base_station)+1
if failed_switch==1009:
        st_bs=8*base_station
        end_bs=(9*base_station)+1
    #else:
#st_bs=1
#end_bs=0
failed_bs_lst=[]
for i in range(st_bs,end_bs):
    failed_bs_lst.append(i)
log_linkcost_cap=open("linkcost_cap.txt","w")
log_data_file=open("data_file.txt","w")

```

```

log_graph_pairs=open("graph_pairs.txt","w")
log_link_nos=open("link_nos.txt","w")
log_top_gen=open("top_gen.txt","w")
d_cost={}
d_bandwidth={}
d_demands={}
d_demand_pairs={}
#Same Network
circle_r = 100 # radius of the circle
# center of the circle (x, y)
circle_x = 0
circle_y = 0
switch_radius=500
switch_list=[]
bs=[]
counter1=0
#1st Competitor Network
circle_r_o = 100 # radius of the circle
# center of the circle (x, y)
circle_x_o = 0
circle_y_o = 0
switch_radius_o=500

```

```

switch_list_o=[]
bs_o=[]
#2nd Competitor Network
circle_r_o2 = 100 # radius of the circle
# center of the circle (x, y)
circle_x_o2 = 0
circle_y_o2 = 0
switch_radius_o2=500
switch_list_o2=[]
bs_o2=[]
#link_cost for same network
switch_to_switch=1
switch_to_bs=1
bs_to_bs=8
bs_to_other_bs=12
#Switch connectivity
dict_switches={}
dict_basestations={}
dict_link_nos={} #dic to store the link_no for find the path
no in data file
bs_no=0
for i in range(1,switches):

```

```

    counter1 +=1
    dict_link_nos[counter1]=(i+1000,i+1001)
    d_cost[counter1]=switch_to_switch
    d_bandwidth[counter1]=switch_switch_cap
counter1 +=1
dict_link_nos[counter1]=(1001,1006)
d_cost[counter1]=switch_to_switch
d_bandwidth[counter1]=switch_switch_cap
counter1 +=1
dict_link_nos[counter1]=(1002,1005)
d_cost[counter1]=switch_to_switch
d_bandwidth[counter1]=switch_switch_cap
counter1 +=1
dict_link_nos[counter1]=(1005,1008)
d_cost[counter1]=switch_to_switch
d_bandwidth[counter1]=switch_switch_cap
counter1 +=1
dict_link_nos[counter1]=(1004,1009)
d_cost[counter1]=switch_to_switch
d_bandwidth[counter1]=switch_switch_cap
#9 switches with coordinates
dict_switches={1:(-1000,1000),2:(0,1000),3:(1000,1000),4:(-1000,0),5:(0,

```

```

switch_list=[(-1000,1000),(0,1000),(1000,1000),(-1000,0),(0,0),(1000,0),
for i in range(1,switches+1):
    switch_cordinate=(dict_switches[i][0],dict_switches[i][1])
    x=dict_switches[i][0]
    y=dict_switches[i][1]
    a=[]
    c=int(math.sqrt(base_station))
    interval=(switch_radius*2)/(c-1)
    for j in range (0,c):
        y1=(y+switch_radius)-(j*interval)
        for k in range(c):
            x1=(x+switch_radius)-(k*interval)
            counter1= counter1+1
            a.append((int(x1),int(y1)))
            bs.append(a)
            dict_basestations[bs_no+1]=(int(x1),int(y1))
            dict_link_nos[counter1]=(i+1000,bs_no+1)
            d_cost[counter1]=switch_to_bs
            d_bandwidth[counter1]=switch_bs_cap
            bs_no=bs_no+1
#####
#####

```

```

####          Fronthaul Failure          #####tw
#####
#####
#print("Considering ",failed_bs_no, " fronthaul failed")
failed_bs_link_no=[]
for i in range(len(failed_bs_no)):
    for key in dict_link_nos:
        if dict_link_nos[key][0] == failed_bs_no[i] or dict_link_nos[key]
== failed_bs_no[i]:
            failed_bs_link_no.append(key)
    for i in range(len(failed_bs_link_no)):
        d_bandwidth[failed_bs_link_no[i]] = 0
#####
#print(dict_basestations.items())
temp_list=[]
for i in range(st_bs,end_bs):
    x_axis=dict_basestations[i+1][0]
    y_axis=dict_basestations[i+1][1]
    for j in range(0,len(dict_basestations)):
        x_1_axis=dict_basestations[j+1][0]
        y_1_axis=dict_basestations[j+1][1]
        x_value=x_axis-x_1_axis

```

