Modeling of a 5G (or B5G) network to estimate its capacity to meet verticals resilience requirements

Rui Li

université PARIS-SACLAY

Modeling of a 5G (or B5G) network to estimate its capacity to meet verticals resilience requirements

Modélisation d'un réseau 5G (ou B5G) pour estimer sa capacité à répondre aux exigences de résilience des verticales

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 573, interfaces: matériaux, systèmes, usage (INTERFACES)
Spécialité de doctorat : Ingénierie des systèmes complexes
Graduate School : Sciences de l'ingénierie et des systèmes
Référent : CentraleSupélec

Thèse préparée dans **Laboratoire Génie Industriel (Université Paris-Saclay, CentraleSupélec)**, sous la direction de **Anne BARROS**, professeure, la co-direction de **Yiping FANG**, professeur, la co-codirection de **Zhiguo ZENG**, professeur.

Thèse soutenue à Paris-Saclay, le 12 décembre 2023, par

Rui LI

Composition du jury

Membres du jury avec voix délibérative

Salah EL AYOUBI Professeur, CentraleSupélec	Président & Examinateur
Dritan NACE Professeur, Université de technologie de Compiègne	Rapporteur & Examinateur
Liudong XING Professeure, University of Massachusetts	Rapportrice& Examinatrice
Laurent MASSOULIÉ Professeur, Inria Paris, DIENS PSL University	Examinateur
Emmanuel DOTARO Ingénieur, Thales Communications & Securityc	Examinateur

THESE DE DOCTORAT

NNT : 2023UPAST202

université PARIS-SACLAY | ÉCOLE DOCTORALE Interfaces: matériaux, systèmes, usages

Titre: Modélisation d'un réseau 5G (ou B5G) pour estimer sa capacité à répondre aux exigences de résilience des verticales

Mots clés: Résilience, systèmes complexes, réseaux 5G, modélisation, analyse de performance.

Résumé: De nombreux secteurs verticaux bénéficieront des promesses des réseaux 5G et au-delà. Plusieurs techniques sont introduites dans la 5G afin d'améliorer le réseau et de l'adapter à différents cas d'utilisation verticaux. Dans le même temps, ces nouvelles caractéristiques rendent le réseau 5G plus complexe. Avant la mise en service, les performances de résilience des réseaux 5G doivent être évaluées en fonctionnement normal et dans des situations de risque. Dans cette thèse, un modèle de réseau 5G a été proposé pour estimer la performance de résilience. Tout d'abord, la complexité et les caractéristiques dynamiques des réseaux 5G sont analysées, avec un point de vue de bout en bout et avec un point de vue multi-couche, en phase de conception et en phase opérationnelle. Deuxièmement, les exigences de résilience des différents domaines verticaux sont examinées. Les menaces et les risques liés à ces domaines verticaux sont également abordés. Troisièmement, différentes méthodes de modélisation sont comparées et un modèle basé sur un réseau de Petri ainsi qu'un modèle généralisé ont été mis en œuvre. Enfin, le modèle a été appliqué pour simuler de multiples cas d'utilisation afin d'estimer la résilience d'un réseau 5G dans différents scénarios.

Title: Modeling of a 5G (or B5G) network to estimate its capacity to meet verticals resilience requirements

Keywords: Resilience, complex system, 5G networks, modeling, performance analysis.

Abstract: Many vertical industries will benefit from the promise of 5G and Beyond networks. Multiple techniques are introduced to 5G in order to upgrade the network and to adapt it to different vertical use cases. At the meantime, these new features make 5G network more complex. Before serving these verticals, the resilience performance of 5G networks must be evaluated in normal operation and risk situations. In this dissertation, a 5G network model has been proposed for estimating 5G resilience performance. Firstly, the complexity and dynamic feature of 5G networks are analyzed, from End-to-end (E2E) and multi-layer perspectives and in design phase and operation phase. Secondly, resilience requirements from different vertical domains are examined. The threats and risks related to these verticals are also discussed. Thirdly, different modeling methods are compared and a Petri Net-based model and a generalized implementation of the model have been carried out. Finally, the model has been applied to simulate multiple use cases to estimate the resilience of a 5G network under different scenarios.

Acknowledgments

I would like to express my heartfelt gratitude to the individuals and organizations who have played an essential role in my academic journey and the completion of this thesis.

First and foremost, I want to sincerely thank my thesis supervisors at CentraleSupelec, Anne Barros, Yiping Fang, and Zhiguo Zeng, and my company supervisor, Bertrand Decocq, for offering me the opportunity to tackle the challenging topic of this thesis and for their unwavering support, guidance, and expertise. I am truly fortunate to have them on my thesis supervision team.

I would like to acknowledge my PhD jury members. Thanks to my thesis opponents, Dritan Nace and Liudong Xing, who kindly agreed to review my PhD manuscript. Also, I want to thank Emmanuel Dotaro, Salah El Ayoubi, and Laurent Massoulié for being as examiners of my thesis committee.

Special thanks go to Véronique Vèque and Christian Tanguy for their roles as my thesis follow-up committee members. Their feedback and suggestions for my midterm thesis work were essential in its development. Christian, in particular, has provided invaluable advice not only for my research but also for my future career.

To the members of the R^3 team in LGI laboratory, I express my gratitude for having interesting discussions in Openspace and during conferences. Their insights have expanded my horizons and inspired my work.

I sincerely appreciate the Orange Smart team (formerly the Brains team) for their countless support and assistance throughout my research.

I also want to acknowledge the partners from EDF, Enedis, and SNCF for providing inspiring use cases that added priceless value to my work.

To my friends, Fanwen, Yilun, and Zheyi, I extend my thanks for their companionship in Paris and for sharing both the happy and sad moments during my academic journey.

This thesis would not have been possible without the financial support provided by Orange Innovation, ANRT, and the chair RRCS. Their generosity enabled me to attend project meetings, conferences, training. I am truly grateful for their contributions to my academic and personal growth.

Last but certainly not least, I want to thank my family for their unconditional support throughout my three years of pursuing my Ph.D. research. Their encouragement and understanding have been a constant source of strength.

Synthèse

De nombreux secteurs verticaux bénéficieront des promesses des réseaux 5G et 5G au-delà. Plusieurs techniques sont introduites dans la 5G afin d'améliorer le réseau et de l'adapter à différents cas d'utilisation verticaux. Dans le même temps, ces nouvelles caractéristiques rendent le réseau 5G plus complexe. Avant la mise en service, les performances de résilience des réseaux 5G doivent être évaluées en conditions de fonctionnement normal et en cas de situations de risque. Dans cette thèse, l'objectif est de relever le défi d'estimer et de valider la résilience du réseau. Premièrement, une recherche documentaire sur le contexte de problématique est réalisée. Un réseau 5G peut être considéré comme un ensemble d'éléments. Ces éléments ou ces fonctions de réseaux, par exemple Gateway, Compresseur, etc., étaient physiques depuis longtemps. Avec l'arrivée de la 5G, il est envisagé que tous ces éléments soient virtualisés. Grâce à la virtualisation, chaque élément des réseaux devient une fonction réseau virtualisée (VNF), pouvant éventuellement être déployée selon les besoins et les exigences. La virtualisation du réseau entraîne à la fois une grande complexité dans l'architecture du réseau et une gestion du réseau plus agile et plus intelligente. En revanche, les nouveaux scenarios d'application d'un réseau télécom 5G présentent de nouveaux risques. La conception du réseau en tenant compte des différentes fonctions du réseau et la gestion du réseau en tenant compte des différents processus dynamiques doivent être adaptées à ces risques pour que la résilience du service soit satisfaite. La résilience, peut être définie comme la capacité d'une entité critique à prévenir tout incident, à s'en protéger, à y réagir, à y résister, à l'atténuer, à l'absorber, à s'y adapter et à s'en rétablir. Mais selon les scénarios, elle peut être interprétée par différent indicateurs.

Deuxièmement, afin de modéliser un réseau 5G, différents outils ou méthodologies peuvent être utilisés. Un simulateur du réseau est puissant pour obtenir le « Full Stack » de 5G. Mais il ne prend pas en charge toutes les parties ou toutes les caractéristiques de la 5G. En tenant compte des éléments choisis précédemment, un modèle mathématique est le mieux adapté dans ce cas pour modéliser les comportements d'un réseau 5G. Le modèle mathématique reproduit tous les algorithmes nécessaires dans les simulateurs et il est complété également par des fonctionnalités non présentées dans les simulateurs. Un Petri Net est puissant et assez flexible pour être appliqué dans ce projet de thèse. Petri Net et ses extensions sont adaptés pour concevoir plusieurs niveaux ou couches de représentation du réseau. Pour implémenter ce modèle, les Petri Nets sont ensuite codés en Python sous forme de simulation des éléments discrets.

Le modèle basé sur le Petri Net est d'abord étudié dans les différents scénarios de risque avant d'être appliqué dans les cas d'utilisation des verticaux. Le premier risque considéré dans la thèse est celui des défaillances du système. Les défaillances des objets physiques et des éléments virtuels avoir un impact sur la qualité, en particulier sur la disponibilité d'un réseau et d'un service. De plus, ces défaillances peuvent se propager entre les éléments. « Self-Healing » ou guérison automatique est l'un des processus dans la gestion du réseau qui atténue l'impact d'une défaillance en réparant ou en redémarrant un élément une fois que sa défaillance est constatée. Cependant, la fréquence de détection pour « Self-Healing » est un paramètre important et doit être configurée correctement conformément aux exigences. Le deuxième risque est la variation du trafic, celui-ci devient de plus en plus fréquent maintenant. Cette variation peut être causée, par exemple par le comportement anormal des utilisateurs ou une attaque externe. Cependant, ce changement de trafic n'est pas facile à anticiper mais peut engendrer une dégradation sévère sur la qualité du service. « Auto-Scaling » ou mise à échelle automatique réseau est l'un des processus de gestion des réseaux qui mitiger l'impact de la variation du trafic en adaptant l'échelle du réseau à la charge du trafic. Il existe plusieurs

stratégies ou algorithmes pour Auto-Scaling, par exemple, les stratégies de seuil, les réglages d'un PID, ou en utilisant l'IA. Une stratégie efficace doit répondre rapidement et correctement à la variation du trafic tout en allouant un nombre raisonnable de ressources. Par conséquent, la stratégie et les paramètres d'Auto-Scaling devront être choisis selon les scénarios et les modèles de trafic appliqués.

Finalement, le modèle construit est appliqué à deux cas d'usage sélectionnés dans deux verticaux. Le premier cas d'usage est « Télé-action » dans le domaine vertical des réseaux électrique. Dans ce cas d'usage, la disponibilité est étudiée. En comparant huit différentes conceptions, seul le design avec redondance partout dans le réseau satisfait l'exigence de cinq 9 sur la disponibilité d'un service de communication de Télé-action. Le deuxième cas d'usage est consacré aux utilisateurs à grandes vitesse, notamment dans le domaine ferroviaire. Dans ce cas d'utilisation, la disponibilité et la fiabilité de service sont considérées. Cependant, ces deux dernières sont liées non seulement à la disponibilité et la fiabilité des réseaux, mais également la maintenabilité de la connexion. Le processus « Handover » est indispensable pour maintenir la session de communication d'un utilisateur lorsqu'il change de point d'accès aux réseaux. Un modèle pour le processus de « Handover » plutôt dans le plan de contrôle d'un réseau 5G est ensuite ajouté. Un programme est développé pour lancer la simulation et estimer la disponibilité, la fiabilité du réseau et du service. Ce programme donne des idées pour les opérateurs du réseau télécom et les opérateurs du train pour voir ensemble comment la conception du réseau peut adapter au service des trains selon leurs vitesses et fréquences de passage.

Co-encadrement et Invité

Co-encadrement

Bertrand DECOCQ — Co-encadrant de thèse
Ph.D., Chef d'équipe, de programme et de projet, Orange

Invité

Thierry COUPAYE — Invité
Responsable de la recherche, Orange

Contents

List of Figures

List of Tables

Acronyms

5G the fifth Generation.

5G NR 5G New Radio.

5GB 5G and beyond.

6G the sixth Generation.

AMF Access and Mobility Management Function.

AUSF Authentication Server Function.

BBU Base Band Unit.

CN Core Network.

CP Control Plane.

CTMC Continuous-time Markov chain.

CU Central Unit.

DN Data Network.

DTMC Discrete-time Markov chain.

DU Distributed Unit.

E2E End-to-end.

gNB gNodeB.

HO Handover.

HPA Horizontal Pod Autoscaler.

IoT Internet of Things.

KPI Key performance indicators.

M2M Machine-to-Machine.

MANO Management and Orchestration.

MTBF Mean time between failures.

MTTF Mean time to failure.

MTTR Mean time to repair.

NF Network Function.

NFV Network Function Virtualization.

NFVI NFV Infrastructure.

NFVO NFV Orchestration.

PDU Protocol Data Unit.

RAN RAN Radio Access Network.

RU Radio unit.

SDN Software-Defined Networking.

SFC Service Function Chain.

SLA Service-level agreement.

SMF Session Management Function.

TN Transport Network.

UDM Unified Data Management.

UE User Equipment.

UP User Plane.

UPF User plane Function.

VIM Virtualized Infrastructure Manager.

VNF Virtual Network Function.

VNFM VNF Manager.

vRAN Virtualized radio access network.

1 - Introduction

1.1 . Research group

This doctoral thesis is supported by CIFRE[1] fellowship program in France with the collaboration of Orange Innovation and CentraleSupélec.

The industrial research work is carried out within the Simulation, Modeling, Analytic, Resilience, opTimization (SMART) team at Orange Innovation Networks. The main missions are as follows:

- Support research activities on cognitive network management, wholesale roaming optimization, and the resilience of complex systems.
- Contribute to discussions on AI/Operator interactions through the Cockpit Assistant Bidirectionnel (CAB) project.
- Contribute to work on the introduction of Artificial Intelligence for networks.

This thesis closely relates to the simulation, modeling, and resilience and contributes to the first team mission.

The academic research work is carried out within the *Risk Reliability Resilience (R^3)* research group within the Laboratory of Industrial Engineering (LGI) at CentraleSupélec, Université Paris-Saclay. The team is strongly connected to industry partners with the chair *Risk and Resilience of Complex System* supported by EDF (French electric utility company), Orange (French telecommunication corporation), and SNCF (France's national railway company).

The main research activities of the team focus on risk, reliability, and resilience analysis of complex engineered systems. The research is organized around three main studied objects:

- Complex systems and infrastructures, cyber-physical systems: to use stochastic processes, data-driven approaches, and Monte Carlo simulation to identify influential parameters and critical items and to define a proper level of abstractions for modeling.
- Industry 4.0 and predictive maintenance: to develop advanced models and optimization methods for dynamic risk management and predictive maintenance.

[1] **Conventions industrielles de formation par la recherche.** in French, an industrial agreement of training through research

- Resilience assessment and optimization: to assess and optimize the resilience of complex systems and (interdependent) critical infrastructures by modeling and optimizing the processes of barrier management, mitigation, crisis management, and recovery.

This thesis project is centered on complex communication systems modeling and communication network resilience evaluation.

1.2 . General insights and problem statement

With the emergence of the fifth Generation (5G) of cellular telecommunications technology, new network services will be offered as depicted in Figure 1.1, such as Machine-to-Machine (M2M) communications, notably involving connected objects that form Internet of Things (IoT). As for traditional data services, they will benefit from much higher throughput. This evolution of 5G creates a revolution for so-called verticals. These are generally industries or specific domains that aim to benefit from the advantages of 5G, both in terms of high performance and in terms of flexibility.

These verticals will also have more demanding and differentiated requirements for 5G-based services. For example, Industry 4.0 envisions a greater robotization of production processes, the healthcare sector will introduce Telehealth, and the transportation sector will promote autonomous vehicles.

These new service scenarios will all demand very low latency and very high network reliability. The Next Generation Mobile Networks (NGMN [1]), an alliance whose members are mobile network operators, in its white paper on 5G, provided an initial summary of these service requirements. Some more detailed requirements for 5G are expressed in specific application areas in a recent white paper "Key 5G Use Cases and Requirements" by 5G Alliance for Connected Industries and Automation (5G-ACIA [2]). These vertical requirements have given rise to technical specifications from the organization in charge of 5G standardization, 3GPP (3rd Generation Partnership Project), either globally to all services relying on a 5G network [3, 4], specific to critical services [5], or specifically to one kind of vertical, such as rail transport [6].

Some of these requirements may directly or indirectly fall into the category of resilience requirements. However, the term "resilience" can have several meanings depending on the risks to be faced, the service to be rendered, and the business to be satisfied. The literature provides examples of how resilience can be defined across a range of domains. In the article "Defining Resilience" [8], Rosowsky defines *resilience* in the event of a natural disaster in the engineering field as the ability to continuously provide critical services or functions in energy, telecommunications, or transport domains. Clément et al. [9] have attempted to synthesize and compare the definitions of resilience and robustness by analyzing the literature from 1975 to 2017 and have classi-

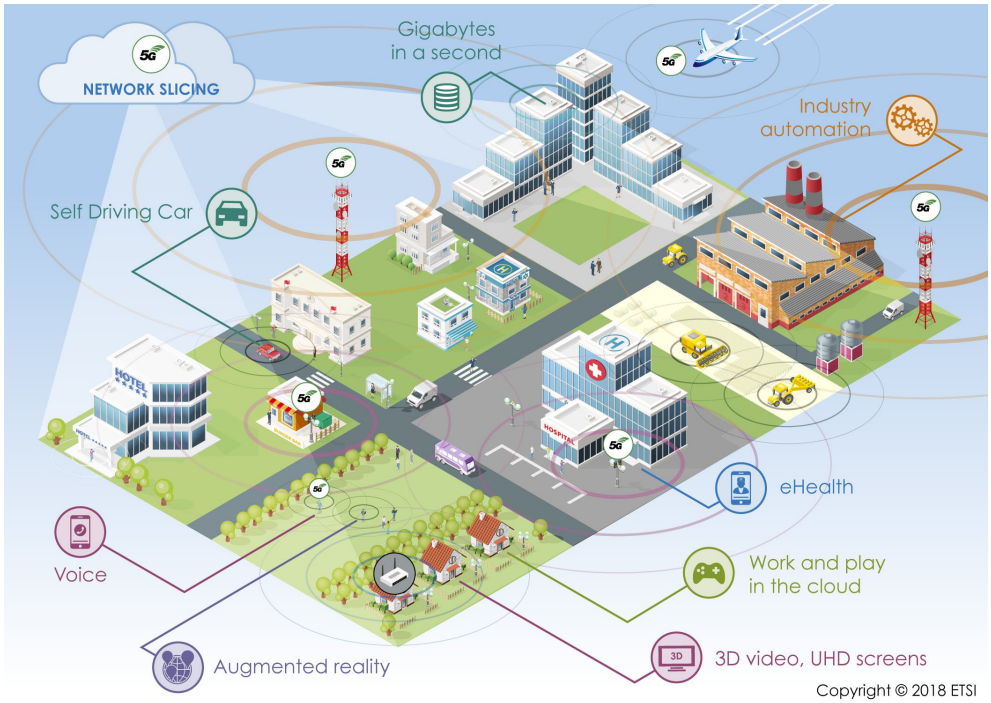


Figure 1.1: Domains and industries that will benefit from 5G [7].

fied these definitions across domains (engineering, IT, environment, biology, business, etc.). For each vertical, we need to decide how to define the resilience of the service provided by 5G and which metrics to use to measure it.

Indeed, with 5G, a vertical customer will be able to ask to supervise its virtual network (slice in the 5G denomination) and thus be able to ascertain whether its resilience requirements are being met dynamically. Upstream, the telecommunication operator will also have contractually committed to a set of metrics for the services rendered to the customer.

To enable such contracts to be signed, it is necessary to ensure, on the operator's side, that the verticals' requirements are achievable. Depending on the scenario, the requirements may vary, as shown in Figure 1.2. The second NGMN white paper [10] detailed a set of new technologies that can enable 5G to "deliver on its promises": Telco-cloud, which is directly linked to the so-called "softwarization" of networks, in other words, the transition from functions performed by physical equipment to purely software functions that can be deployed on any type of IT servers; Mobile Edge Computing, which enables complex calculations to be performed in the network, but as close as possible to users; the concept of autonomous networks, which, for example, minimizes human intervention in launching a service or managing the network.