```

y_value=y_axis-y_1_axis
dist=math.sqrt( x_value**2 + y_value**2)
if (dist<bs_radius)and (dist!=0):
    for key, value in dict_basestations.items():
        if dict_basestations[j+1] == value and (key,i+1)
not in temp_list and (i+1,key) not in temp_list:
            counter1=counter1+1
            dict_link_nos[counter1]=(i+1,key)
            d_cost[counter1]=bs_to_bs
            if (st_bs)<j<end_bs:
                d_bandwidth[counter1] = bs_bs_failed
            else:
                d_bandwidth[counter1] = bs_bs_cap
            temp_list.append((i+1,key))

bs_first_network=bs_no
#print ("First Competitor Network")
dict_switches_o={}
dict_basestations_o={}
#bs_no_o=50
#9 switches with coordinates
dict_switches_o={1:(-1000,1000),2:(0,1000),3:(1000,1000),4:(-1000,0),5:(
switch_list_o=[(-1000,1000),(0,1000),(1000,1000),(-1000,0),(0,0),(1000,0)

```

```

for i in range(1,switches_o+1):
    switch_cordinate_o = (dict_switches_o[i][0], dict_switches_o[i][1])
    x_o = dict_switches_o[i][0]
    y_o = dict_switches_o[i][1]
    dict_switches_o[i+1001+switches]=switch_cordinate_o
    #switch_list_o.append((int(x_o),int(y_o)))
    a_o=[]
    c = int(math.sqrt(base_station_o))
    interval = (switch_radius_o * 2) / (c - 1)
    for j in range (0,c):
        y1_o = (y_o + switch_radius_o) - (j * interval)
        for k in range(c):
            x1_o = (x_o + switch_radius_o) - (k * interval)
            a_o.append((int(x1_o),int(y1_o)))
            bs_o.append(a_o)
            dict_basestations_o[bs_no+1]=(int(x1_o),int(y1_o))
            bs_no=bs_no+1
temp_list=[]
list_bs_sp = []
nearest = {}
#print(st_bs,end_bs)
for i in range(st_bs,end_bs+1):

```

```

x_axis_o=dict_basestations[i+1][0]
y_axis_o=dict_basestations[i+1][1]
for j in range(1,len(dict_basestations_o)+1):
    x_l_axis_o=dict_basestations_o[j+bs_first_network][0]
    y_l_axis_o=dict_basestations_o[j+bs_first_network][1]
    x_value_o=x_axis_o-x_l_axis_o
    y_value_o=y_axis_o-y_l_axis_o
    dist=math.sqrt( x_value_o**2 + y_value_o**2)
    if (dist<bs_radius_o):
        for key, value in dict_basestations_o.items():
            if dict_basestations_o[j+bs_first_network] ==
value and (key,i+1) not in temp_list and (i+1,key) not in temp_list:
                counter1=counter1+1
                dict_link_nos[counter1] = (i + 1, key)
                #print(dict_link_nos[counter1])
                d_cost[counter1] = bs_to_other_bs
                d_bandwidth[counter1] = bs_other_bs_cap
                temp_list.append((i+1,key))
                list_bs_sp.append((i + 1, key))
                for m in range(base_station + 1, bs_first_network):
                    if m not in range(st_bs,end_bs):
                        #print(m)

```

```

mx_axis_o = dict_basestations[m +
1][0]

my_axis_o = dict_basestations[m +
1][1]

mx_value_o = mx_axis_o - x_1_axis_o
my_value_o = my_axis_o - y_1_axis_o
dist = math.sqrt(mx_value_o ** 2
+ my_value_o ** 2)

nearest[dist] = (int(mx_axis_o),
int(my_axis_o))

nearest_value = min(nearest.items(), key=lambda
x: x[0])[1]

#print (nearest_value)

for k, value in dict_basestations.items():
    if k not in range(st_bs , end_bs ):
        if nearest_value == value and (k,
j + bs_first_network) not in temp_list and (j + bs_first_network,
k) not in temp_list:

            counter1 = counter1 + 1

            # print ("(N%d, "%(j+51), "N%d) "%key, end="
temp_list.append((j + bs_first_network,
k))

```

```

temp_list.append((k, j + bs_first_network
list_bs_sp.append((j + bs_first_network,
k))

dict_link_nos[counter1] = (j
+ bs_first_network, k)

#print(dict_link_nos[counter1])
d_cost[counter1] = bs_to_other_bs
d_bandwidth[counter1] = bs_other_bs_cap

second_network=bs_no
#print ("Second Competitor Network")
#dict_switches_o2={}
dict_basestations_o2={}
dict_switches_o2={1:(-1000,1000),2:(0,1000),3:(1000,1000),4:(-1000,0),5:
switch_list_o2=[(-1000,1000),(0,1000),(1000,1000),(-1000,0),(0,0),(1000,
#bs_no_o=50
for i in range(1,switches_o2+1):
    switch_cordinate_o2 = (dict_switches_o[i][0], dict_switches_o[i][1])
    x_o2 = dict_switches_o2[i][0]
    y_o2 = dict_switches_o2[i][1]
    #switch_cordinate_o2=(int (x_o2),int (y_o2))
    dict_switches_o2[i+1001+switches+switches_o]=switch_cordinate_o2
    switch_list_o2.append((int (x_o2),int (y_o2)))

```

```

a_o2=[]

c = int(math.sqrt(base_station_o2))

interval = (switch_radius_o2 * 2) / (c - 1)

for j in range (0,c):
    y1_o2 = (y_o2 + switch_radius_o2) - (j * interval)

    for k in range(c):
        x1_o2 = (x_o + switch_radius) - (k * interval)
        a_o2.append((int(x1_o2),int(y1_o2)))

        bs_o2.append(a_o2)

        dict_basestations_o2[bs_no+1]=(int(x1_o2),int(y1_o2))

        bs_no=bs_no+1

temp_list=[]

nearest = {}

list_bs_sp = []

for i in range(st_bs,end_bs):
    x_axis_o=dict_basestations[i+1][0]
    y_axis_o=dict_basestations[i+1][1]

    for j in range(1,len(dict_basestations_o2)+1):
        x_1_axis_o=dict_basestations_o2[j+second_network][0]
        y_1_axis_o=dict_basestations_o2[j+second_network][1]
        x_value_o=x_axis_o-x_1_axis_o
        y_value_o=y_axis_o-y_1_axis_o

```

```

dist=math.sqrt( x_value_o**2 + y_value_o**2)
if (dist<bs_radius_o2):
    for key, value in dict_basestations_o2.items():
        if dict_basestations_o2[j+second_network] ==
value and (key,i+1) not in temp_list and (i+1,key) not in temp_list:
            counter1=counter1+1
            dict_link_nos[counter1]=(i+1,key)
            d_cost[counter1] = bs_to_other_bs
            d_bandwidth[counter1] = bs_other_bs_cap
            temp_list.append((i+1,key))
# print(dict_link_nos[counter1])
for m in range(base_station + 1, bs_first_network):
    if m not in range(st_bs, end_bs):
        mx_axis_o = dict_basestations[m +
1][0]
        my_axis_o = dict_basestations[m +
1][1]
        mx_value_o = mx_axis_o - x_1_axis_o
        my_value_o = my_axis_o - y_1_axis_o
        dist = math.sqrt(mx_value_o ** 2
+ my_value_o ** 2)
        nearest[dist] = (int(mx_axis_o),

```

```

int(my_axis_o))

nearest_value = min(nearest.items(), key=lambda
x: x[0])[1]

#print (nearest_value)
#print (second_network)
for k, value in dict_basestations.items():
    if k not in range (st_bs,end_bs):
        #print(k)
        if nearest_value == value and (k,
j + second_network) not in temp_list and (j + second_network,
k) not in temp_list:

            temp_list.append((j + second_network,
k))

            temp_list.append((k, j + second_network)
list_bs_sp.append((j + second_network,
k))

            dict_link_nos[counter1] = (j
+ second_network, k)

            d_cost[counter1] = bs_to_other_bs
            d_bandwidth[counter1] = bs_other_bs_cap
            print(dict_link_nos[counter1])
print (dict_link_nos,file=log_link_nos) #contains all the link

```

```

no for finding the route no
#print(dict_link_nos)
for i in range (switches):
    #print (''%d':["%(i+1001),end="",file=log_top_gen)
    list_switch=[]
    for j in range (1,len(dict_link_nos)+1):
        try:
            if (i+1001)==dict_link_nos[j][0]:
                list_switch.append(dict_link_nos[j][1])
            if (i+1001)==dict_link_nos[j][1]:
                list_switch.append(dict_link_nos[j][0])
        except:
            pass
    for k in range(len(list_switch)):
        # print (''%d' "%list_switch[k],end="",file=log_top_gen)
        # print (''%d':[" % (i + 1001), end="", file=log_top_gen)
        if failed_switch!=(i+1001) and failed_switch!=list_switch[k]:
            print("g.addEdge(",i + 1001,",",list_switch[k], ")",
file=log_top_gen)
        # if k!=len(list_switch)-1:
        #     print (",",end="",file=log_top_gen)
    # print ("]",file=log_top_gen)

```