This concept could primarily benefit from Artificial Intelligence technologies. Besides, the evolution of the network itself, with a new radio network and a new CN, is also encouraging for improving network performance. However, having these technologies is insufficient to ensure that all resilience requirements are met. As the commercialized "full 5G" network is not yet deployed, making the measurement directly on the network is impossible. In the near future, contracts will have to be signed as soon as the 5G service is launched. Operators should have estimated the network resilience before deployments. It is consequently necessary to go through a modeling or simulation phase.

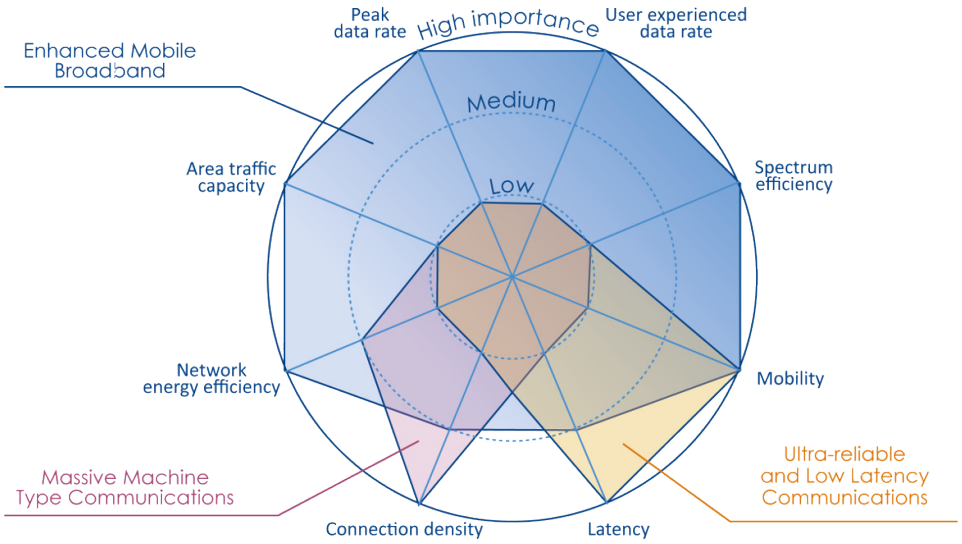


Figure 1.2: Key capability requirements in three main 5G usage scenarios defined by ITU [11]. The radial axes represent different capacity requirements, each of which is classified into three levels: low, medium, and high.

The 5G network is by nature a complex system, whether in terms of the architecture and implemented functions of RAN Radio Access Network (RAN) and Core Network (CN), or in terms of the different planes that make it up [12]: User Plane (UP), Control Plane (CP), data plane, assurance plane, orchestration, etc. The increasing complexity of mobile networks and the need to approach them as complex systems are described in [13]. In this paper, Sergiou et al. reviewed basic models of complex networks from the perspective of communication networks, focusing on their structural and evolutionary properties. Based on this analysis, several complex network models are presented as potential candidates for 5G modeling.

Resilience metrics should also be carefully chosen according to the complex network model. A few examples of resilience metric selection have been published, such as [14], which focuses on engineering systems, and [15], where two metrics are proposed for critical infrastructure networks. As for the 5G modeling in the form of complex systems, this could be inspired by work such as [16] on electrical networks or [17], which deals with interdependent systems. The latter provides a way of modeling 5G through interdependent subsystems. Indeed, other fields also approached the modeling of their networks in the form of complex systems and adopted metrics potentially transferable to 5G, such as in rail transport [18] and water distribution [19]. The former, for example, presents metrics for Transport Network (TN) that can both give indications of the requirements of TN concerning 5G and provide ideas for metrics that can be transposed to telecommunication networks.

1.3 . Scientific challenges

The thesis aims to assess the ability of a 5G network to meet the requirements of verticals in terms of resilience. The whole work includes multiple objectives. First, the vertical usages should be investigated in order to better understand service requirements and threats under specific scenarios. The specification of requirements needs to be agreed upon with vertical industries. Then, a 5G network model in the form of a complex system including all network layers should be proposed according to the selected use cases. All situations and parameters involved in the use case can be presented in such a model. The resilience-related metrics also need to be defined before assessing 5G ability. These metrics can be defined according to specific use cases and risks. Finally, simulation tools should be selected or developed in order to estimate 5G resilience and eventually optimize the 5G network to best meet resilience requirements at the lowest possible cost.

The main scientific challenges of this thesis are as follows:

- To clearly define the notion of resilience. The term “Resilience” is generally used to estimate the system’s capacity to face and adjust to a risk. The notion is evolving, and the definition varies from one domain to another. The term “Resilience” is also closely related to “Reliability” and “Availability”. According to the user case, a distinction should be made between network reliability/availability and service reliability/availability. Some manufacturers are defending new ideas in standardization, such as the fact that failing to meet service requirements demanded by the vertical should be taken into account in service unreliability and unavailability. Therefore, some network performance indicators, such as “Latency”, and “Packet loss”, should also be considered.

- To select the metrics that enable to estimate whether the requirement has been met. We must determine how to quantitatively measure these indicators for the E2E service delivery or for the different layers or planes on which the 5G network is based. Then, we need to convert them into "Resilience" metrics for 5G network assessment.

- To model a 5G network. It will be necessary to determine which elements in the overall architecture play an essential role in relation to the concerned metrics and use cases, and which elements can be neglected in the model. It can be highly dependent on the use case. In order to estimate the roles of the various elements, we need to analyze all the basic components involved in providing the desired service. Sometimes a service can strongly rely on the functions in the UP, while sometimes a service would rely heavily on the functions in the CP. The 5G Core has a service-based architecture, and all Network Functions (NFs) are softwarized [20]. These functions, which used to be physical devices, can, therefore, be instantiated on servers using virtualization techniques or containerization techniques. Besides, the elements in the access and TN, which can also be virtualized, should also be considered in the network model.

- To acquire knowledge and insights related to resilience. The model developed during the thesis should be applied to network simulation and resilience estimation. This mission includes not only the nominal operation of the 5G network but also generating faults or malfunctions, if necessary, in relation to the resilience metrics concerned. It means taking into account certain resilience mechanisms implemented in the 5G layer at the infrastructure or orchestration level. The operational part of the 5G system will thus also be included. Some related problems to be anticipated have been addressed in [21]. Finally, the mission can be extended to adapt the generic model and apply to a specific vertical use case.

Although the work of the thesis is based on 5G networks, it can be extended to 5G and beyond (5GB) or the sixth Generation (6G) networks. Extensions can be made in the model to meet the next-generation communication network's new features.

1.4 . Contributions of the thesis

1.4.1 . The current state of knowledge on 5G resilience

Before looking into the scientific challenges, a first investigation is to assess the complexity of a communication network system, which is detailed in

Section 2.2. In order to have a comprehensive perspective on 5G networks, three major issues are examined:

- The E2E composition of a 5G network for service delivery.
- The vertical structural layers in a 5G network service implementation.
- The orchestration and management of 5G network.

Most network elements and their behaviors were identified during this phase, which is summarized in Paper I [22]. Both E2E and multi-layer perspectives of setting up a network have been studied.

The complexity of 5G network modeling also includes its flexible deployment and dynamic management thanks to its multi-layer architecture and Network Function Virtualization (NFV). This allows a 5G network to change its scale or service delivery route with the environment, the traffic, and the requirements.

Therefore, the two different perspectives, together with flexibility and dynamics in the network, are taken into consideration in the 5G model.

After the context of 5G has been studied, in Section 2.3, various use cases and their requirements are investigated in response to the first two scientific challenge. The resilience requirement can be quite different according to scenario, so it need eventually to be checked case by case.

The threats and risks have been studied to understand how the network performance would be impacted in the presence of adverse events. Indicators such as latency, availability, and reliability are most often addressed by vertical service requirements. The resilience can be described by the value change on these indicators. The resilience loss is introduced to quantitatively measure the network's resilience under a major adverse event.

1.4.2 . A Petri Net-based Model

Concerning the third scientific challenge, a review of different modeling methods is carried out to compare and choose proper modeling tools that can be applied to network resilience assessment. Many works have been done to model a 5G network, but only a few have targeted the resilience aspect, and even fewer have taken into account the complex network structure and network dynamics. Section 2.4 highlights the main results.

After the comparison, Petri Net has been used to develop a 5G network model. Some researchers have applied Petri Net-based models to analyze network availability and reliability. However, the existing literature considered only limited network elements and their behaviors.

In the proposed Petri Net-based network, the containerized NFs are introduced. The components of the NF, microservices, are also considered. Two

dynamic processes, namely, Self-healing and Auto-scaling, are studied for resilience estimation. The main results are presented in Chapter 3, which is based on the work in Paper II [23], Paper III [24] and Paper IV [25].

1.4.3 . Insights related to resilience

The Petri Net model has been generalized to a program on the Python platform. Applied to different scenarios, the program helps acquire knowledge and insights related to 5G network resilience. The main results are presented in Chapter 4.

Paper II [23] and Paper III [24] introduce the self-healing mechanism to the model, explaining how 5G networks can largely improve availability by carrying out rapid a self-healing process in the presence of failure.

Another mechanism, auto-scaling is discussed in Paper IV [25] and Paper V [26]. 5G networks, by dynamically reacting to traffic change using auto-scaling, can alleviate congestion. Thanks to this mechanism, the latency and the packet acceptance rate are improved, and the overall resilience loss is reduced.

The issue of congestion propagation is addressed in Paper IV [25]. Some solutions to mitigate the impact, including prioritization and network isolation, are proposed and compared in Paper V [26].

1.4.4 . Network service resilience assessment for vertical use cases

Two use cases from vertical domains that are able to benefit from the proposed model. The result and discussion of these two use cases are detailed in Chapter 5.

The first is the Tele-action use case from the electric network. This use case can be inspired by the results from Paper IV [25] and Paper V [26] to compare the different resilience performances using different management solutions in the presence of congestion propagation.

The second use case is for the high-mobility users, the trains. High-mobility users are suffering from frequent session changes. Such a scenario focuses on the CP of 5G instead of the UP. This increased the complexity of the model by considering additional functions and their relations in the 5G core. There are new research questions to be answered:

- What is the scope of the system under consideration? How to simplify the large system?
- What are processes to be considered during the train's journey?
- How to optimize the resilience of train services?

These questions are addressed in Section 5.2, which are based on Paper VI [27] and VII [28]. The impacts of different network parameters on service

performance are discussed. An interactive platform to present train service interruption and resilience is developed, which is presented in Appendix A.

1.4.5 . The continuation of the thesis

With the development of the project, new research topics are opened.

The model we developed can be converted or implemented into digital twins, where both the network elements and the management systems are simulated. The idea is to have a digital version of the real 5G system to run testing, including disrupted testing.

Another forward-looking aspect of the thesis is to extend this work to 6G. Although 6G is still in the design and concept phase, the work that has been done has been beneficial for 6G use case simulation and indicator validation. Part of this work has contributed to an ongoing European 6G flagship project, Hexa-X-II. This project leads the way to the E2E system design and the enabling platform delivering novel services for the next-generation of wireless networks. The thesis work will contribute to modeling and simulating the next generation network use cases, engaging in resilient network design.

2 - Resilient 5G network and related work

2.1 . Introduction

5G has appeared in daily life for not a long time, and it has yet to be fully deployed. With the ambition to engage in many vertical industries, the 5G system still needs to develop largely research to prove its capacity, especially in terms of resilience, to be competent for these applications.

Applying resilience analysis to the telecommunication domain is an interdisciplinary challenge. Firstly, 5GB is at the forefront of the telecommunications industry. Before digging into the topic, it is important to understand 5G and how it is introduced to verticals. Secondly, each vertical industry may have different requirements for their services and application scenarios to take into account. Lastly, network modeling requires a good knowledge of mathematics to apply appropriate approaches for modeling such a complex network system.

The 5G network is both static and dynamic. In the design phase, the 5G layout is decided by its targeted usages and is well-fitted for specific services. However, in the operation phase, the architecture will dynamically change as the failures and reparations occur. Network management, in addition, may also take part in changing the network structure manually or automatically according to the environment. It is essential to highlight the elements and their relations that may contribute to system resilience.

The 5G resilience is a relatively vague notion. Indeed, the resilience of vertical services may depend on what kind of threats a service faces and what major indicators are indispensable for the vertical. The resilience evaluation should take into consideration various indicators at a time.

When evaluating the network performance, there are multiple options for mathematical models. The selected model must structure a complete 5G network with proper granularity to capture all the essential features concerning resilience.

To sum up, in this chapter, we try to respond to the following questions:

- In the design and operational phases, what is the complexity of the 5G network that we look into?
- What is the granularity of the model? What elements and characteristics should be taken into account when modeling?
- What exactly are the requirements of verticals? How can they be used for resilience assessment?

- What are the metrics to evaluate the performance and the resilience of the 5G network? How can we measure these metrics?
- Which mathematical approach is suitable to model such complex systems?

2.2 . Complexity of 5G network

2.2.1 . Complexity in 5G system design phase: architecture

Before modeling a 5G network, it is imperative to have a comprehensive point of view of the 5G network and decide what elements are essential and should be considered in the model. Indeed, the term "resilience" can be related mainly to the capacity of a system to plan for adverse events (see Section 2.3 for a detailed definition of "resilience"). Therefore, estimating the resilience of 5G must not overlook the design phase, where the architecture of 5G is conceived.

The 5G promise of a complete networked society with unlimited access to information about anything for anyone demands key features beyond what the current 4G offers [29]. Many differences have been made to enhance the telecommunication system since 4G. As proposed in Paper I [22], the 5G system can be disassembled from horizontal and vertical perspectives.

2.2.1.1 Horizontal architecture

Figure 2.1 presents the E2E horizontal integration of a 5G network. From this perspective, a 5G system includes, in general, terminals, the Next generation RAN Radio Access Network (RAN), the Transport Network (TN), the 5G Core Network (CN), and the Data Network (DN).

Terminals are where a network service starts. Terminals are end-user devices, also called User Equipment (UE). 5G networks are designed to create a new ecosystem for vertical industries, including use cases such as health care, energy, and public transport. The UE in 5G is not limited to smartphones. For instance, vehicles, smart-wears, and IoT terminals are also considered 5G UE. 5G needs to meet the various needs of these users, which is why 5G becomes customized and assigns different networks to different usages. In the first step, these usages are classified into three basic categories by the International Telecommunication Union (ITU) in Figure 2.2:

- enhance Mobile Broadband (eMBB): such as live gaming, where a user transfers a huge amount of data, could benefit from a 5G high-speed network connection.
- massive Machine-Type Communications (mMTC): such as smart factory,

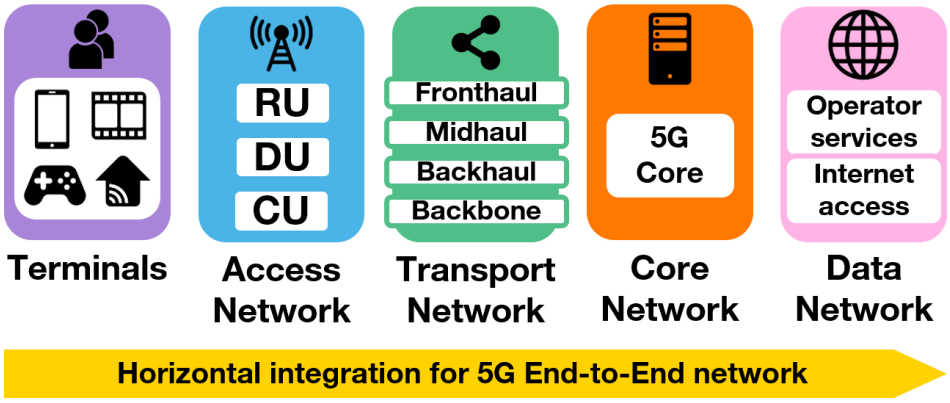


Figure 2.1: End-to-end horizontal integration of a 5G system. Figure from Paper I [22].

where tens of thousands of similar terminals use similar network services.

- Ultra Reliable and Low Latency Communications (URLLC): such as virtual reality and telemedicine, where a user will be guaranteed low latency and high reliability.

Resilience for the UE can be translated into the capacity of the network to provide an acceptable level of service according to usage requirements with and without adverse events.

In the RAN, 5G New Radio (5G NR) technology provides more frequency bands to support various new services with different requirements. However, introducing Virtualized radio access network (vRAN) and Open RAN to 5G is also considered to increase network flexibility. This evolution reshapes the RAN by dividing a Base Band Unit (BBU) into Distributed Unit (DU) and Central Unit (CU). While a DU takes charge of real-time scheduling functions by being placed closer to the UE, a CU is responsible for non-real-time functions. The split of CU and DU functions may depend on deployment scenarios, constraints, and support services. With the idea of Open RAN, the DU, and the CU can be virtualized on multiple platforms and be shared with operators (see Section 2.4 for more details on Network Function Virtualization). DUs and CUs can thus be deployed flexibly, co-located with Radio unit (RU), in edge cloud or regional data center shared by multiple vendors using standardized interface [31].

Some other technologies are also available for 5G RAN. The utilization of orthogonal frequency-division multiplexing (OFDM) allows multiple communication channels to coexist. Thus, it is possible to treat high-frequency and low-frequency bands simultaneously to obtain both higher bandwidth

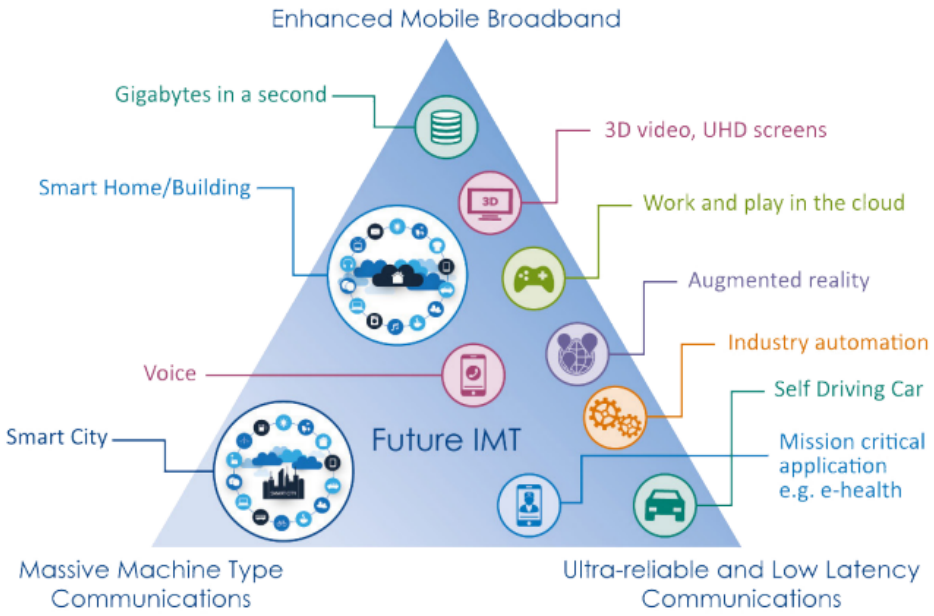


Figure 2.2: 5G usage scenarios proposed by ITU [30].

and broader coverage. Intelligent antennas using “massive MIMO” are implemented, which can further improve network capacity [32, 33]. Besides, the beamforming technology ensures the signal transmission in a specific direction where it is useful to users rather than sending in all directions, such that less interference is created and less energy is consumed. These advanced technologies are out of the scope of dissertation work, thus, are not included at current system-level model.

TN also plays a crucial role in ensuring a good-performance network. TN includes the fronthaul of RU, the backhaul between the base station and CN, optionally a midhaul between DU and CU, and the backbone between CN data centers. Different transmission technologies are used for each part of TN, for example: dark fiber for fronthaul and midhaul with direct connections between the network nodes (RU to DU and DU to CU respectively), WDM rings for backhaul and backbone networks. If Network Slicing is applied to create different virtual networks for different services, it can be based in the first step on VLAN/VPN for each transport segment (called basic soft-slicing or logical isolation between slices), and later on, new technologies like Segment Routing-Traffic Engineering (SR-TE) for enhanced soft slicing with specific performance or designed per type of slice, and in the third step on Flexible Ethernet (FlexE) or Time Sensitive Networking (TSN) for hard slicing where the slices are fully isolated with guaranteed services performance. For resilience purposes, the

IP network, from the Edge to the CN, is doubled, relies on WDM rings, and can react in 50 milliseconds in case of failure.