```

#print ("same basestation")
for i in range (len(dict_basestations)):
    # print ("%d' :["%(i+1),end="",file=log_top_gen)
    #print("g.addEdge(", i + 1, end="", file=log_top_gen)
    list_switch=[]
    for j in range (1,len(dict_link_nos)+1):
        try:
            if (i+1)==dict_link_nos[j][0]:
                list_switch.append(dict_link_nos[j][1])
            if (i+1)==dict_link_nos[j][1]:
                list_switch.append(dict_link_nos[j][0])
        except:
            pass
    for k in range(len(list_switch)):
        #print ("%d' "%list_switch[k],end="",file=log_top_gen)
        if failed_switch != (i + 1) and failed_switch != list_switch[k]:
            print("g.addEdge(",i+1,"" ,list_switch[k],")", file=log_top)
#print ("other basestaion")
for i in range (len(dict_basestations_o)):
    #print ("%d' :["%(i+1+bs_first_network),end="",file=log_top_gen)
    list_switch=[]
    for j in range (1,len(dict_link_nos)+1):

```

```

try:
    if (i+bs_first_network+1)==dict_link_nos[j][0]:
        list_switch.append(dict_link_nos[j][1])
    if (i+bs_first_network+1)==dict_link_nos[j][1]:
        list_switch.append(dict_link_nos[j][0])
except:
    pass

for k in range(len(list_switch)):
    #print ("%d'%d'"%list_switch[k],end="",file=log_top_gen)
    if failed_switch != (i + 1 + bs_first_network) and failed_switch
!= list_switch[k]:
        print("g.addEdge(", (i + 1 + bs_first_network), ",", list_swit
file=log_top_gen)

        #if k!=len(list_switch)-1:
            #print ("",end="",file=log_top_gen)
        #print ("]",end="",file=log_top_gen)
    # if i!=len(dict_basestations_o)-1:
        #print ("",file=log_top_gen)
#####
#####
## Switch Failure/Aggregation Network Failure ##
#####

```

```

#####
#print("Considering ",failed_switch_no, " failed")
#To put the capacity 0 for switches or links failed
failed_switch_link_no=[]
for i in range(len(failed_switch_no)):
    for key in dict_link_nos:
        #print (dict_link_nos[key])
        if dict_link_nos[key][0]==failed_switch_no[i] or dict_link_nos[k
:
            failed_switch_link_no.append(key)
    for i in range (len(failed_switch_link_no)):
        #print(failed_switch_link_no[i])
        d_bandwidth[failed_switch_link_no[i]]=0
remove_link=[]
for i in range (len(failed_switch_no)):
    failed_switch=failed_switch_no[i]
    if failed_switch==1001:
        for key,value in dict_link_nos.items():
            if dict_link_nos[key][0]==failed_switch or dict_link_nos[key
                remove_link.append(key)
    if failed_switch==1002:
        for key,value in dict_link_nos.items():

```

```

        if dict_link_nos[key][0]==failed_switch or dict_link_nos[key]
            remove_link.append(key)
if failed_switch==1003:
    for key,value in dict_link_nos.items():
        if dict_link_nos[key][0]==failed_switch or dict_link_nos[key]
            remove_link.append(key)
if failed_switch==1004:
    for key,value in dict_link_nos.items():
        if dict_link_nos[key][0]==failed_switch or dict_link_nos[key]
            remove_link.append(key)
if failed_switch==1005:
    for key,value in dict_link_nos.items():
        if dict_link_nos[key][0]==failed_switch or dict_link_nos[key]
            remove_link.append(key)
if failed_switch==1006:
    for key,value in dict_link_nos.items():
        if dict_link_nos[key][0]==failed_switch or dict_link_nos[key]
            remove_link.append(key)
if failed_switch==1007:
    for key,value in dict_link_nos.items():
        if dict_link_nos[key][0]==failed_switch or dict_link_nos[key]
            remove_link.append(key)

```

```

if failed_switch==1008:
    for key,value in dict_link_nos.items():
        if dict_link_nos[key][0]==failed_switch or dict_link_nos[key]
            remove_link.append(key)
if failed_switch==1009:
    for key,value in dict_link_nos.items():
        if dict_link_nos[key][0]==failed_switch or dict_link_nos[key]
            remove_link.append(key)
for i in range(len(remove_link)):
    del dict_link_nos[remove_link[i]]
    del d_bandwidth[remove_link[i]]
#print(dict_link_nos)
#for key, value in dict_link_nos.items():
#    if dict_link_nos[key][0] == 1 or dict_link_nos[key][1]==1:
#        print(dict_link_nos[key])

#print(d_demand_pairs)
# print ("demand_no=%d;BFS(graph, \"%d\", \"%d\", path_queue);"%(co
log_dict_link_nos=open("dict_link_nos.txt", "w")
log_d_cost=open("dict_link_cost.txt", "w")

```

```

log_d_bandwidth=open("dict_link_bandwidth.txt","w")
log_d_demands=open("dict_demands.txt","w")
log_d_demand_pairs=open("dict_demand_pairs.txt","w")
print (dict_link_nos,file=log_dict_link_nos)
#print (len(dict_link_nos))
#print (d_demands,file=log_d_demands)
print (d_bandwidth,file=log_d_bandwidth)
print (d_cost,file=log_d_cost)
print (d_demand_pairs,file=log_d_demand_pairs)
#print (dict_switches)
#print (dict_link_nos)
#print (d_cost)
#print (d_demands)
#print (d_demand_pairs)
#print (d_bandwidth)

#print (len(dict_link_nos))
d_demands={}
for i in demand_from_list:
    for j in demand_to_list:
        if (i!=j):
            d_demands[i,j]=demand

```