In the CN, one of the most important characteristics is the separation of the UP functions from the CP functions [34]. UP functions mainly take care of traffic forwarding, while the CP functions manage the authentication, network slice selections, etc. The principal advantage of such separation is being able to scale the CP functions flexibly and independently on UP functions in case of traffic peak and vice versa. Another benefit lies in the flexibility to separately deploy CP functions so that some functions can be deployed in a centralized data center or a distributed one close to the RAN, according to the use case requirement.

The 5G core is targeted to be cloud-native. The container-based virtualization will be largely adopted. A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably in isolated user spaces from one computing environment to another, in any cloud or non-cloud environment [35]. Then the underlying network can be implemented as microservices in these containers. The 5G core network adopts a Service Based Architecture (SBA). The architecture is presented in Figure 2.3. The main benefit of such architecture is that each NF can easily communicate with each other via the application programming interface (API). Thus, these NFs can be both consumers seeking to consume the NF services provided by other NFs and NF service providers providing their exposed services to NF service consumers. Each NF can provide multiple NF services for different NF consumers and can consume NF services from multiple service providers.

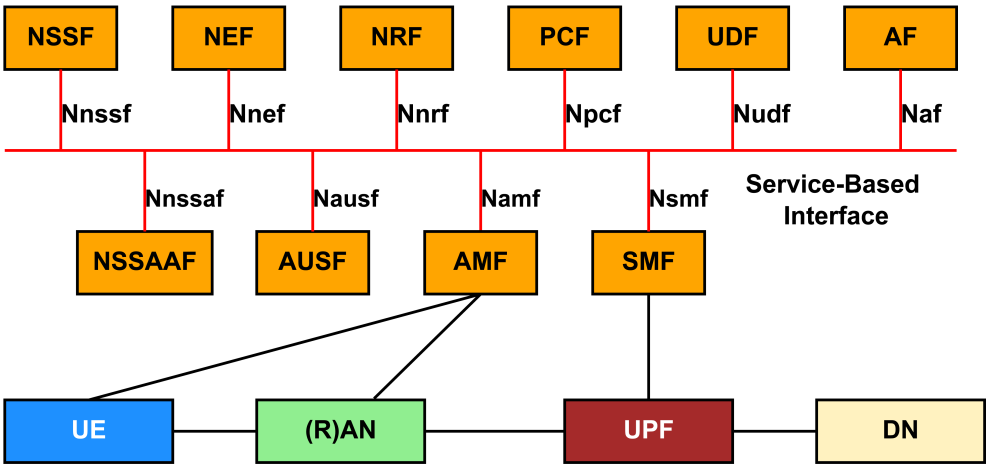


Figure 2.3: Service-based architecture of 5G CN.

2.2.1.2 Vertical architecture

NFV technology reforms the network by separating software from the hardware with the help of virtualization. Instead of being embedded in a physical device, most NFs can be deployed on a virtual platform like a virtual machine (VM) or container. Virtualizing NF enables flexible distribution of hardware resources to improve the service performance and rapid launch of new function instances. By separating software from hardware, an NF becomes a part of physical resources and a set of software applications. This vastly decreases deployment and maintenance costs. All core NFs and some access NFs are potential subjects of virtualization in 5G networks in some scenarios [36]. Software-Defined Networking (SDN) technology introduces a new structure design by splitting the control plane and the data plane. It simplifies and improves network management. Adopting both NFV and SDN, the 5G network becomes more flexible and easily controlled. It can be applied to much more complex scenarios that a 4G system cannot serve.

Figure 2.4 shows another way to present the virtualized network. Instead of emphasizing the relation between VNFs, Figure 2.4 introduces a generic three-tier architecture of a 5G NFV [37]. At the top is the operation layer, with Business Support Systems (BSS) and Operations Support Systems (OSS) to support various E2E telecommunication services. Some processes covered by OSS/BSS include network management, service delivery, fulfillment, assurance, and billing. Lower down is the Network Service and Network Function layer. A Virtual Network Function (VNF) inside this layer will be managed by Element Managers (EMs). EM's role includes security management and fault management for the exposed network function services provided by VNFs. At the bottom lies the NFV Infrastructure (NFVI). Storage and compute resources are two main physical hardware resources that are often pooled. Another physical resource is networking facilities, including routers and links.

The virtualisation layer abstracts the hardware resources and decouples the VNF software from the underlying hardware, ensuring a hardware independent life cycle for the VNFs. For the majority of current deployments, the virtualisation layer in an NFVI comprises a hypervisor to partition physical servers into VMs and a network controller, typically an SDN controller, to help partition the physical network that connects the physical servers into multiple virtual networks interconnecting groups of VMs. While most NFV deployments are still based on hypervisor technologies, container-based virtualisation (a.k.a. Operating System (OS) virtualisation) is gaining momentum and might become the norm for 5G. Containers provide an isolation capability that allows multiple VNF instances to share the same host OS, while virtual machines require a separate guest OS for each VNF instance.

2.2.2 . Complexity in 5G system operational phase: network man-

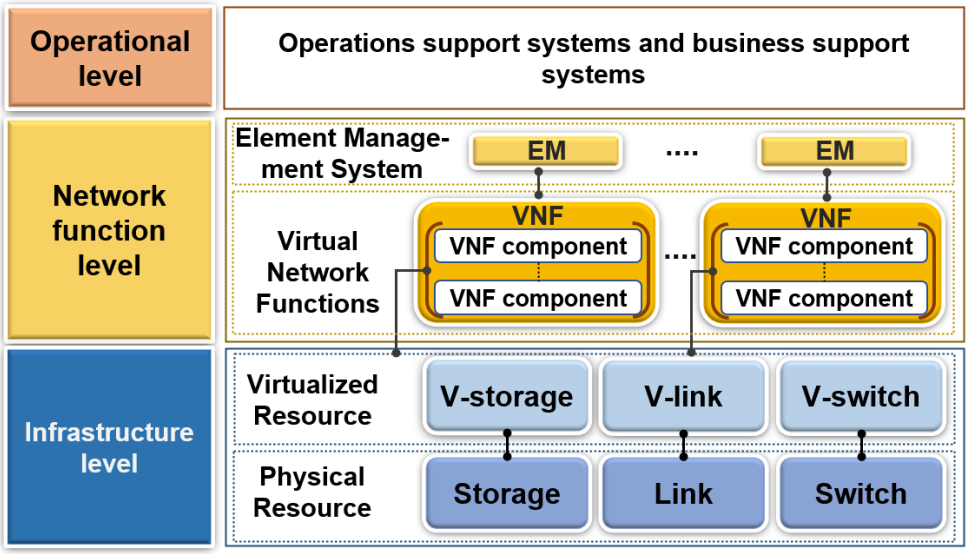


Figure 2.4: Vertical integration of a 5G network.

agement

5G resilience is also closely related to how the network reacts to or absorbs an adverse event (see Section 2.3 for the definition of " resilience"). Thus, Network management is indispensable for modeling a 5G network.

NFV Management and Orchestration (MANO) is in charge of many management and orchestration aspects, such as, the management of NFVI and orchestrates the allocation of resources needed by the Network Services and VNFs [38]. NFV MANO includes three functional blocks. NFV Orchestration (NFVO) is generally responsible for the life cycle operations of a Network Service. NFVO functions can be classified into E2E resource orchestration and Network Service orchestration. The VNF Manager (VNFM) is in charge of the life cycle operations as well as performance, fault, and configuration management of a VNF. Specifically, the management includes instantiation, healing, operation (changing the state), information modification, changing connectivity, scaling, and termination. Each VNF manager serves one or multiple VNFs according to the network design. The third block, Virtualized Infrastructure Manager (VIM), involves all life cycle operations of a virtualized resource. Concretely, a VIM controls and manages the interaction of a VNF with physical and virtualized resources, including compute, storage, and network resources. Similar to VNFM, multiple VIMs can be deployed in the 5G network.

The trend of 5GB network research is to apply automatic and intelligent network management. Kubernetes is proposed to be a potential cloud-native MANO [39, 40] enabler for next-generation networks. When the NFV is carried out by containerization, Kubernetes can be selected as the automatic de-

ployment system [41]. Kubernetes is in charge of deploying containers and managing the life cycle of containers, such as load balancing, self-healing, etc. A Kubernetes cluster is a set of nodes that correspond to a set of worker machines. Kubernetes will deploy pods (groups of one or more containers) on nodes. This thesis work assumes that a pod is equivalent to one container and one microservice application, which is also a component of a VNF. A node is equivalent to a physical machine or a server.

The automatic management can be done by mapping NFV MANO to Kubernetes [42]. In the Kubernetes context, VIM can manage the virtual resources life cycle and expose physical and virtual resources to other management systems. These functionalities can be provided by managed Kubernetes solutions such as EKS in the case of AWS solutions.

At the VNFM level, Kubernetes can manage the life cycle of Pods (sets of containers), and scale them horizontally, vertically, or both. However, the information of a detailed view of deployed virtualization aspects of the associated VNFs is not exposed as expected by ESTI MANO.

Still, with the help of managed Kubernetes solutions, it is possible to realize some functions required of NFVO, such as track scale status, virtualized resources usage, and connectivity to VIMs to manage the resources of VNFs.

In [43, 44, 45], Kubernetes-based NFV MANO systems have been designed and tested. However, machine learning-based solutions are expected to offer more flexible and intelligent network management in the future. Appendix A presents an exploratory study on reinforcement learning methods for solving network scaling problems.

2.3 . 5G network resilience evaluation

Resilience is a relatively new field in system engineering that has drawn significant attention over the last decade. Resilience could be defined as the ability of a system to prepare and plan for, absorb, recover from, and more successfully adapt to adverse events [46].

Recently, European Parliament and Council defined resilience in a directive as the critical entity's ability to prevent, protect against, respond to, resist, mitigate, absorb, accommodate and recover from an incident [47].

The notion of resilience can also be extended to a 5G network. Resilience for a communication network is defined as the ability of the network to face various incidents, maintain an acceptable level of service, and return to normal operation [48]. A resilient 5G network should be able to offer services with high Quality of Service (QoS) all the time regardless of the adverse events. QoS is the ability of a service to comply with quality requirements and service levels as agreed (or targeted) with the end-user.

Inspired from the approach adopted by [49], we decide address two main

issues related to quantitatively assess the resilience. The first is to know what precisely a QoS is and what could be the threat to delivering services with high QoS. The second is how to translate or to map the QoS evaluation into criteria for resilience assessment.

2.3.1 . Use cases and resilience challenges

2.3.1.1 Vertical service resilience requirements

As introduced in Section 2.2, 5G is supposed to support various vertical services, and the QoS requirements vary from one case to another. An analysis of the landscape of 5G use cases can be found in [50]. Some requirements are resumed in Table 2.1.

Table 2.1: Perspectives on vertical industries for 5G, summarized from [51, 52, 53].

Use case	Latency	Reliability	Other requirements
V2X for cooperation	Very low	Critical	Mobility
Massive connectivity for non-time-critical sensing	Not critical	Not critical	Data privacy
Real-time control for remote medication	10-100 ms E2E	Critical	High resiliency
V2X for cooperative farm machinery	10-30 ms E2E	Critical	Low mobility
Intelligent Distributed Feeder Automation	Ultra low	Critical	Isolation
On-train safety device to ground communication	Critical	High	High mobility

The QoS is often interpreted into important performance parameters of the telecommunication system, typically referring to the Key performance indicators (KPI) [54]. Depending on the use cases, the KPIs may include latency, availability, reliability, throughput, etc.

Assessing the resilience of different services requires us to estimate how these KPIs evolve with the changing environment, especially in the presence of adverse events.

2.3.1.2 Threats to network resilience

The resilience requirements are not just related to the KPI but also based on the challenges from the environment of the scenarios.

Table 2.2 summarizes the main resilience threats of the 5G network and their corresponding scenarios.

Table 2.2: Potential threats impacting 5G network resilience.

Threat	Scenarios	Consequences
System failure	Hardware failure, software bug, etc.	Service degradation or service non-delivery
Natural disasters	Earthquake, floods, etc.	Service degradation or service non-delivery
Cyber attack	Malicious users	Information leak, service interruption
Human errors	Wrong system design	service non-delivery
Frequent handover	High mobility usage	service interruption
Policy conflicts	Multiple NFV MANO provider	Conflicting actions

In the 2020 ENISA annual Telecom Security Incidents report [36], 50% of telecom incidents in 2020 are marked as system failure. System failure is a primary threat to the telecom system. It is often due to hardware failure, software bugs, faulty software changes and updates, etc. The main consequence of this type of threat is reducing telecommunication network capacity and service quality if no redundancy or immediate network management is provided.

Human errors are the second frequent cause for telcom security incidents yet the most impacting ones in terms of hours lost [55].

Natural disasters are the third frequent cause to a communication system. They include flood, earthquake, storm, fire, etc [56]. The consequence of these disasters can be a power outage or damage to equipment, which leads to degradation of network service quality or even a loss of network.

Cyber attacks are also unignorable threats to telecommunication systems [57].

There are also many threats related to new verticals of 5G.

The traffic pattern in the 5G network becomes unpredictable. Especially with the appearance of IoT, a huge amount of mMTC users can generate Big Data and create traffic fluctuation, which may congest the 5G network and reduce QoS [58, 59].

High mobility is also a challenging scenario for 5G [60, 61]. Effective mobility management should be carried out carefully when preparing the 5G network for high-speed users. Handover (HO) failure due to high mobility is one of the major threats to the network in this scenario.

The introduction of NFV brings about new threats. For example, different providers of NFV MANO could have policy conflicts when making a management decision [62]. In [62, 63], other threats concerning NFV are also discussed.

More challenges and threats to 5G are overviewed in [64, 65]. Moreover, the challenges may propagate in the network. Due to the complexity of the communication network, these threats can cause cascading failures from components to system level, leading to a further impact on the network [66, 67].

In the scope of this thesis, we consider principally two types of threats. The first type is internal threats, i.e. the system failure, which have a direct impact 5G performance. The second is external threats, such as attacks, abnormal user behaviors, natural disaster, which lead to network overload or failure and have an indirect impact on the 5G system performance. In addition, the propagation of the threat (the Domino effect) is also included.

2.3.2 . Resilience related metrics and KPIs

In order to quantitatively estimate the network service resilience and the related KPIs, we selected several metrics from or extended from standardization:

- End-to-end latency [4]: the time that it takes to transfer a given piece of information from a source to a destination, measured at the communication interface, from the moment it is transmitted by the source to the moment it is successfully received at the destination.
- Communication service availability [4]: percentage value of the amount of time the E2E communication service is delivered according to an agreed QoS, divided by the amount of time the system is expected to deliver the E2E service.
- Communication service reliability [27]: the ability of the communication service to perform as required for a given time interval under given conditions. It can be measured by the Mean time to failure (MTTF) or Mean time between failures (MTBF) of the communication service.
- Network availability [27]: percentage value of the amount of time the network operator can provide E2E service and response to CP signaling messages to any UE by using the 5GB network deployed in a considered area, divided by the total considered time.
- Network reliability [27]: the ability of the communication network to provide E2E connection and response to CP signaling messages to any UE in a considered area. It can be measured by the MTTF of the considered network system.
- Packet transmission reliability [4] (Communication service packet acceptance rate): in the context of network layer packet transmissions, the percentage value of the packets successfully delivered to a given system entity within the time constraint required by the targeted service out of all the packets transmitted.

Some of these metrics are often used as Service-level agreement (SLA). It reflects the performance of a specific service using the network. The others present the overall performance of the network. All aforementioned metrics will be taken into account in the thesis work.

These service performance metrics also contribute to define the resilience. As proposed by Bruneau et al.[68], the resilience triangle can be used to quantify the resilience concept. In Figure 2.5, the resilience loss is quantified by calculating the area of the degradation in the service performance over time. Once the KPI P is chosen, the resilience can be estimated. The estimated resilience loss RL of the network service under a certain incident from time t_i to t_f is given in Equation (2.1):

$$RL = \int_{t_i}^{t_f} [1 - P(t)]dt \tag{2.1}$$

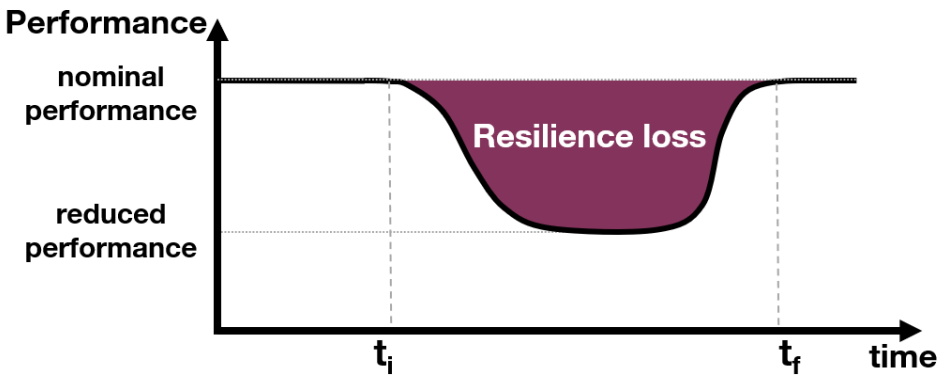


Figure 2.5: The resilience loss represented by resilience triangle.

2.4 . 5G network performance evaluation methods

The telecommunication network is by nature a complex system due to its very diversified service requirements and its heterogeneity in applications, devices, and networks [69, 70]. As the 5G system becomes increasingly larger and acquires even more components, it eventually becomes a system of systems.

In the state of the art, various methods has been be applied for communication network modeling.

2.4.1 . General models for communication network

Before focusing on 5G network, some works have addressed the resilience issue of different kinds of communication networks. Gomes et al. [71] con-

ducted a comprehensive literature survey on network vulnerability and strategies to protect the network against large-scale natural disasters. However, no quantitative method is given for evaluating vulnerability or resilience. Çetinkaya et al. [72] categorized different resilience challenges and built a framework for measuring network performance using packet delivery ratio by simulation. In [73], Sterbenz et al. described a comprehensive framework consisting of a resilience strategy, metrics for quantifying resilience, and evaluation techniques for internet resilience evaluation. They later described a model for multilevel resilience analysis and derived a composition of a multilevel state-space resilience metric in [74]. Similarly, in [75], Tipper highlighted the complexity of providing resilience in multi-layer networks and discussed potential solutions and challenges. Gomez et al. [76] proposed a novel architectural solution and a novel device-to-device communication protocol to enhance 4G-LTE resilience under element and link damage by comparing various indicators, such as connectivity and delay.

As mentioned in the above works, communication networks are already complex. Compared to 4G resilience evaluation, there are more challenges in 5G resilience evaluation. These challenges include a more complex structure and diverse evaluation scenarios, more simulation parameters and performance metrics to be considered [77].

In recent years, there has been an increasing interest in refining techniques and methods for evaluating the performance, reliability, and availability of novel communication systems for both industry and academia [78, 79].

Two main methods can be used to gain insight into the performance of a 5G network. The first is to use a system-level simulator developed based on the standardization of the 5G communication network, which is explained in Subsection 2.4.2. The second method is to abstract the network to build a mathematical model and obtain performance either by model analysis using mathematical formulas or simulation, presented in Subsection 2.4.3.

2.4.2 . Simulation tools for communication network

Various simulation tools have supported 4G networks before the arrival of 5G. With the standardization process of 5G, they gradually begin to support 5G, and some new simulation tools are also developed. Some common simulation tools are presented below:

2.4.2.1 ns-3

ns-3 [80] is a discrete-event network simulator. It is an open simulation environment for networking research. It is mainly used to model Wi-Fi, WiMAX, or LTE and various static or dynamic routing protocols such as OLSR and AODV for IP-based applications.

ns-3 simulator is developed exclusively in C++. However, the optional

Python bindings promise the freedom of script editing. ns-3 also comes with a powerful library and extensions. With newly added modules, namely, 5G-LENA [81] and mmWave Cellular Network Simulator [82], ns-3 can be applied to simulate 3GPP 5G networks. The former simulates 5G New Radio (NR) cellular networks, and the latter evaluates the cross-layer and E2E performance of 5G mmWave networks.

Various research works have been carried out with the help of ns-3. Larrañaga et al.[83] implemented a configured grant scheduling in ns-3 with 5G-LENA and applied it to a case study in Industry 4.0 scenarios. In [84], Koutlia et al. proposed a QoS provisioning support for delay-critical traffic and multi-flow handling. This solution is validated using the ns-3 5G-LENA simulator. Based on ns-3 mmWave Cellular Network Simulator, MilliCar, a Module for V2X Networks for performance evaluation through an E2E full-stack approach, is developed in [85].