```
#print(d_demands)
print(d_demands, file=log_d_demands)
```

REFERENCE LIST

- [1] J. Prados-Garzon, O. Adamuz-Hinojosa, P. Ameigeiras, J. J. Ramos-Munoz, P. Andres-Maldonado, and J. M. Lopez-Soler, “Handover implementation in a 5g sdn-based mobile network architecture,” in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2016, pp. 1–6.
- [2] What is 5g ? [Online]. Available: <https://www.sdxcentral.com/5g/definitions/what-is-5g/>
- [3] T. Taleb, A. Ksentini, and B. Sericola, “On service resilience in cloud-native 5g mobile systems,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 3, pp. 483–496, 2016.
- [4] T. Taleb, M. Corici, C. Parada, A. Jamakovic, S. Ruffino, G. Karagiannis, and T. Magedanz, “Ease: Epc as a service to ease mobile core network deployment over cloud,” *IEEE Network*, vol. 29, no. 2, pp. 78–88, 2015.
- [5] “Why is 5g important?” Nov 2019. [Online]. Available: <https://www.verizon.com/about/our-company/5g/why-5g-important-discover-importance-5g-technology>
- [6] “Cplex optimizer.” [Online]. Available: <https://www.ibm.com/analytics/cplex-optimizer>

- [7] R. Abhishek, S. Zhao, and D. Medhi, "Spartacus: Service priority adaptiveness for emergency traffic in smart cities using software-defined networking," in *IEEE Smart Cities Conference (ISC2)*, 2016.
- [8] R. Abhishek, S. Zhao, S. Song, B.-Y. Choi, H. Zhu, and D. Medhi, "Buddi: Bug detection, debugging, and isolation middlebox for software-defined network controllers," in *2016 12th International Conference on Network and Service Management (CNSM)*. IEEE, 2016, pp. 307–311.
- [9] R. Abhishek, S. Zhao, D. Tipper, and D. Medhi, "Sesame: Software defined smart home alert management system for smart communities," in *2017 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*. IEEE, 2017, pp. 1–6.
- [10] A. Alqahtani, R. Abhishek, D. Tipper, and D. Medhi, "Disaster recovery power and communications for smart critical infrastructures," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.
- [11] M. M. S. Maswood, R. Abhishek, and D. Medhi, "Network optimization for differentiated qos traffic in an sdn environment for pop-data center traffic," in *2018 IEEE 39th Sarnoff Symposium*. IEEE, 2018, pp. 1–6.
- [12] C. Charnsripinyo and D. Tipper, "Topological design of 3g wireless backhaul networks for service assurance," in *Proc. of 5th Intl. Workshop on Design of Rel. Comm. Net. (DRCN)*, 2005.

- [13] E. A. Lemamou, P. Galinier, and S. Chamberland, “A hybrid iterated local search algorithm for the global planning problem of survivable 4g wireless networks,” *IEEE/ACM Transactions on Networking*, vol. 24, pp. 137–148, 2016.
- [14] D. Chen, S. Garg, and K. S. Trivedi, “Network survivability performance evaluation: a quantitative approach with applications in wireless ad-hoc networks,” in *Proc. 5Th Acm International Workshop On Modeling Analysis And Simulation Of Wireless And Mobile Systems*, 2002, pp. 61–68.
- [15] L. Xie, P. E. Heegaard, and Y. Jiang, “Network survivability under disaster propagation: Modeling and analysis,” in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*. IEEE, 2013, pp. 4730–4735.
- [16] O. Diaz, F. Xu, N. Min-Allah, M. Khodeir, M. Peng, S. Khan, and N. Ghani, “Network survivability for multiple probabilistic failures,” *IEEE Communications Letters*, vol. 16, no. 8, pp. 1320–1323, 2012.
- [17] J. P. Sterbenz, E. K. Çetinkaya, M. A. Hameed, A. Jabbar, S. Qian, and J. P. Rohrer, “Evaluation of network resilience, survivability, and disruption tolerance: analysis, topology generation, simulation, and experimentation,” *Telecomm. systems*, vol. 52, pp. 705–736, 2013.
- [18] D. Medhi and D. Tipper, “Multi-layered network survivability-models, analysis, architecture, framework and implementation: An overview,” in *DARPA Information Survivability Conference and Exposition, 2000 (DISCEX’00)*, vol. 1, 2000, pp. 173–186.

- [19] M. M. Rahman and S. S. Heydari, "A self-healing approach for lte evolved packet core," in *25th IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, 2012.
- [20] T. Taleb and K. Samdanis, "Ensuring service resilience in the eps: Mme failure restoration case," in *2011 IEEE Global Telecommunications Conference-GLOBECOM 2011*. IEEE, 2011, pp. 1–5.
- [21] "5g core (5gc) network: Get to the core of 5g," Mar 2020. [Online]. Available: <https://www.ericsson.com/en/networks/topics/5g-ready-core>
- [22] "5g virtual software network area." [Online]. Available: <https://www.5tonic.org/research/5g-virtual-software-network-area>
- [23] P. Beming, L. Frid, G. Hall, P. Malm, T. Noren, M. Olsson, and G. Rune, "Lte-sae architecture and performance," *Ericsson Review*, vol. 3, pp. 98–104, 2007.
- [24] "Policy and charging rules function (pcrf) in lte epc core network technology." [Online]. Available: <https://goo.gl/nM3rxu>
- [25] S. Sesia, M. Baker, and I. Toufik, *LTE-the UMTS Long Term Evolution: From Theory to Practice*. John Wiley & Sons, 2011.
- [26] R. Abhishek, D. Tipper, and D. Medhi, "Network virtualization and survivability of 5g networks: Framework, optimization model, and performance," in *2018 IEEE Globecom Workshops (GC Wkshps)*, Dec 2018, pp. 1–6.

- [27] L. Ford and D. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956.
- [28] R. Abhishek, D. Tipper, and D. Medhi, "Network virtualization and survivability of 5g networks," *Journal of Network and Systems Management*, 2020.
- [29] E. Semaan, J. Ansari, G. Li, E. Tejedor, and H. Wiemann, "An outlook on the unlicensed operation aspects of nr," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2017, pp. 1–6.
- [30] 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, 2017.
- [31] "Solutions for nr to support non-terrestrial networks." [Online]. Available: <https://portal.3gpp.org>
- [32] A. Thantharate, R. Paropkari, V. Walunj, and C. Beard, "Deepslice: A deep learning approach towards an efficient and reliable network slicing in 5G networks," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, October 2019.
- [33] R. Abhishek, D. Tipper, and D. Medhi, "Resilience of 5g networks in the presence of unlicensed spectrum and non-terrestrial networks," in *DRCN2020, International Conference on Design of Reliable Communication Networks*.

VITA

Rohit Abhishek was born in Bihar, India, on September 26, 1988. He received his Bachelor's degree in Electrical and Communication Engineering from Rajiv Gandhi Technical University and Masters in Electrical University from the University of Missouri-Kansas City, USA.

Mr. Abhishek worked as a system engineer for Tata Consultancy Services (2012-2013) and interned with Nokia (2017-2018), Mediatek (2019), and Tencent America (2019). He has published over eight research papers.

Mr. Abhishek currently works as a Researcher in Tencent America, Palo Alto, focussing on the 3GPP Standards.