However, the main drawback of ns-3 is that it currently uses an LTE core for the CN. The full virtualization and service-based architecture of 5G has yet to be available in the simulator.

2.4.2.2 OMNeT++

OMNeT++ [86] is an extensible, modular, component-based C++ simulation library and framework primarily for building network simulators. It is already used for simulating multiple scenarios, such as massive IoT environmental monitoring [87], Cellular V2X [88, 89], network slicing emulation [90].

OMNeT++ also supports various libraries and tools. Among them, Simu5G [91] provides a collection of models with well-defined interfaces, which can be instantiated and connected to build arbitrarily complex simulation scenarios. Based on Simu5G, Virdis et al. [92] evaluated MEC Deployments in 4G/ 5G NSA/ 5G SA scenarios, Pusapati et al. [93] proposed a simulation framework for NR-V2X communications, Tham [94] developed a test-bed for 5G-based vehicular communication systems. This library provides a full protocol stack of 5G NR, however, the support for 5G CN is still missing. Plus, it has only been applied to particular cases.

2.4.2.3 NetSim

NetSim [95] is a stochastic discrete-event simulator targeted for experimentation and research on networks. NetSim supports the latest advances in 5G, including MIMO, beamforming, SA/NSA modes, HARQ, OLLA, FR1 & FR2, Interference, BLER, Code Block Segmentation, Mobility, Handover and comes with a range of inbuilt example scenarios. It has been applied to multiple usages in recent research. To test an integrated distribution grid protection system using 5G URLLC, Iqbal and Chen [96] used NetSim to calculate the service

throughput, delay, packet loss, and jitter. In [97], NetSim was used to simulate data transmission over 5G to test the communication protocol between Solar Micro-Inverters and the SCADA control System. [98] used NetSim simulation to identify the parameters that significantly impact the 5G network and create a DDoS attack dataset.

However, the underlying C protocol code is highly optimized, making it hard to create personalized scenarios.

2.4.2.4 Riverbed Modeler

Riverbed Modeler [99] offers a comprehensive development environment to model and analyze communication networks and distributed systems. It simulates all network types and technologies (including VoIP, TCP, OSPFv3, MPLS, LTE, WLAN, IoT protocols, IPv6, etc.) to analyze and compare the impacts of different technology designs on E2E behavior.

This simulation tool has been used for many scenarios, such as evaluation of network topology for V2V communication scenario [100], analysis and implementation of packet preemption for time-sensitive networks [101], evaluation of an innovative IoT-based healthcare framework for biomedical applications [102], analysis of handover management [102].

Since Riverbed Modeler is a commercialized software, it could be difficult to be modified or developed to adapt to our scenarios.

2.4.2.5 MATLAB and Simulink

MATLAB and Simulink are largely used to design, optimize, and test wireless communication systems [103]. Mathworks also introduces 5G Toolbox [104] to provide algorithms and applications for the modeling, simulating, and verifying 5G New Radio communications systems. The toolbox supports both link-level and system-level simulation.

Vienna 5G system-level Simulator [105] is also a MATLAB-based simulation tool. It evaluates the average performance of large-scale networks, including user throughput, transmission latency, etc., through Monte Carlo simulations. It has been used to simulate 5G performance under different scenarios, such as UAV [106], drones [107], vehicular applications [108].

One limit of MATLAB is the inconveniences when integrating the program with other languages.

2.4.2.6 OpenAirInterface

OpenAirInterface (OAI) [109] is an open experimentation and prototyping platform to enable innovation in communication networks. OAI implements the full protocol stack to run on a real execution environment respecting frame

timing constraints [110]. It is nearly the most realistic platform compared to the other simulators discussed above. The OAI-5G CN project plans to achieve a full standalone 3GPP-compliant 5G CN implementation, which is currently in progress.

Some works have been done based on OAI. Costanzo et al. [111] used an OAI-based Software Defined Radio prototype to test a network slicing solution for enabling the efficient coexistence of eMBB and IoT services, sharing the same RAN. In [112], a novel prototyping tool based on OAI is presented to facilitate prototyping V2X applications in large-scale scenarios. Bertolini and Maman [113] analyzed the performance in terms of E2E throughput and latency of the X2 handover procedure using an OAI-based implementation.

2.4.2.7 Open5GS

Open5GS is a C-language Open Source implementation of 5G Core and Evolved Packet Core [114]. It supports different communication network features, including 3GPP Release 17 compliant, Handover, IPv6, etc. It has been used for various tests, such as attacking 5G Core/RAN test-bed [115], real-time video conferencing network test [116], cloud-native 5G framework with containerized E2E monitoring [117].

2.4.2.8 Free5GC

Free5GC [118] is an open-source project for 5th generation (5G) mobile CN based on 3GPP Release 15. Free5GC offers an operational implementation of service-based 5G CN, including multiple 5G core VNFs.

Free5GC has already been widely used in different research projects. An experiment to validate the effectiveness of multiple network slicing in providing better performance is carried out in [119]. Chai and Lin [120] used free5GC to realize different 5G core configurations to evaluate dedicated slice performances regarding registration time, response time, throughput, resource cost, and CPU utilization. Chiu et al. [121] designed a cloud-native management and orchestration framework for 5G E2E slicing based on free5GC. In [122], an evaluation of the difference in forwarding performance between the public and private clouds is performed by deploying Free5GC.

2.4.2.9 Comparison of simulation tools

Based on [123, 124, 125, 126], Table 2.3 summarizes the characteristics of these simulation tools mentioned above.

As pointed out in [127], a complete and accurate 5G simulator should be able to incorporate all the diverse technologies. An open-source simulator

Table 2.3: Summary of simulation tools for communication network, by referring to [123, 124, 125, 126].

Simulation tool	Main domain	Programming language	Simulation method	5G support	References
ns-3	Research (open source)	C++ & Python	Discrete-event	Extensions of 5G modules	[80]
–5G-LENA	Research (open source)	same as ns-3	same as ns-3	No 5G SA support for now	[81, 83, 84]
–mmWave Cellular Network Simulator	Research (open source)	same as ns-3	same as ns-3	MmWaves 5G cellular networks	[82, 85]
OMNeT++	Research (open source)	C++	Discrete-event	Supported by extensional tools	[86, 87, 89, 90]
–Simu5G	Research (open source)	C++	same as OMNeT++	support for 5G New Radio access	[91, 92, 93, 94]
NetSim	Industry and research (commercialized)	interface with MATLAB and Python tools	(Stochastic) discrete-event	5G NR including mobility, handover, and 5G Core (AMF, SMF, UPF)	[95, 96, 97, 98]
Riverbed Modeler	Industry & academia (commercial)	C & C++	Discrete-event	not yet for native 5G	[99, 100, 101, 102, 102]
MATLAB	Research (licensed)	MATLAB	-	-	[103]
–5G Toolbox	Research (licensed)	MATLAB	-	5G NR	[104]
–Vienna 5G	Research (Academia license)	MATLAB	Discrete-event & Monte Carlo simulation	various 5G features	[105, 106, 107, 108]
OpenAirInterface	Research (open source)	C++	(hybrid) discrete-event	Support for 5G SA Core in progress	[109, 110, 111, 112, 113]
Open5GS	Research (open source)	C language	Implementation of 5G Core	5G core	[114, 115, 116, 117]
free5GC	Research (open source)	Go language-based	Implementation of 5G Core	5G core	[118, 119, 120, 121, 122]

is always favorable as it is more accessible and safe for development. Programming language is another concern for simulator selection. Libraries and packages from MATLAB and Python can vastly simplify the scenario setup and data analysis. Discrete-event simulation is the most commonly used simulation method. It focuses on the state change of the network. A discrete-event simulation-based simulator is thus a proper option. In addition, a simulator should also support the features considered in the thesis work, such as dynamic network configuration, 5G CN, and handover.

Flexible virtualization of the network, microservice-based VNF, and network and service management (scaling, healing, handover), are closely related to resilience issues. They compose the most important criteria for simulator selection. The support for features such as mm-Wave communication is outside the scope of the thesis work. During the thesis work, the support for service-oriented and microservice-based 5G VNFs in these simulators was still in progress. A mathematical model could be a better choice to overcome the obstacle at the first step.

2.4.3 . Mathematical models for performance evaluation

Various mathematical models have been applied to evaluate network system performance. In this section, these models are classified according to the focusing domains.

2.4.3.1 Availability and reliability models

Availability and reliability for a communication service are already discussed in Subsection 2.3.2. In a more general case, for a system or a piece of equipment, availability measures the ability of a piece of equipment to be operated if needed at time t. And reliability measures the ability of a piece of equipment to perform its intended function for a specific interval without failure. It reflects the probability that the piece of equipment will last at least until time t from time 0.

Exponential distribution is often used for the reliability function of electronic equipment [128]. The failure rate of the item does not change significantly with age. Mean time to failure (MTTF), the inverse of the failure rate, can then be used to describe the reliability of the item. If the piece of equipment is repairable, MTTF and Mean time to repair (MTTR) are used to describe the failure and repair processes, the availability can be deduced as :

$$\text{Availability} = \frac{MTTF}{MTTF + MTTR}$$

We consider first a simplified system composed of one underlying infrastructure server and two same virtualized applications hosted on the server. It can be an abstracted subsystem of the 5G network. The system operates only when the server and at least one of these applications are working.

Availability and reliability analysis for such a system can be performed using different formalisms. Availability and reliability modeling techniques can be classified into three categories [129]:

- **Non-state space models:** such as Reliability Block Diagram (RBD), and Fault Tree (FT).

  Reliability block diagram, a symbolic representation of a system's reliability performance, is often used to model the interconnections among elements. An RBD is drawn as a series of blocks connected in parallel or series configurations, where parallel blocks indicate redundant subsystems or components. An RBD shows the effect of component failures on system performance, and each component is represented by two states: operating or failed [130]. Figure 2.6 shows an example of RBD of a system.

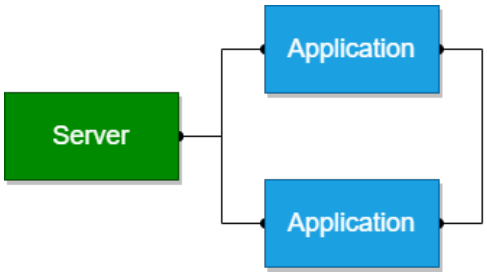


Figure 2.6: Reliability block diagram representation of a system composed of one server and two applications.

  Sinche et al. [131] performed a mathematical modeling of several IoT reliability models based on device and link redundancy using RBD.

  In [132], RBD is used for calculating the system reliability in the context of providing a new scheme to enhance the QoS of a cloud computing system.

  In [133], when developing a reinforcement learning-based solution for a dynamic Service Function Chain (SFC) placement problem, the SFC availability is estimated through RBD models.

  Netes [134] used the RBD model to represent a redundancy and common cause failure in reliable 5GB communication systems.

  Fault tree [135] is a useful analytical tool for the reliability and safety of complex systems. FT provides a structured approach using a graphical tree to represent the essential elements that cause a system failure. It is also used to model the failure conditions of elements and subsystems

in a communication network. Figure 2.7 shows an example of FT of a system.

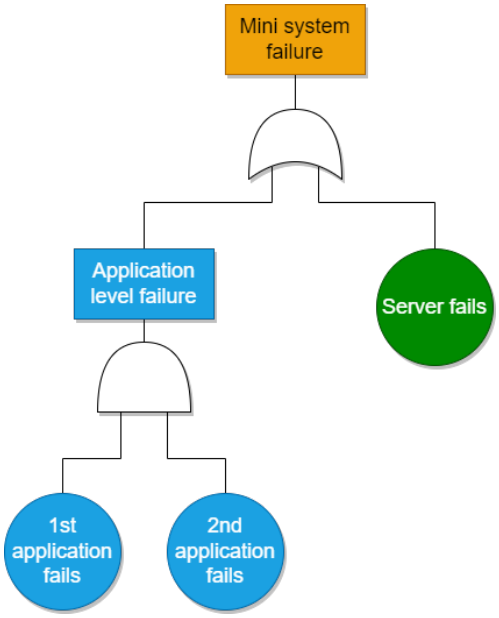


Figure 2.7: Fault tree representation of a system composed of one server and two applications.

In [136], an FT-based reliability model for cloud computing considering both server failures and VM failures is proposed.

Butoi and Silaghi [137] built an enhanced fault tree model for every service in a cloud environment for assessing the health state of a node and performing load balancing in an autonomous manner.

Zhu et al. [138] built a 5G CN fault tree model for availability and reliability analysis.

- **State-space models:** such as Markov Process and Petri Net (PT).

A **Markov chain** enables us to model a dynamical system, the state of which changes over time. Depending on when the state changes, Markov models can be classified into Discrete-time Markov chain (DTMC) or Continuous-time Markov chain (CTMC). Markov models are developed to solve various problems in communication networks [139]. Figure 2.8 shows an example of a Markov chain representation of a system.

In [140], Xing and Shrestha considered a problem of reliability modeling and analysis of hierarchical clustered wireless sensor networks. This paper applies the Markov chain method to compute the reliability of the

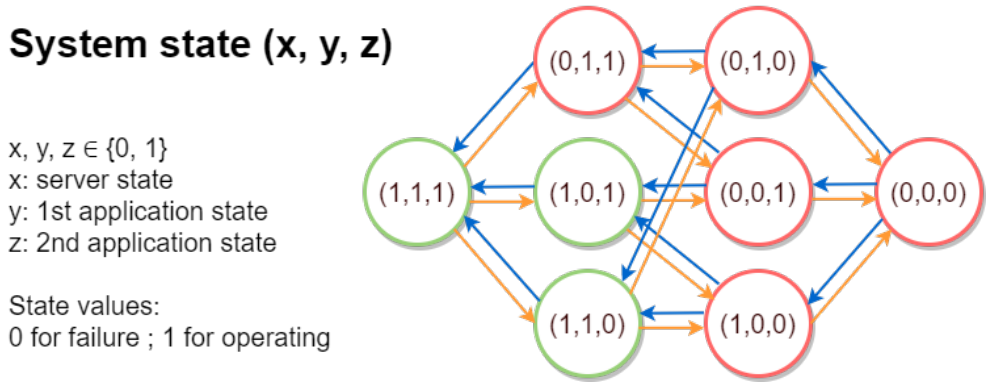


Figure 2.8: Markov chain representation of a system composed of one server and two applications.

dynamic subsystems of the hot spare base station and the cold spare cluster head.

Farooq et al. [141] exploited a CTMC with exponential distribution for failures and recovery times to model the reliability behavior of a base station.

In [142], efficient proactive restoration mechanisms are introduced to ensure service resilience in cloud-native 5G mobile systems. A Markov model is developed for analytical modeling and performance evaluation of the proposed solutions.

Zhu et al. [143] constructed the continuous-time Markov model to capture the behaviors of the edge UPF, especially to compute service transient availability and steady-state availability.

Di Mauro et al. [144] provided a homogeneous CTMC for VNF multi-state model for availability evaluation of multi-tenant service function chaining infrastructures.

In [145], to quantitatively investigate a container-based series-parallel SFCs system, a multi-dimensional semi-Markov process model is explored to depict the behaviors of all functions from suffering from software aging until recovery.

Indeed, for many real-world network components, the failure time and repair time distributions are not easily mathematically traceable, such as Weibull and Pareto distributions. As the exponential random variable is the only continuous random variable with Markov property, exponential distribution approximation of state transition is used when applying such an approach.

Petri Net is also known as place/transition net. It uses a finite set of places to represent the different states of a system and a finite set of transitions to represent the state-changing process. Instead of imposing a time variable on the state change, a Petri Net marking evolves through the consumption and production of resources (tokens) [146]. Petri Net's variants, such as Stochastic Reward Net (SRN) and Colored Petri Net (CPN), are widely used for communication network performance estimation. Figure 2.9 shows an example of Petri Net representation of a system.

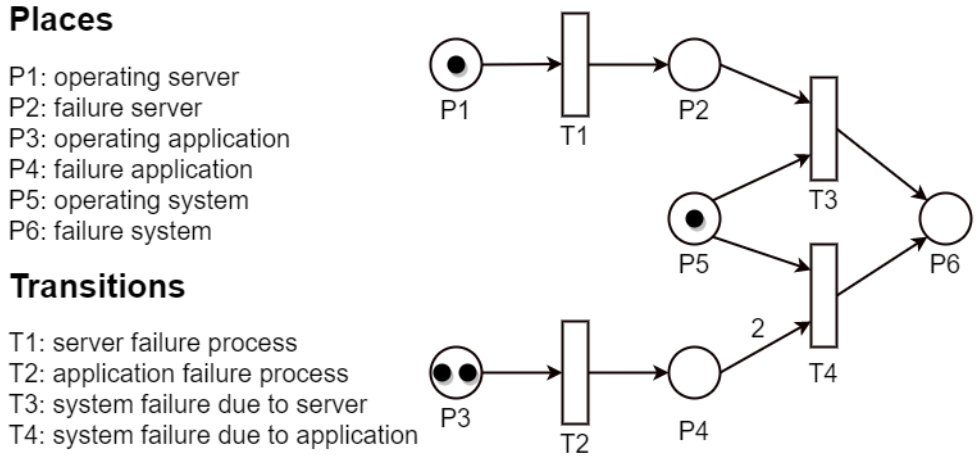


Figure 2.9: Petri Net representation of a system composed of one server and two applications.

Kim et al. [147] proposed an SRN to model and analyze a virtualized system's availability by incorporating various component failure and recovery behaviors.

In [148], a Petri Net model for SFC reliability evaluation is proposed, based on which an SFC reliability evaluation algorithm and an SFC optimization algorithm are studied.

Tola et al. [149] used a Petri Net model to quantitatively evaluate the steady-state availability of the NFV MANO system and identify the most influencing parameters for different deployment configurations. In order to evaluate the performance and availability of the train-to-train communication system, stochastic Petri Nets are proposed by Song et al. [150] to formalize the system. Communication availability and efficiency are considered in the model.

In [151], a generalized stochastic Petri Net model is used to compare a network evolution model using factors that influence the application

availability.

- **Hierarchical models:** It is quite common that multiple approaches are combined to model different parts or levels of a complex system.

  Kim et al. [152] constructed a virtualized system model using a two-level hierarchical approach. Fault trees are used in the upper level, and CTMCs are used in the lower level. Both hardware failures and software failures and application failures are considered in the model.

  Similarly, in [153] proposed a three-level hierarchical model. An RBD is applied to the first level to model interconnections between upper-level block elements. Then, in the intermediate level, an FT formalism is used to model hardware and software failures. Finally, the CTMC formalism is exploited to model subsystem availability.

  In order to understand and analyze the availability of containerized and virtualized systems, Sebastio et al. [154] proposed a two-level model with an FT model at the top level for Operating System and state-space model for bottom-level VM, container instances availability.

  In [155], an FT model is constructed to explain the main elements of an overall 5G-MEC system that may affect a complete system failure. Then, a generalized version of Petri Net is applied to the bottom level for each element composing the 5G-MEC system.

To summarize, non-state space models evaluate the overall system availability and reliability from its subsystems and components, and state space models provide detailed information on the states of each subsystem and component of the entire system. However, when the system grows and evelopes, the state space may explode. Instead of looking for an analytical solution, simulation can be a better choice for availability and reliability estimation.

2.4.3.2 Latency models

Besides availability and reliability analysis, latency is essential for performance evaluation as well. E2E latency is defined in Subsection 2.3.2, which depends on the whole process of packet transmission. In the scope of the work, network service latency is composed of transmission time in the TN, processing time at each VNF (a set of microservices) in RAN and CN, and the waiting time during each process. Other types of latency, such as time spent on air propagation or on a switch, are not considered. Indeed, the E2E latency is computed by adding up the time a packet or a request spends on each network E2E service delivery process.

Transmission time in the TN has been largely studied. Pérez et al. [156] derived a theoretical expression based on trivial geometry calculation for propagation and queueing delay in front-haul traffic in 5G TNs, assuming a G/G/1 queueing model. In [157], delay in 5G Ethernet mobile front-haul networks is addressed, and the suitability of different packet switching mechanisms for Time Sensitive Networks is discussed. Cominardi et al. [158] presented a 5G TN characterization and the expected traffic mixture. A simulation framework based on SimPy is developed to understand the QoS applicability in 5G TNs, in which both transmission delay and queueing are considered. Larsen et al. [159] calculated the front-haul delay by considering transmission, processing, propagation, and switch delays. The packet size is a major variable for transmission delay. In [160], the E2E delay bounds for a mixed fronthaul and backhaul 5G network are studied. The E2E delay is calculated with consideration of data flow.

We consider a whole fiber TN for 5GB to simplify the work. All packet transmissions are point-to-point. The TN is assumed to have high capacity and availability, so no failure or congestion happens when transmitting a packet. Based on these assumptions, the transmission delay can be calculated by distance and light speed, as pointed out in [161, 162]. In the single-mode optical fibers, the transmission delay in 1 km optical cable τ can be computed as:

$$\tau = \frac{1}{v_\phi}$$

v_ϕ is the phase velocity of light in the fiber. The refractive index of light in the optical cable $\eta \approx 1.5$.

$$v_\phi = \frac{c}{\eta}$$

The transmission delay of a packet in 1 km optical cable is thus approximately $5\mu s$.

Processing time and waiting time are often jointly considered using a queueing model in many research works. Agarwal et al. [163] modeled VNFs as M/M/1 queues to solve a VNF placement and CPU allocation problem. In this model, the service rate of each queue reflects the amount of CPU each VNF allocates. They also identified queueing theory as the best tool to model 5G networks, owing to the nature of their traffic and the processing. This work has been extended to fit the 5G MANO framework [164].

In [165], an M/M/1 queueing model is applied to calculate processing and queueing time in the MEC nodes for a 5G network.

Ye et al. [166] developed an M/D/1 queueing model to calculate packet delay at the first NFV node, and they adopted an M/D/1 queueing model as an approximation to evaluate the average packet delay for each flow at each subsequent NFV node. In their work, CPU and bandwidth resources are allocated among different flows at each NFV node.

To enhance the reliability and decrease the amount of resource consumption, in [167], utilization of sub-chains is proposed. The solution is analyzed by using queueing theory by modeling an SFC as M/M/1 and M/M/m tandem network of queues.

In [168], each entity of the proposed network is modeled as an M/M/1 node with a single server, single queue, where the requests arrive at the base station based on Poisson distribution and the service times of the nodes follow an exponential distribution.

The queueing model can also be inserted into other models, for example, Petri Net models. Liu et al. [169] proposed a generalized stochastic Petri Net with queueing integrated to simulate the service reliability of a cloud data center. The queue is used to represent the processing procedure of a service request. In [170, 171, 172] queueing Petri Nets are proposed to simulate the network queueing and congestion processes.

2.4.4 . Mathematical models supporting network management

Proper mathematical models should allow us to estimate network performance and the events related to resilience evaluation, especially the reactions to the changes. The models are supposed to support basic network management as the network reacts to a changing environment.

Two management scenarios are mainly considered in the scope of the thesis work, namely congestion and failure.

2.4.4.1 Network congestion management

Instead of passively waiting for the congestion to disappear, intelligent and resilient network management would react to or even prevent the congestion. This could be done by rerouting, by network scaling with NFV-MANO or by overload control mechanism at 5G level.

Both traffic rerouting and scaling are dynamic problems. Much research work has been done to achieve a better rerouting itinerary without congesting other parts of the network by solving optimization problems, such as in [173, 174, 175]. The focus of congestion management in this work is mainly on auto-scaling since it becomes a new trend in 5G and beyond network management. Instead of changing the traffic route, the network can change its scalability locally to mitigate or avoid congestion. However, the challenge for a state space model could be the variant number of states of the system when the system scale changes.

Rotter and Van Do [176] presented a queueing model for a threshold-based algorithm-controlled UPF instance scaling management. The 5G system is described using CTMC, and the steady-state performance is estimated.

In [177], an analytical model based on Markov chain and queueing model is proposed to test an adaptive VNF scaling algorithm. This model considers

and quantifies the different service capacity issues and the impact of VNF capacities.

By limiting the maximum number of servers that could be activated, the 5G system is modeled with a CTMC by Ortin et al.[178]. A generic auto-scaling mechanism for communication services based on occupation thresholds is analyzed. The impact of the activation delay and the finite lifetime of the servers on performance, in terms of power consumption and failure probability, are also discussed.

2.4.4.2 Autonomous recovery process models

Unlike traditional equipment, the virtualized network has the capacity of self-healing. For example, a containerized application can be relaunched or restarted if a failure is detected. The repairing time of such an entity is thus considerably reduced. However, this changes the nature of repairing time from an exponential distribution, which is common for most manual repair processes. The repair of such an entity may not be easily assumed as a stochastic process, making it hard to model the system with Markov processes.

In [179], Nikmanesh et al. developed a framework for analyzing proactive self-healing in contrast to reactive one in 5GB networks. The framework is adapted to the Markov Decision Process. An exponential distribution approximates the time length of system state change.

In [172], the proposed Petri Net-based model considered the self-healing process. However, the recovery process is assumed to be immediate.

A summary of aforementioned mathematical models is given in Table 2.4.

2.5 . Conclusion

In this chapter, the research background and state of the art are presented.

First, the complexity of the 5G network proposed in Paper I [22] is highlighted. The main challenges of 5G modeling exist both in E2E and multi-layer system integration and dynamic network management.

Second, the resilience requirements and threats are investigated according to different use cases and scenarios. Various resilience-related metrics are studied as well. The above work provides the basis for selecting a suitable 5G network performance evaluation method.

Finally, different network simulation tools, mathematical models, and their applications are examined and compared. Current simulation tools are less flexible and can not model what's needed for 5G. Therefore, we need to start from mathematical models and their extensions and rebuild the necessary simulation algorithms behind these simulators.

The main contribution of this chapter is to analyze the possible modeling

Table 2.4: Comparison of mathematical models.

Model		Target metrics				Remarks	References
		Availability	Reliability	Latency	Acceptance		
Non-state space system-level models	RBD	✓	✓			•Good scalability •No support for CCFs •No support for component interactions	[130, 131, 132, 133, 134]
	FT	✓	✓			•Good scalability •Limited by the gates •No support for component interactions	[135, 136, 137, 138]
State space system-level models	Markov process	✓	✓			•Limited by stochastic process •Hard to be analytically solved	[139, 141, 142, 143, 144, 145]
	Petri Net	✓	✓			•Various extensions •Support for dynamic behaviors •Solved by simulation	[146, 147, 148, 149, 150, 151]
Queueing models				✓	✓	• Model need to be well selected •To be integrated into other models	[156, 157, 158, 159, 160, 161, 162]
Transmission model				✓		•Based on distance and light speed in fiber •To be integrated into other models	[163, 164, 165, 166, 167, 168]

methods for the 5G network regarding its new features. The choice of modeling method should consider the following facts:

- The focus of the thesis is a system-level network model. Detailed communication techniques, such as modulation, are not necessary and can be simplified in modeling.
- NFV is the most important enabler that provides resilience through its dynamic behaviors.
- Different layers in the 5G network, together with their management systems, should be taken into consideration in the model.
- The E2E integration of the 5G network is indispensable when considering service delivery.
- The model for the 5G system should be dynamic, flexible, and easy to be applied to different scenarios.

At an early stage of 5G network deployment, many parameters and the network structure are still undetermined. Using mathematical model can be a more flexible solution than a highly developed simulator. Considering the large number of possible space states, and the complex relations between elements and layers, Petri Net is a good candidate to model 5G networks. In the next chapter, the Petri Net-based 5G network model is explained.

3 - Network system modeling

3.1 . Introduction

The previous chapter addresses the question of determining which elements to model and using which modeling techniques. Petri Net is chosen for 5G network modeling, as Petri Net can capture well the dynamics and complexity of a 5G network. Petri Nets enable a discrete event system of any kind whatsoever to be modeled. However, the classic Petri Net is not powerful enough to capture all the characteristics of such a system. In order to adapt to different events, especially dynamic events, various extensions and variants are developed [180].

Different tools can help realize such a Petri Net modeling process. They can also launch simulations and analyze the performance based on the results. Nevertheless, not all of these tools support the variants of Petri Nets, and they may not be well-tailored to fit a 5G system model. Instead of using a developed tool, it is also possible to focus only on the scope of the thesis work by creating a Petri Net platform dedicated to the desired features, which enables higher freedom for development and optimization.

Building a Petri Net for a complex system is not easy. A hierarchical [181] perspective suggests structuring large Petri Nets as a set of interrelated subnetworks or modules. The transitions and relations of elements become inter-module and intra-module transitions and relations.

For some Petri Net, the state probabilities of a Deterministic Stochastic Petri Net can be obtained analytically rather than by simulation if at most one deterministic transition is allowed to be enabled in each marking [182]. When the structure becomes complex, or the transition enabling is a non-Markovian process or based on logic expression, it is hard or even impossible to solve the Petri Net analytically, such as in the cases of [183, 184, 185]. In such cases, discrete-event simulation can be applied for analyzing large-scale Petri Net models [186, 187].

In this chapter, the Petri Net-based 5G network model is proposed. Firstly, the characteristics and extensions of Petri Net are introduced based on Paper II [23] and Paper III [24]. By integrating different extensions and variants, a Timed Stochastic Colored Queueing Petri Net is selected as the modeling tool. The structure of the complex network model is explained by layers and modules based on these two papers. The relations between different subnetworks are also included. Secondly, different implementation and simulation tools are compared. A model implemented using the CPN tool and a Python-based platform are presented. The former provides the know-how for building a Petri Net with the help of an interactive graphical interface. The latter

provides a flexible and extendable platform for implementing a large-scale and complex Petri Net adapted to the 5G scenarios.

3.2 . Petri Net-based network model

3.2.1 . Petri Net

As well defined in [188], the Petri Net is a 5-tuple $\mathcal{N} = \langle P, T, F, W, M_0 \rangle$, where P is a finite set of places often representing the different states of a system. Places are graphically presented in circles. T is a finite set of transitions representing the state-changing process. Transitions are graphically presented in rectangles or squares. F is a finite set of arcs with $F \subseteq (P \times T) \cup (T \times P)$.

W is a multi-set of arcs $(P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ assigning the weight to inputs and outputs of a transition. M is the marking of the Petri Net graph and $M_0 = P \rightarrow \{m_1, m_2, ..., m_{|P|}\}$, therefore, assigning the initial marking of the graph. Tokens of the graph describe the dynamic and concurrent activities of systems. The marking in Petri Net records the token number of each place.

A Petri Net example is given in Figure 3.1, the initial marking of the Petri Net is $M_0 = \{1, 0, 0, 0\}$. If the first transition t_1 is enabled, it will consume one token from p_1 and create one token at p_2. If enabled, the second transition t_2 will consume one token from p_2 and create one token at p_3 and p_4.

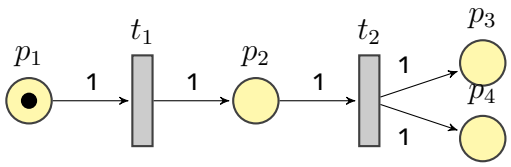


Figure 3.1: A classical Petri Net example.

3.2.1.1 Characteristics of Petri Net

A transition t is enabled when the number of tokens at input spaces is greater or equal to the input arcs' weight. In the example of Figure 3.1, the transition t_1 is enabled. This enabled transition will fire when the event takes place. By firing the transition, one token from the input place p_1 will be removed, and one token will be added to the output place p_2. Then the marking of the Petri Net is modified to $M_1 = \{0, 1, 0, 0\}$. Now the transition t_2 is enabled. Then, if the second event takes place, the Petri Net evolves again. Now we have obtained the final state of the Petri Net with marking $M_2 = \{0, 0, 1, 1\}$.

3.2.1.2 Extensions of Petri Net

The classical Petri Net is not directly applicable to telecommunication systems. Some state-changing processes in such a system could be stochastic, time-dependent, and require additional information. Some extensions of Petri Net can help model a complex network better.

Stochastic Petri Net

One of the most useful extensions of Petri Net is Stochastic Petri Net [189]. It includes a new set $R = \{r_1, r_2, ..., r_{|T|}\}$, representing the firing rate of each transition. This extension could be applied to describe a failure process in the telecommunication network.

Under the original formalism, every transition is formulaic and predictable, i.e., if we have given input places with given input tokens, we have definitive outputs. Nevertheless, in reality, the failure process of infrastructure, for example, is a typical stochastic process in the network system. The failure time can be described by an exponentially distributed random variable. This process in Petri Net will be the case of one input place with multiple output place transitions. The original place represents the normal state of an element. As shown in Figure 3.2, p_1 and p_2 stand for normal and failed states respectively. Transition t_1 is a transition for failure process with a failure rate r_1 and t_2 is the transition of staying in a normal state.

The formal definition of a Stochastic Petri Net is given as [189]:

$$SPN = \langle P, T, F, W, M_0, R \rangle,$$

where $R = \{r_1, r_2, ..., r_{|T|}\}$ is the set of firing rates which could be marking-dependent associated with the transitions.

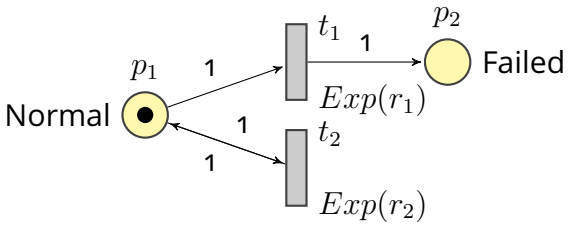


Figure 3.2: A Stochastic Petri Net example.

Timed Petri Net

In order to describe a time-dependent process, for instance, the packet transmission, Timed Petri Nets [190] are introduced. A new set $D : T \to \mathbb{Q}_0^+$ associates each transition with a specific non-negative number representing the time factor.

In the 5G network system, network elements' behaviors could be time-dependent. Some processes, for example, getting a response from a VNF,

can take time to complete. To capture the time feature, we introduce Timed Petri Net.

Timed Petri Nets are similar to classical ones but associated with a time duration d_t with the transition t. The firing of a transition will now take d_t time unit. If this d_t equals 0, the transition is considered as immediate.

A formal description of Timed Petri Net is given in [190]: A Timed Petri Net is a 6-tuple $TPN = \langle P, T, F, W, M_0, D \rangle$ such that:

1. $\langle P, T, F, W, M_0 \rangle$ is a Petri Net
2. $D : T \rightarrow \mathbb{Q}_0^+$ associates each transition with a specific non negative rational number

The function D is also called the duration function. Figure 3.3 shows an example of Timed Petri Net. After being enabled, the transition t_1 takes three time units to finish the firing.

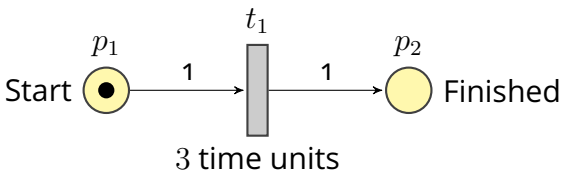


Figure 3.3: A Timed Petri Net example.

Colored Petri Net

Colored Petri Net attaches a value to a token. It indeed distinguishes different kinds of tokens that a place holds. This value could be a number, a word, or a tuple of information. If we model each packet as a token, this colored token allows us to add attributes such as the packet's identifier or the network service type.

A Colored Petri Net is a multi-tuple $CPN = \langle \Sigma, P, T, F, C, G, E, I \rangle$ This extension adds the following items [191, 192]:

1. Σ is a finite set of non-empty types, called color sets.
2. C is a color function $P \rightarrow \Sigma$ defining the type of tokens allowed in a place.
3. G: $T \rightarrow \mathbb{B}$ associates the transition with a precondition g (Boolean expression). The transition will be fired only when g returns true value.
4. E is an arc expression function defined from F into expressions such that $\forall a \in F : Type(E(a)) = C(p)$. p is the place connected to a.
5. I is an initialization function mapping place $p \in P$ with an expression such that $I(p)$ is associated to $C(p)$.

An example is given in Figure 3.4. In this Colored Petri Net, two types of "colors" are defined with $\Sigma = \{'\text{IoT packet}', '\text{VR packet}'\}$. The transition t_1 will be enabled if there is one violet "VR packet" token at p_1 while the transition t_2 will be enabled if there is one blue "Iot packet" token at p_1. After these two transitions, the input token has been selected, and "VR packet" and "IoT packet" tokens are at output place p_2 and p_3, respectively.

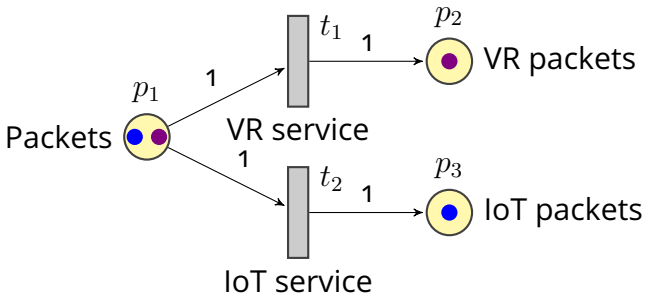


Figure 3.4: A Colored Petri Net example.

Queueing Petri Net

Queueing Petri Net (QPN) combines a Petri Net with queues. More specifically, it integrates queues into places of a Petri Net. For a timing queueing place, when a token encounters a queue, the token will be fired to the queueing place by the input transitions and is inserted into the queue according to the scheduling strategy. It is then scheduled for a service. After completion of its service, a token is moved to the depository, where it becomes available for output transitions of the place. QPN also introduces immediate queueing places, which allow pure scheduling aspects to be described. Tokens in immediate queueing places can be viewed as being served immediately.

A Queueing Petri is a multi-tuple $QPN = \langle CPN, Q, W \rangle$, as shown in Figure 3.5. This extension adds the following items [193]:

1. $CPN = \langle \Sigma, P, T, F, C, G, E, I \rangle$ is the underlying Colored Petri Net.

2. $Q = (\tilde{Q_1}, \tilde{Q_2}, (q_1, ..., q_{|P|}))$ where $\tilde{Q_1} \subset P$ is the set of timed queueing places, $\tilde{Q_2} \subset P$ is the set of immediate queueing places, $\tilde{Q_1} \cap \tilde{Q_2} = \emptyset$ and q_i denotes the description of a queue. q_i takes all colors of $C(p_i)$ into consideration, if p_i is a queueing place, as the case of p_2 in Figure 3.5. Otherwise, q_i equals “null”.

3. $W = (\tilde{W_1}, \tilde{W_2}, (w_1, ..., w_{|T|}))$ where $\tilde{W_1} \subset T$ is the set of timed transitions, $\tilde{W_2} \subset T$ is the set of immediate transitions, $\tilde{W_1} \cap \tilde{W_2} = \emptyset$, $\tilde{W_1} \cup \tilde{W_2} = T$ and w_i is extended from weight function in classical Petri Net. $w_i \in [C(t_i) \rightarrow R^+]$ such that $\forall t_i \in T, c \in C(t_i)$: w_i is interpreted as the rate of a negative exponential distribution specifying the firing

delay due to the color$c \in C(t_i)$, representing the processing time of the service, if transition t_i is a timed transition. w_i can also be a weight specifying the relative firing frequency due to the color$c \in C(t_i)$, if t_i is an immediate transition.

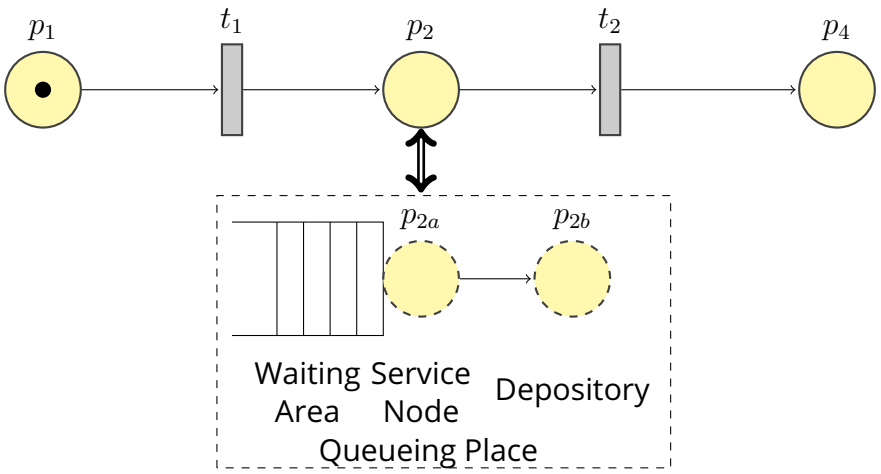


Figure 3.5: A Queueing Petri Net example.

3.2.1.3 TSCQPN model

Combining the extensions as aforementioned, we use a Timed Stochastic Colored Queueing Petri Net (TSCPN) to describe the 5G system. Such a TSCQPN is a multi-tuple: $TSCQPN = \langle \Sigma, P, T, F, W, M_0, Q, C, G, E, I, R, D \rangle$.

3.2.2 . Petri Net-based telecommunication network model

3.2.2.1 5G system high-level structure

The resilience assessment of telecommunications networks requires us to pay attention to the system level and operation level requirements.

At the system level, the 5G network system topology is considered hierarchical, as presented in Figure 3.6. This structure corresponds to the E2E integration of 5G in Subsection 2.2.1. The considered 5G system comprises five physical sites, including four locally distributed sites and a central data center. In each site, NFs are virtually implemented. We assume that VNFs are containerized. Each VNF consists of container-based microservices (equivalent to sub-functions). These microservices have multiple replicas in parallel to share the load. These basic units are managed by a microservice level controller, which is connected to Kubernetes, taking charge of the utilization of the resource pool of the site.

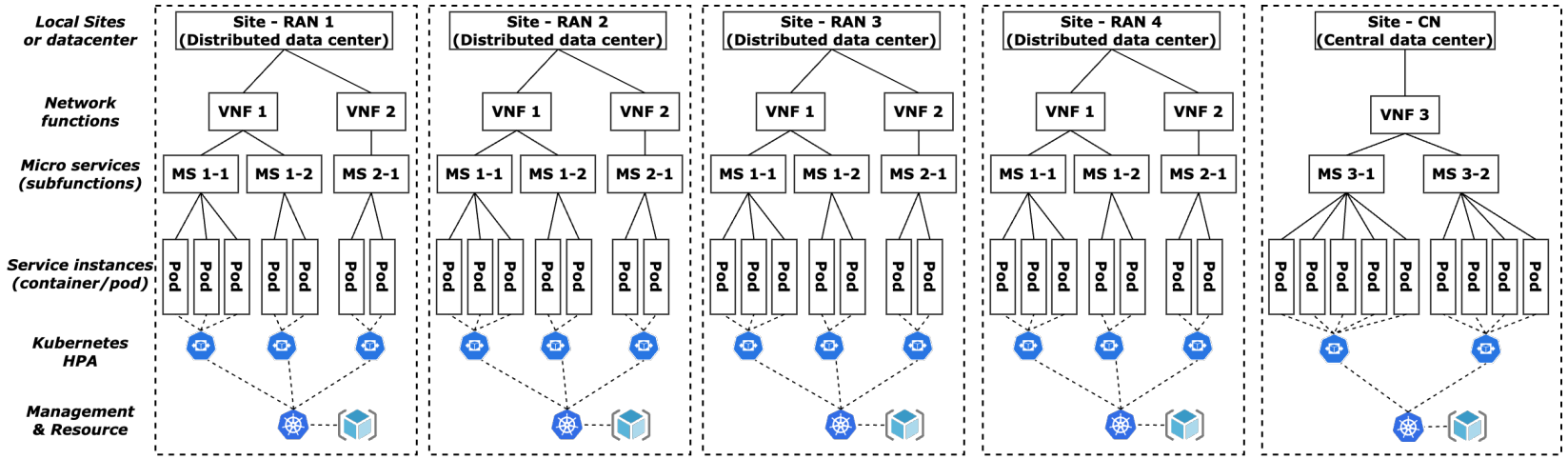


Figure 3.6: 5G container-based NFV hierarchy topology example with four local sites and one centralized data center site. Figure based on Paper V [26].

At the operational level, a set of requirements are demanded for service delivery. It is indeed explained by the E2E integration of 5G service in Subsection 2.2.1. A service packet is processed in 5G networks by SFC, a series of VNFs, and can be further extended into a series of microservices.

A first intuitive assumption for building a Petri Net for an SFC is considering a pipeline of m VNFs. This pipeline can then be modeled in Petri Net with a set of $2(m+1)$ places and $2m+1$ transitions as shown in Figure 3.7.

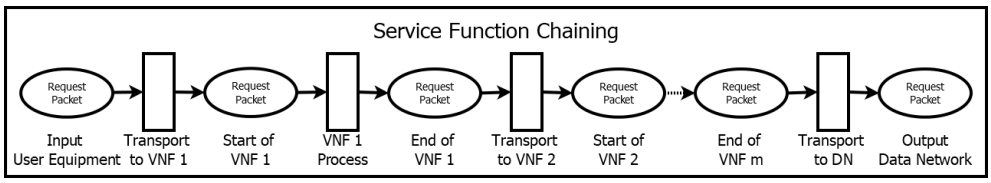


Figure 3.7: Petri Net of SFC example. Figure from Paper II [23].

In fact, when considering a general case with multiple end-users, the SFC may be presented differently. According to the 5G system topology, some VNFs are distributed, and some are collocated. For example, the four local sites in Figure 3.6 take charge of user packets from four cells. Service end-users randomly appear in one of the cells. Before sending packets to the internet, the end-user establishes a Protocol Data Unit (PDU) session, which builds connectivity between the end-user and the network. Once the PDU session is launched, the end-user starts sending packets to the network until the session terminates. These packets follow an SFC containing three VNFs. In our case, the three VNFs can be DU (providing support for the lower layers of the protocol stack), CU (providing support for the higher layers of the protocol stack) in vRAN, and User Plane Function (UPF, connecting the data from the RAN to the DN) in CN. The packets are locally processed at the distributed RAN sites for DU and CU and then at the CN for UPF.

Figure 3.8 shows an exemplified service delivery level Petri Net, including the local site layer and the Network Function layer. Local RAN sites 1-4 and CN correspond to Site - RAN and Site - CN in Figure 3.6. The VNF processes correspond to the Network Functions layer in Figure 3.6. As explained in Table 3.1, p_1 is the starting place, representing the end-users from the cells. Then, they start PDU sessions presented by a sub-Petri Net represented in transition t_1. The established PDU sessions in place p_2 keep generating packets with t_2 during the session's lifetime. These packets in p_3 will then start the vRAN process in the local site where it starts. In a Local RAN (site i, for example), the packet becomes input in place p_{4Ri}, the ingress gateway, and processed in the VNF process sub-Petri Net t_{RiVNF}. After being processed by the VNF, it arrives as p_{5Ri}. As VNFs are processed in order, transition t_{5Ri} sends the packet back to p_{4Ri} to pursue the next VNF, CU, if the packet finishes all processes in DU. If a

packet is processed in both DU and CU, it will be transmitted to CN p_{4C}, where it will pursue processes with UPF. Finally, after being processed in t_{CVNF}, the packet arrives at p_{5C} and then transition t_6 transmits the packet to DN p_6.

Table 3.1: Descriptions of transitions in E2E service delivery. Table from Paper V [26].

Transition	Type	Input token	Output token
t_1: PDU generation	Sub-Petri Net	User	PDU session
t_2: Packet generation	Sub-Petri Net	PDU session	New packet
t_{3Ri}: Radio transmission	Timed	New packet	Packet
t_{RiVNF}: RAN VNF process	Sub-Petri Net	Packet	Packet
t_{CVNF}: CN VNF process	Sub-Petri Net	Packet	Packet
t_{5Ri}: VNF route	Immediate Timed(to CN)	Packet	Packet
t_6: Packet reception	Immediate	Packet	Packet

We consider an E2E service with an ordered SFC. In this system, the UE sends service request packets to the SFC in the network. We assume that every considered packet conveys the same data size, and its SFC always follows the same order of VNFs.

An SFC is a series of VNFs connected by links. The TN is considered a perfectly reliable system. We only consider a fixed time delay spent on the transmission link between UE and VNF, and between different VNFs.

By using hierarchical Petri Net, the 5G system is decomposed into sub-Petri Nets, such as t_{CVNF}, and t_{RiVNF}, which are given in the following sections. Since the exact 5G system structure may vary between operators and service providers, we briefly introduce a generic system model based on our assumptions.

3.2.2.2 VNF level Petri Net

A VNF is, in fact, an application that consists of several microservices. Each microservice takes charge of a set of functionalities of the VNF. When a packet

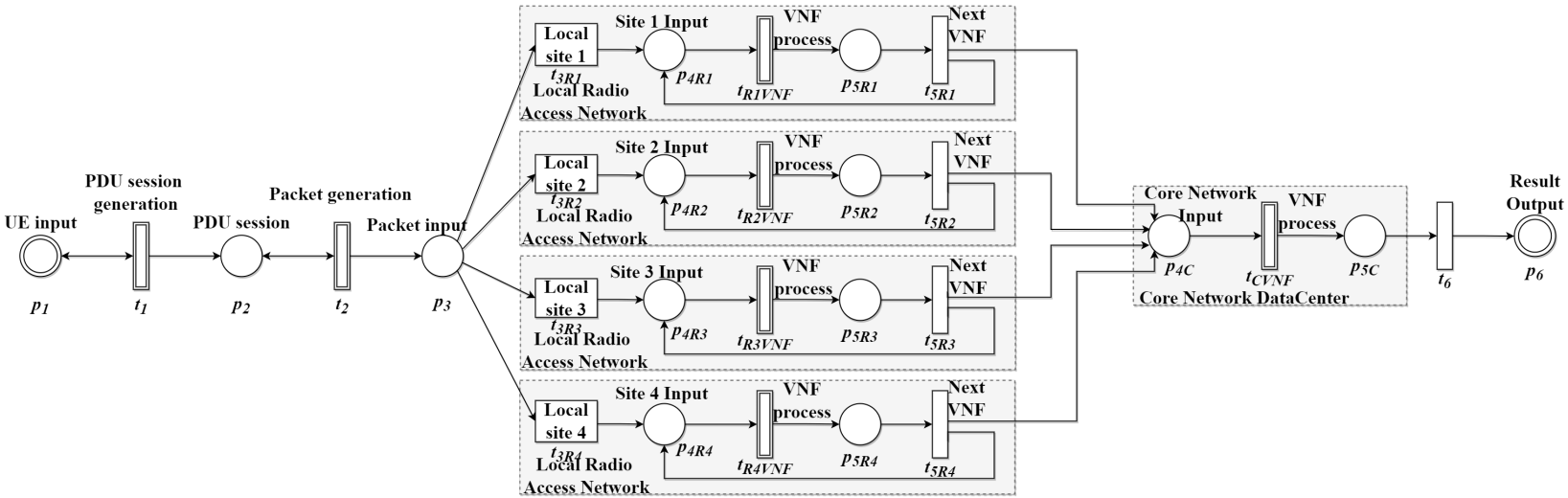


Figure 3.8: Service delivery level Petri Net. Example with four radio cells and one CN data center. Figure from Paper V [26].

visits a VNF, it may consume one or several microservices that a VNF provides. The sub-Petri Net transitions t_{RiVNF}, and t_{CVNF} lead the service packet to the corresponding VNF needed according to its SFC and its PDU session. One of the VNF processes, the VNF A process, is shown in Figure 3.9. The VNF A comprises two microservices, namely AM1 and AM2. In this level, after one microservice is processed, the packet will pursue the other microservice in the same VNF or leave the VNF and move to another VNF, according to the processing sequence. The transitions are explained in Table 3.2.

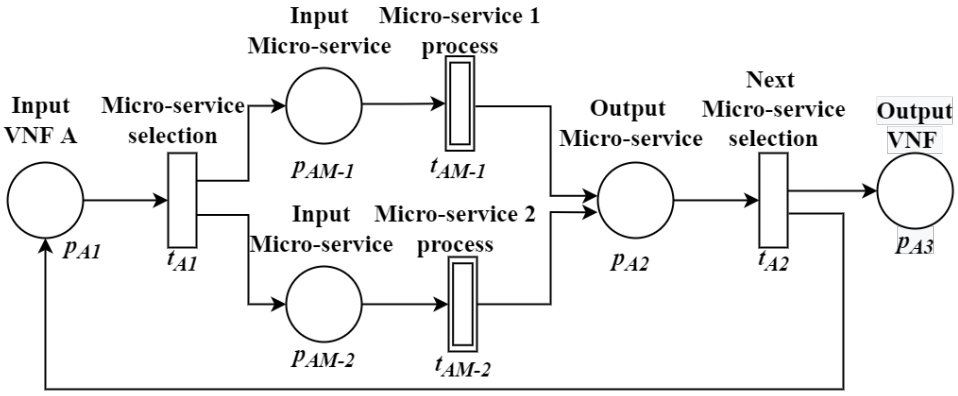


Figure 3.9: VNF processing level Petri Net. Example of VNF A. Figure from Paper V [26].

Table 3.2: Descriptions of transitions in VNF level Petri Net.

Transition	Type	Input token	Output token
t_{A1}: MS selection	Immediate	Packet	Packet
t_{AM-1}: MS-1 process	Sub-net	Packet	Packet
t_{AM-2}: MS-2 process	Sub-net	Packet	Packet
t_{A2}: MS route	Immediate	Packet	Packet

3.2.2.3 Microservice level Petri Net

There exist many ways of virtualization. In the context of the thesis, we choose to model the deployment of these microservices in containers. Kubernetes is

used as the system for automating deployment and managing containerized applications.

Pods are the smallest deployable units that one can create and manage in Kubernetes. A pod is one or a cluster of containers with shared storage and network resources and a specification for running the containers. We assume that only one container is deployed on a pod. For each container, it corresponds to a microservice the VNF supplier predefined. Pods are running on Kubernetes nodes. All these nodes are physical machines.

A queueing Petri Net models the microservice process. A detailed microservice example is given in Figure 3.10.

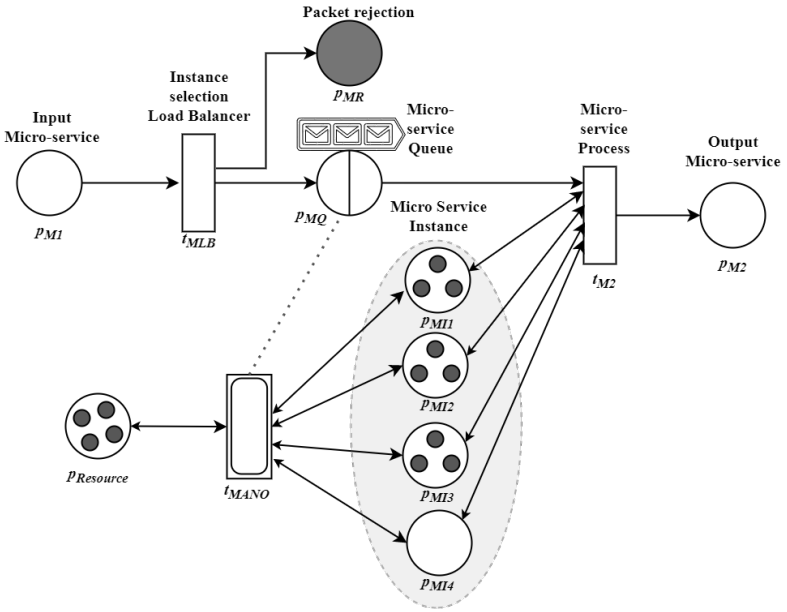


Figure 3.10: Microservice level Petri Net. Figure from Paper V [26].

When a packet arrives at the microservice p_{M1}, it will pass through t_{MLB}, a resource-based load balancer, to different microservice instances. By adopting NFV in 5G, these instances are either VM-based or container-based. In this 5G model, we assume that all NFs are container-based and are managed by the Kubernetes platform. Each container is deployed on a pod. Based on the resource limit of the site, we also assume a maximum of n (depending on the total amount of resources a site possesses) pods that can be instantiated to share the traffic load. A pod is equivalent to a container, requiring specific resources (CPU in our case) to instantiate. The place $P_{Site1Resource}$ provides a shared resources pool to all microservices on the site. When instantiating a pod instance, CPU resource tokens will move to the corresponding pod place. When deleting a pod instance, its resource tokens will return to the site resource pool. To process a packet that arrives at the load balancer, t_{MP} takes

requested CPU resources from the pod with the most CPU resources. This timed transition will bring the packet to p_{M2} and return the resource after a processing time. When there are no available resources in any of these pods, this packet will have temporally waited until there is a new resource. The system may reject a newly arrived packet if the queue is already full of waiting packets. A detailed explanation of transitions and places is listed in Table 3.3 and Table 3.4.

Table 3.3: Explanation of transitions in microservice sub-Petri Net. Table based on Paper V [26].

Transition	Type	Input token	Output token
t_{MLB} Load balancer	Immediate	packet	packet
t_{MP} MS process	Timed	packet	packet
t_{MANO} MS controller	Sub-Net	CPU resource	CPU resource

Table 3.4: Descriptions of places in microservice sub-Petri Net. Table based on Paper V [26].

Place	Token color	Explanation
p_{M1}	Packet	Packet to be processed in MS
p_{MR}	Packet	Packet rejected due to queue length
p_{MQ}	Packet list	Microservice packet queue
p_{M2}	Packet	Packet processed by MS
$p_{Resource}$	Resource unit	Resource pool of the site
p_{MIi}	Resource unit	Pod with a certain available capacity

3.2.2.4 Orchestration and management

The infrastructures that deliver an E2E function are physical links and physical servers. We neglect the TN failure and only consider the time delay since the

TN is considered 100% reliable. We assume that each Kubernetes node corresponds to the Orange Data Center physical machine. Each physical machine has a certain amount of CPU, storage, and network resources. Pods can only be hosted on the server with enough resources.

Kubernetes is an enabler for the orchestration and management of containerized applications. It can automatically handle the network management, including self-healing in case of failure and auto-scaling according to the traffic.

Failure and Self-healing

We assume a pod failure equals a container failure and, thus, a microservice application instance failure. Kubernetes will do Self-healing to terminate the unavailable pods and create new ones to replace them, as shown in Figure 3.11. In this sub-net, places p_{pa}, p_{pf}, p_{pt} represent the "Available", "Failed", "Terminated" states of a pod. Pod is a set of applications. It is often assumed that a pod failure process t_{pf} is described by an exponential distribution $X \sim \text{Exp}(\lambda)$, and with a constant failure rate of $\lambda = \text{MTTF}^{-1}$. The place P_{MANO} represents Kubernetes orchestrator, which launches a liveness probe once in a while to detect the healthiness of pods. This time interval is called health check interval or periodsecond in Kubernetes implementation. If a pod is unhealthy, Kubernetes starts the self-healing by terminating and recycling the resource back to the resource pool. Meanwhile, the transition t_{ps} will start a new pod instance, allocating sufficient new resources.

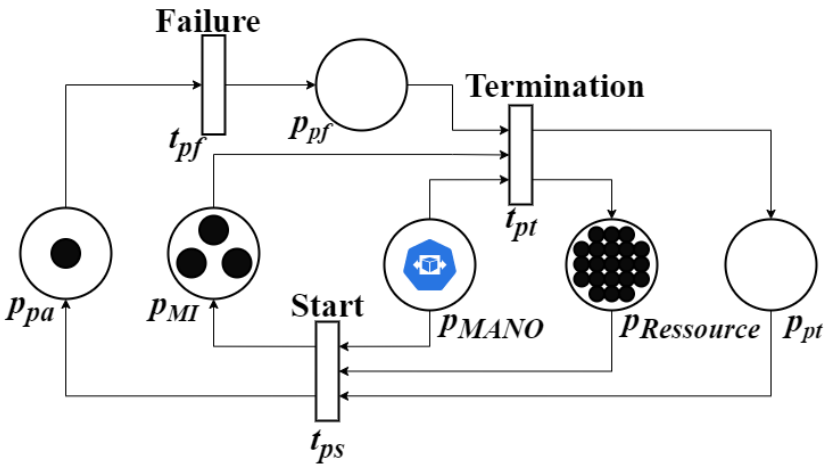


Figure 3.11: Microservice self-healing Petri Net. Example of one pod. Figure from on Paper II [23].

Traffic variation and Auto-scaling

We demonstrate microservice management using a site containing four microservices as shown in Figure 3.12. This sub-Petri Net is divided into four subparts (four microservices) and one shared resources place. Each subpart

can perform scaling-out and scaling-in functions proposed by Kubernetes Horizontal Pod Autoscaler (HPA) [194].

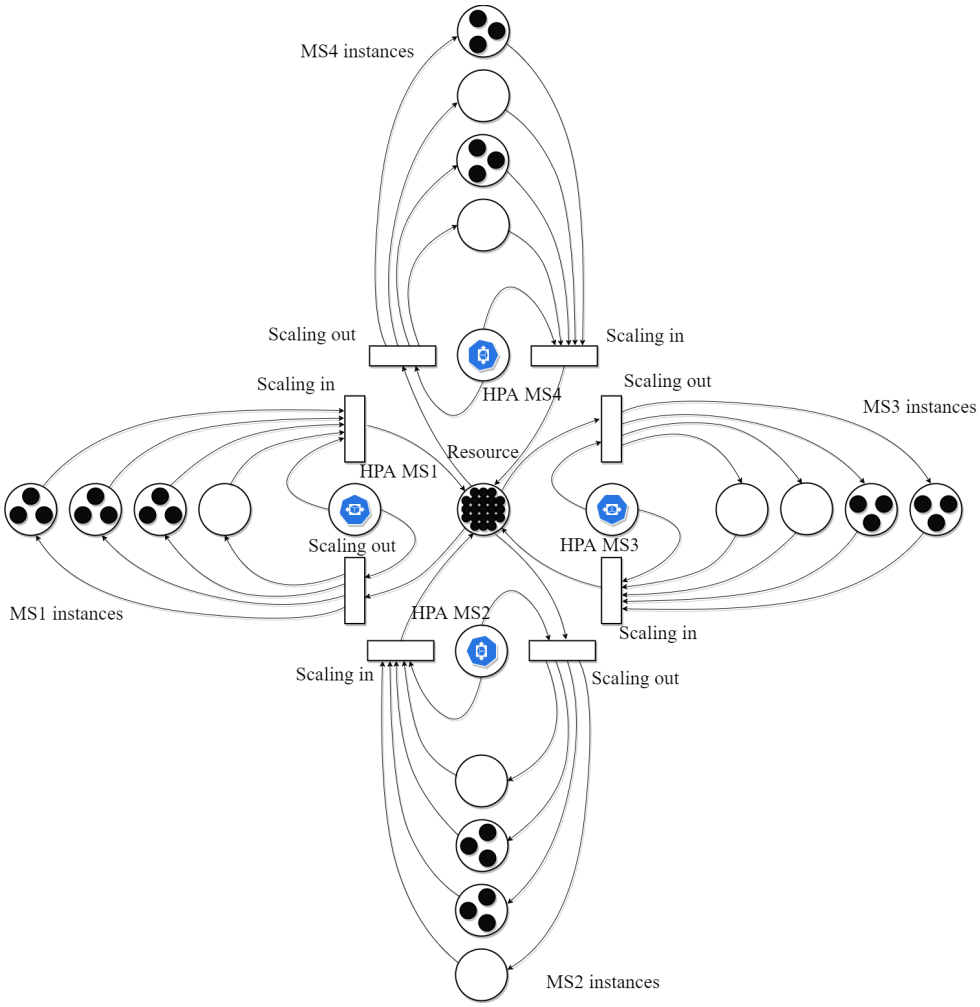


Figure 3.12: Microservice management level Petri Net. Example of a site with four microservices. Figure from Paper V [26].

The built-in algorithm of the HPA controller runs auto-scaling intermittently (the default interval is 15 seconds). The basic algorithm for auto-scaling is proposed in Algorithm 1. This auto-scaling mechanism observes the average pod CPU resource usage metrics intermittently. This time interval is called scaling interval or the sync period in Kubernetes. By applying auto-scaling, Kubernetes updates resource allocation, intending to scale the workload to match demand automatically. The manageable objects of the HPA controller are the pod instances of the microservice in a VNF. A target resource utilization rate is predefined for each microservice, then the controller fetches the CPU utilization metrics and takes the mean utilization value. If this value is outside

a specified range, the HPA controller calculates the desired pod replica number needed to obtain the target utilization rate. If the desired number exceeds the current one, it launches a scaling-out action to create supplementary replicas. On the contrary, if the desired number is smaller than the current one, it removes the unnecessary pods. In general, the goal is to dynamically change and adapt the network scale so that in a light traffic period, the system uses fewer pods to save energy and resource allocation. During a heavy traffic or incident, the system creates more pods to avoid overload and guarantee network service resilience.

Algorithm 1 Auto-scaling algorithm

Input:
CPU metric values: $I = [I_1, I_2, ..., I_n]$, desired CPU metric value V, upper bound:B_U, lower bound:B_L

Output: new replica number: N

1: desired number of pod replicas: $N \leftarrow n$
2: sum of indicator values: $s \leftarrow 0$
3: **for** $i = 1$ to n **do**
4: $\quad s \leftarrow s + I_i$
5: **end for**
6: average of indicator values: $a \leftarrow \frac{s}{n}$
7: desired replica number: $d \leftarrow$ ceil $(\frac{a}{V})$
8: **if** $a > B_U$ or $a < B_L$ **then**
9: $\quad N \leftarrow d$ ▷ new replica number
10: **end if**
11: return N

3.3 . Petri Net-based model implementation

3.3.1 . Petri Net implementation and simulation tools

Different libraries developed for modeling Petri Nets are accessible and facilitate the implementation of system modeling.

In MATLAB, Petri Net toolbox [195] is a software tool for the simulation, analysis, and design of discrete-event systems based on Petri Net models. It supports five kinds of Petri Net models, including Timed and Stochastic Petri Nets. It can also be combined with the 5G toolbox in MATLAB. However, it does not support Colored and Queueing Petri Nets. The firing function lacks customization flexibility.

In Python, SNAKES [196] and COPADS [197] are two powerful libraries for implementing Petri Net models. One of the most important advantages of the using these Python libraries is support for other libraries, especially for result

analysis. SNAKES [196] adopts a high-level object-oriented programming as all transition rules and tokens are implemented as Python objects. It may be considerably more challenging to translate a text-based Petri Net specification into a model in SNAKES. The lack of a graphical interface makes the modeling process complicated. COPADS is a compilation of Python data structures and its algorithms. It has a PNet class to represent a Petri Net model. It supports simulation result analysis. However, it has the same issue of needing GUI support. The program is not well adapted for representing complex systems. Moreover, these libraries define the interactions between places and transitions in a specific way and may not be adapted to the customized transition functions we need. Therefore, developing a new platform in Python is a better way to fit the requirements of modeling a dynamic 5G network.

Some libraries are already developed and embedded in a piece of software. **Oris Tool** [198] is a Java-based software for Timed and Stochastic Petri Nets analysis. It allows customized enabling functions. The available analysis methods can compute transient and steady-state probabilities. However, it may be complicated to apply the tool for colored Petri Nets. **QPME** [199] is a tool for stochastic modeling and analysis based on the Queueing Petri Net modeling formalism based on Java. It supports colored tokens and hierarchical Petri Nets. QPME has not been maintained or updated for years, making troubleshooting hard. **WoPeD** [200] is an easy-to-use, compact tool for editing, managing, simulating, and analyzing workflow nets. It does not support high-level nets or colored Petri Nets. **CPN Tools** [201] is a widespread tool for editing, simulating, and analyzing Colored Petri Nets. It has a graphical interface for building Petri Nets and presenting the relations between components or subsystems. Petri Nets with time and high-level networks are supported. The tool also provides support for simulation-based performance analysis. However, the transition functions are defined using an extension of Standard Meta Language.

A comprehensive comparison of different existing Petri Net libraries or tools can be found in [202, 203]. These libraries or tools alleviate modeling difficulties, especially in the early stages. Based on the above research, CPN was chosen as the modeling tool for its interactive graphical interface and strong compatibility with various Petri Net extensions. A 5G model is built using CPN tools in the first step. This is a very easy tool to get started with. The simulation of a simplified use case is examined to further understand the drawbacks and shortcomings of the tool. Then, in the second step, inspired by CPN tools, we constructed a similar 5G model platform using the Python language, based on the Petri Net model and the experiences of CPN tools, which can be gradually extended to be applied to different 5G scenarios.

3.3.2 . Petri Net modeling using CPN tools

As explained in Subsection 3.2.2, the Petri Net can be structured by subnets according to the architecture of the 5G system. The 5G system can be built in CPN tools following the same idea.

In the top layer, we define the service net with the highest hierarchy as in Figure 3.13.

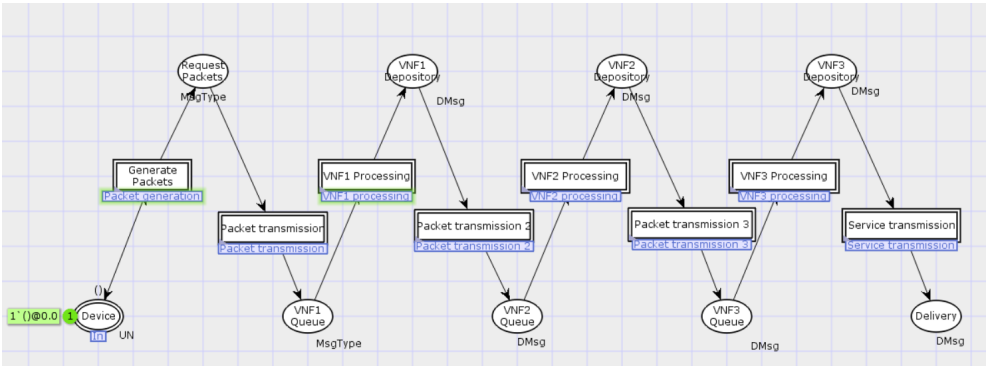


Figure 3.13: Service Function Chain module in CPN tools.

In this layer, the packet transmission starts from the Device. Packets are generated and transmitted to a set of VNFs until they are delivered. The packet transmission is a timed process.

The VNF process is represented by Figure 3.14. The VNF in the figure contains two microservices. In this module, a “States” place collects microservice information and interact with the Kubernetes orchestrator when it launches a detection probe to get resource utilization rate or liveliness of pod information for auto-scaling or self-healing mechanisms. For each packet arriving in the VNF, it will queue up before the microservice it needs. The packet will leave the queue and be processed only when at least one corresponding microservice pod has enough free resource to service it. The resource will be reserved to serve the dedicated packet during the process and will become available when the process is finished.

In this sub-Petri Net, a packet will be processed in different microservices. When the packet demands a microservice, it enters a queue to wait for an available microservice pod. Then, it will be processed by a pod with available resources. After finishing the microservice, it will go to the next microservice or leave the VNF. At the same time, the network elements are dynamically changing during the process.

The management system takes charge of element failure surveillance and reparation. Figure 3.15 shows an example of pod failure. The lifetime of each pod follows an exponential distribution with the same failure rate. In Figure 3.15, the MTTF of a pod is 40 days. Figure 3.16 shows the pod self-healing process. Once having detected a pod failure, VNFM, provided by Kubernetes

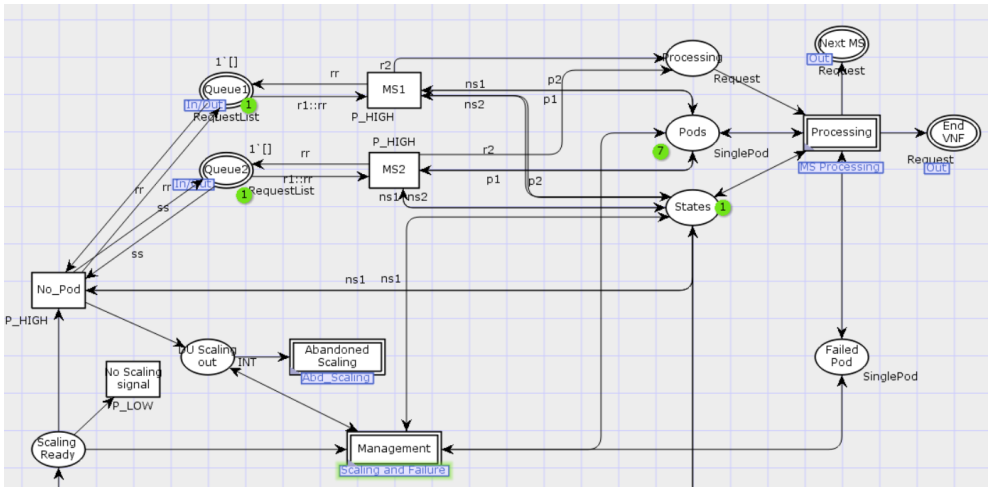


Figure 3.14: VNF process module in CPN tools.

orchestrator, will remove the failed pod and start a new pod instant to replace it. This new pod will become available after the preparation time to restart.

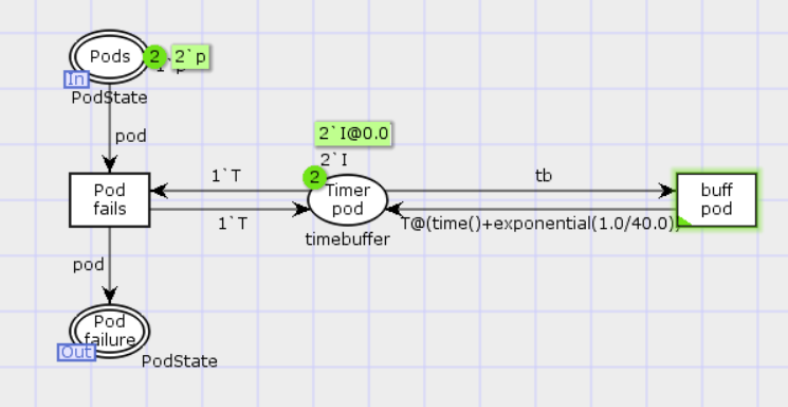


Figure 3.15: Pod failure module in CPN tools.

When the traffic drops, the scaling-in mechanism can be triggered. The scaling-in process of the Petri Net module is depicted in Figure 3.17. Once the scaling decision is made, the least used pod will be selected and be gracefully terminated. Then, the VNF state information will be updated.

When the packet number increases, the scaling-out mechanism will be activated. As presented in Figure 3.18, the new pod will be instantiated on this node server if a node can provide enough resources. The new pod goes through a preparation process before it finally becomes available.

In this thesis work, a 5G network model comprising 3 VNFs is developed

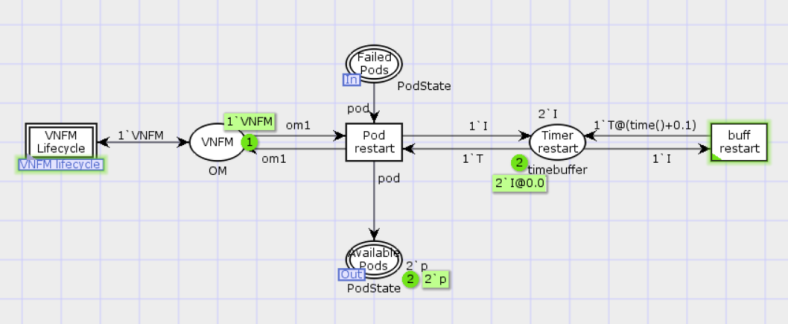


Figure 3.16: Pod self-healing process module in CPN tools.

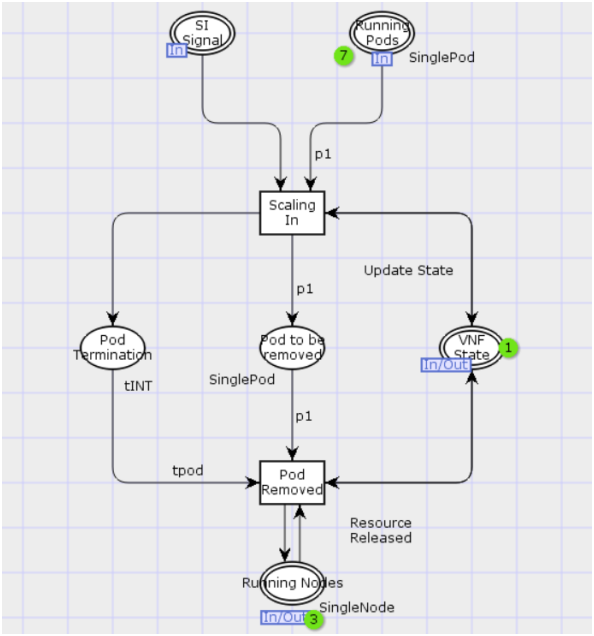


Figure 3.17: Pod scaling-in process module in CPN tools.

using CPN tools. The model shows both the relations between different elements and the dynamic behaviors of the system. However, when assembling these subnetworks together, the model becomes complex. This slows down the simulation and increases the difficulty of modifying the model.

Another limitation of CPN tools is the programming language for expressing the relations between entities. It is hard to optimize the simulation only by using SML code. When launching the simulation, all the aforementioned processes work simultaneously. The expressions at a concerning transition will be checked at each time step. When simulating thousands of thousands of packets, the simulation becomes even slower.

In CPN tools, the way to create and modify a Petri Net is by using its GUI.

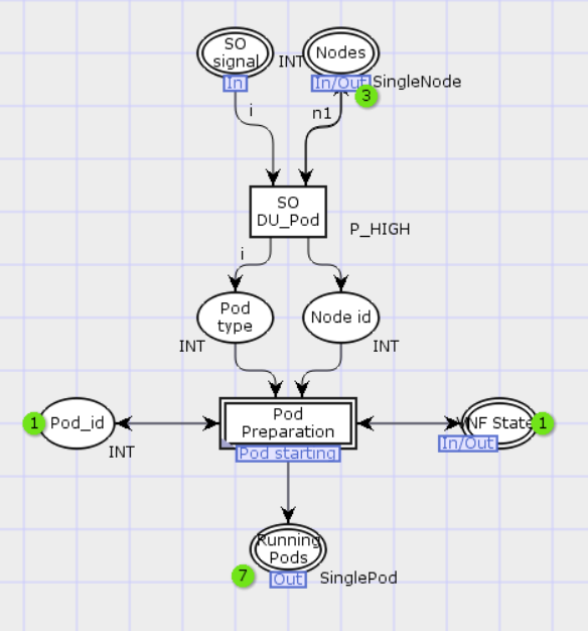


Figure 3.18: Pod scaling-out process module in CPN tools.

Consequently, the Petri Net in the graphical interface becomes unreadable due to the massive arcs of transitions already created.

Therefore, in the second step, a Petri Net-based modeling platform is developed using Python to overcome the disadvantages of CPN tools.

3.4 . General 5G model implementation in Python

3.4.1 . Object-oriented model

Inspired by other Petri Net model realizations in Python, we decided to keep the idea of object-oriented programming when developing the 5G model.

Instead of designing classes for the places and transitions of the Petri Net, we create classes for network elements. In this way, the tokens of network elements become objects in the program. The token colors, representing the types and features of network elements, become attributes of the object. The places in the Petri Net, representing the states of elements, also become the "state" attributes as instance variables. The queue place of the microservice process becomes an attribute of class "MS". A transition, including timed and stochastic transitions, becomes a callable method of the instance representing its major input place. Figure 3.19 shows a summary of classes in the program.

Although this program retains almost all of the information, including the Places and Transitions of the TSCQPN, its structure differs from the Petri Net

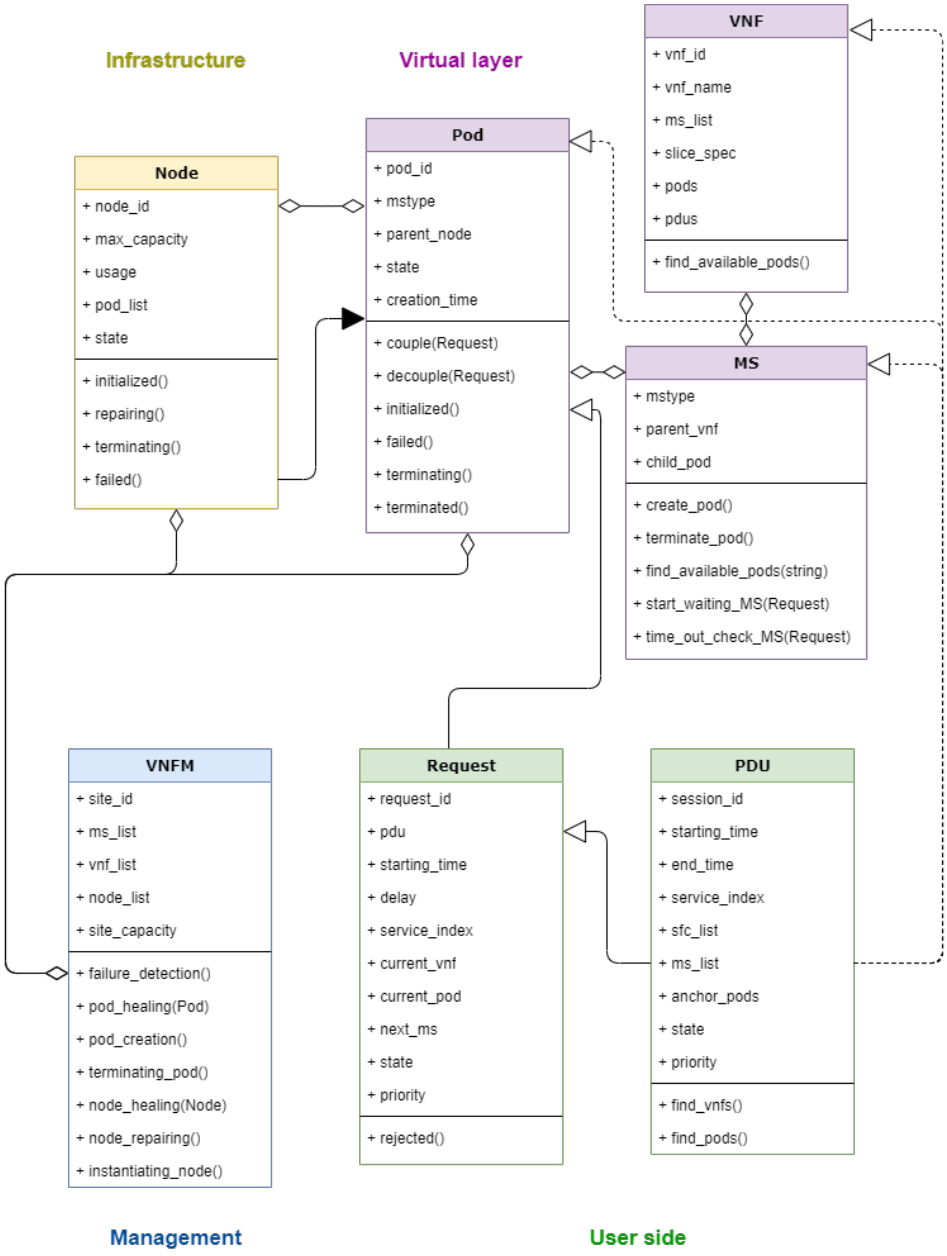


Figure 3.19: Classes and their attributes and methods in the 5G network modeling program.

libraries including CPN Tools as it is object-oriented. For example, the transition of "pod termination" can be triggered as a result of a scaling action or a healing action. In Petri Net, these two events create two different sets of input places of "pod termination" transition. However, in the Python-based modeling platform, these transitions are the same method of the pod class. As a result, this platform is much simpler.

3.4.2 . Discrete event simulation

The discrete event simulation is applied to get the Petri Net-based model running.

In Python, we use SimPy [204] as the process-based discrete-event simulation framework. SimPy is an asynchronous event dispatcher. All the timed or stochastic transitions in the Petri Net can be transformed into methods where new events are generated. Since these events are not immediate, they will be scheduled at a given simulation time. Events are sorted by priority, simulation time, and event id.

The platform based on the SimPy framework is divided into several modules, as depicted in Figure 3.20. Before the network starts working, an initializing module will create and initialize the default network setup with a certain number of available nodes, VNF, and pods. When a 5G network is operating, multiple end-users transmit data. In the PDU generator module, each user generates one or several PDU sessions. Once the PDU session is established, it starts generating request packets until the end of the session. The packet process will generate processing events to be scheduled in SimPy. Besides, the failure of the network can also be scheduled in parallel. Auto-scaler can be an additional module to generate intermittent checks for scaling decisions, which are also scheduled events.

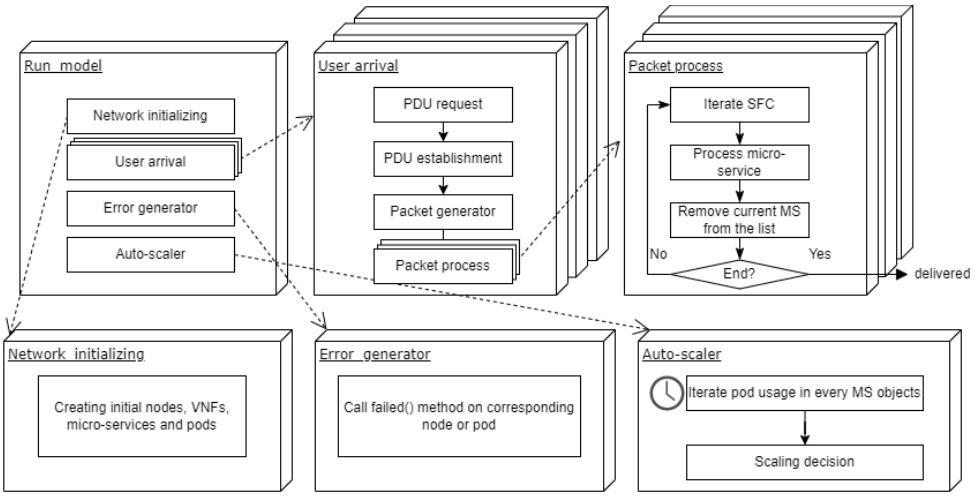


Figure 3.20: Different modules in the 5G network modeling program.

The simulation will start with the “Run model” main module. Different modules can be integrated into the main module according to the scenario. In the train service, for example, a dedicated module is added for dynamically changing the anchoring VNF according to the environment (see Appendix A for more details).

3.4.3 . Monte Carlo simulation

For a small size problem, analytical methods are applicable for performance evaluation. However, when number of components increases, the size of the state space grows exponentially. Plus, the network is not a simple series or parallel component structure. The component number varies with time due to the stochastic nature of the events. The Monte Carlo method provides the most obvious alternative in such scenarios.

In fact, Monte Carlo simulation is a statistical method for performance analysis. The performance can be approximated by taking the empirical mean of the independent samples from a large number of simulations. The service performance can be acquired from the result of delivered packets. The system performance can be obtained from the result of the state evolution of the network elements.

3.5 . Conclusion

In this chapter, the details on 5G system modeling contained within Paper II, Paper III, Paper IV, and Paper V are presented. The main contribution of the chapter is to present the proposed 5G model formalism and its implementation.

Different Petri Net extensions are explained and selected based on the characteristics of the 5G network. The TSCPN proposed in Paper II and III is extended to TSCQPN, where a queue model is added. This brings a more complete formalism.

The implementation of Petri Net in Paper II and Paper III is based on CPN tools. The interactive interface facilitates the model development but it is limited by the inflexibility of the coding language, and its capability.

The implementation of the Petri Net-based general model in Paper IV and Paper V is realized by object-oriented programming in Python. The various classes and modules make the modeling platform capable of launching simulations under various network setups and scenarios.

The CPN tools provide a first insight into 5G system modeling and allow quick implementation of a 5G system, which can already be applied for evaluating some simple scenarios. The general model for a 5G network system takes into account the dynamics and various relations between network elements. In the next chapter, the applications of these two implementation

methods are presented.

4 - Resilience knowledge and insights from the model

4.1 . Introduction

As the 5G model implementations have been carried out, it can then be used to estimate network resilience under different scenarios. Before considering a real vertical use case, in this chapter, we apply the model to different generic scenarios with different network architectures and network managements to estimate performance and acquire knowledge and insights related to resilience from the results.

Different network architectures are considered to cover different usage of 5G systems. In Section 4.2, A VNF-level model is used to evaluate the performance of individual VNFs. In Section 4.3, an SFC model simulates the behavior of the entire service delivery process. Section 4.4 proposes a larger network model including multiple sites. This architecture considers users in large areas containing multiple cells and enables user mobility.

The proposed 5G model can be applied to one or multiple network services. In Section 4.4, two services co-exist in the 5G system. The model also support different manners to isolate them in order to protect the reliability-sensitive service or improve the overall resilience of the 5G network.

The resilience performance evaluation can be carried out in both long and short timescales. A long timescale scenario is presented in Section 4.2, where random failures on the network elements are injected into the model to get an overview of network availability for providing valuable guidance in the design phase of the network. These random system failure are rare events, so a long timescale simulation is necessary. The short timescale, with simulation duration in seconds, is presented in Section 4.3 and Section 4.4, where a specified major threat is injected into the model to get the resilience loss during the adverse event. The major threat, such as the traffic flow change, is an abrupt and drastic phenomena, and therefore requires a short timescale simulation.

4.2 . The effect of self-healing

4.2.1 . Scenario introduction

A first system performance estimation is made by looking at the virtualization and infrastructure layers without mapping them to network communication services to test and validate the proposed models. One single VNF, which includes the microservice applications in the virtualization and physical

resources, can be a good example to showcase. The metric used in this scenario is the availability of the network to provide this single VNF for service. The aspect of latency and resilience loss will be discussed in the complete service delivery scenarios in later sections.

In this scenario, only random system failures are considered threats. There is no specific focus on a particular adverse event. The system failure of a network can be classified into physical layer failures and virtual layer failures as in Table 4.1.

Table 4.1: Network failure classification.

Fault element		**Causes**	**Consequences**	**Actions**
Physical layer	Server	Physical resource damage	Degradation on service performance	Reparation or service migration to new servers
	Switch	Physical damage	Failure on the request routing	Using backup switches or rerouting
	Link	Physical damage	Failure on request transmission	Using backup link or rerouting
Virtualized and logical layer	VM or container	Bugs or malicious attacks	Failure on VNF	Relaunch or migration
	VLink	HTTP connection failures HTTP/2 DoS attacks	HTTP error response	Using backup HTTP2 connections
	MANO	Software bugs or attacks	Failure on network management	Manual reparation

We consider a system with one VNF composed of two virtual microservices, and we assume these microservices can be deployed by the Kubernetes platform on a Data Center with three available servers. Each microservice contains exactly one container, and only one container is deployed on a Kubernetes pod. All pods are deployed on nodes that are physical servers.

Two types of failures, the physical failure on servers (nodes) and the software failure on microservice containers (pods), are chosen as the main risks to the VNF sub-model. The two kinds of failures occur randomly. The failure times of these two failures follow exponential distributions. The architecture of the considered VNF is shown in Figure 4.1.

When a failure occurs, the VNF performance degrades, or even the VNF fails. Indeed, each microservice may have multiple pod instances in order to provide services for users. Then a microservice can be seen as a binary

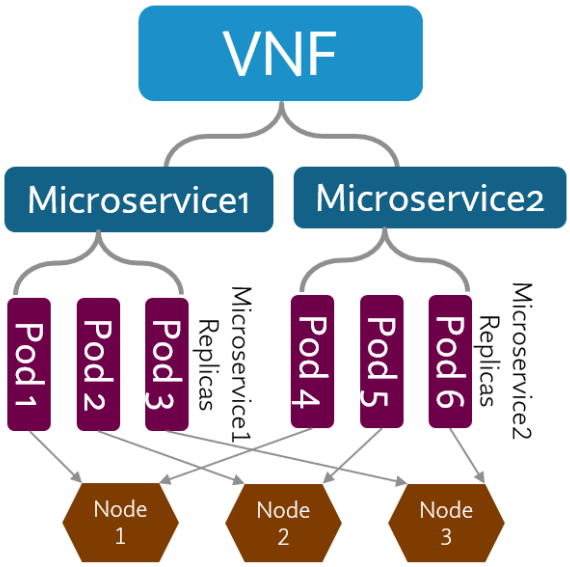


Figure 4.1: Architecture of the considered VNF.

k-out-of-n:G system[1]. It requires successfully operating k instances out of n total components. The number n in the scenario is set to three. The value of $k, 0 < k \leq n$, depends on the microservice's load. Take Microservice 1 in Figure 4.1 as an example, it contains three pods, so $n = 3$. If Microservice 1 is designed to require two pods to sustain traffic in normal state, then $k = 2$. It is a 2-out-of-3:G system. If three pods are required to support traffic in normal operation, then $k = 3$. It is a 3-out-of-3:G system. If any one of the pods fails, then the Microservice 1 will be considered unavailable to provide compliant service.

The pod repair process is automatic and is managed by Kubernetes. Kubernetes throws a liveliness detection probe to check the running status of a pod at every health check interval. If the probe fails, Kubernetes terminates the pod and creates a new one as presented by the sub-Petri Net in Figure 3.11 in Subsection 3.2.2.

Regarding a node failure, the self-healing process is slightly different. A node is a physical server. A node cannot be terminated. The technical team will manually repair the failed server, and afterward, the repaired server will become a free server available for use in the Data Center. A node failure also leads to failure of the pods deployed on the node. Those pods will be replaced by new ones on other available nodes.

All related VNF parameters are given in Table 4.2.

A microservice is considered available at time t if the number of working

[1]a binary k-out-of-n:G describes a system of n components that works if and only if at least k of the n components work

Table 4.2: VNF parameters. Table from Paper II [23] and III [24].

Parameter	Value
Pod failure	MTTF = 1258 hours [205]
Pod termination time	30 seconds (fixed value)
Node failure	MTTF = 8760 hours (exponential)
Node repair	MTTR = 0.5 hours (exponential)
Average time for pod instantiation	5 seconds (exponential)
Average time for node (re-)creation	1 second (exponential)
Node capacity	3 pods per node
Data Center Capacity	3 servers
Self-healing probe periodsecond	0 (immediate), 2, 5, and 10 seconds

pods equals or exceeds the desired replica number k. The uptime of a microservice is the duration of time that a microservice is available. Then, the availability of a microservice i can be calculated as:

$$A_i = \frac{\text{microservice } i \text{ uptime}}{\text{total simulation time}}$$

A VNF is considered available at time t if both the two concerning microservices are working. The availability of a VNF can be calculated as:

$$A_{VNF} = \frac{\text{VNF uptime}}{\text{total simulation time}}$$

4.2.2 . Analytical solution

An easier way to obtain an analytical solution to validate the models and simulations is to consider the case of a 3-out-of-3:G system for each microservice. In this way, the VNF is working only if all these six pods are working. Since a node failure will also cause pod failures, the state of a node also contributes to VNF availability. Usually, the load balancing will force the pods to be evenly distributed across the servers. However, when a node fails, the failed pods will be relaunched immediately on the other two working servers. When the node is just repaired, it is initially empty, then the node's failure impacts a pod only when a new pod has been deployed on it because of the redeployment due to a failure of itself or of the node where the pod is initially deployed.

Then, there are two main contributors to VNF failures, pod and node failure, impacting VNF. We assume that these two events are independent.

The availability of pod can be computed as:

$$A_{pod} = \frac{MTTF_{pod}}{MTTF_{pod} + \overline{T_{detection}} + MTTR_{pod}} \tag{4.1}$$

The average detection time $\overline{T_{detection}}$ is approximated as half of the periodical health check interval[1]. The pod's MTTR, $MTTR_{pod}$, is the average time for pod instantiation.

When only considering the failures that impact pods, the availability of the node can be computed as:

$$A_{node} = \frac{\widetilde{MTTF}_{node}}{\widetilde{MTTF}_{node} + \overline{T_{detection}} + MTTR_{node} + T_{creation}} \tag{4.2}$$

Indeed, the only time duration that concerns a pod failure is the time to detect the node failure $\overline{T_{detection}}$ and the time to instantiate the implicated pod on other nodes $MTTR_{pod}$. The $\widetilde{MTTF}_{node}$ is the modified MTTF of node, representing the expected (average) time T_{up} between the moment the pods that failed due to the node failure are redeployed and the moment the node fails again with running pods. It can be approximated by using the following formulas:

$$\begin{aligned}\widetilde{MTTF}_{node} = &\underbrace{\mathbb{E}[T_{repair,node} + T_{creation} - T_{detection} - T_{instantiating,pod}]}_{\text{node recovery}} \\ &+ \underbrace{\mathbb{E}[T_{failure,node} + \widetilde{MTTF}_{node} | T_{failure,node} \leq T_p]}_{\text{node fails without pod}} \\ &+ \underbrace{\mathbb{E}[T_{failure,node} | T_{failure,node} > T_p]}_{\text{node fails with deployed pods}}\end{aligned} \tag{4.3}$$

$$\begin{aligned}\widetilde{MTTF}_{node} = &\underbrace{MTTR_{node} + \mathbb{E}[T_{creation}] - \overline{T_{detection}} - MTTR_{pod}}_{\text{node recovery}} \\ &+ \underbrace{\int_0^{T_p} f(t)(t + \widetilde{MTTF}_{node} + \overline{T_{detection}} + MTTR_{pod}) \cdot dt}_{\text{node fails without pod}} \\ &+ \underbrace{\int_{T_p}^{\infty} f(t) \cdot t \cdot dt}_{\text{node fails with deployed pods}}\end{aligned} \tag{4.4}$$

$\widetilde{MTTF}_{node}$ is divided into three parts in Equation (4.3) and Equation (4.4). The first part contributes to the recovery process from the moment the implicated pods are redeployed to the moment the node is repaired and recreated.

[1]For a pod or a node, as $t_p \ll MTTF_{pod}$ and $t_p \ll MTTF_{node}$, the distribution of $T_{dectection}$ can be considered uniform. Indeed, for whatever failure happens in the d-th detection interval $]t_d, t_d + t_p]$, $P(T_{dectection} = t) = P(t_{failure} - t_d | t_d < t_{failure} < t_d + t_p) \approx \frac{1}{t_p}$.

The second part represents the case where the node fails without any pod deployed. The node will be repaired (with the full node repair process) and then start again from the newly deployed state. The third part represents the case that the node fails with working pods so that the failure will impact the overall VNF availability. T_p is the average time from when all six pods are working to when one pod fails. $f(t)$ is the probability density function of node failure. The two cases of node failing before pod deployment and of node failing after pod deployment are also depicted in Figure 4.2.

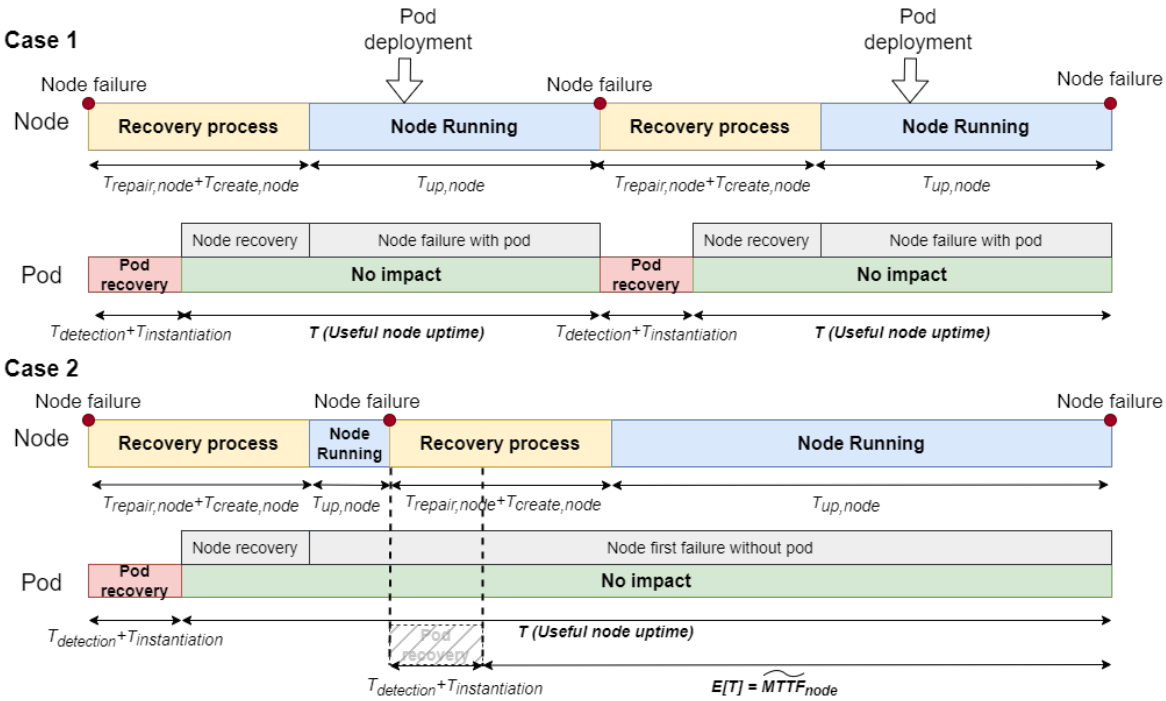


Figure 4.2: Two cases of node failure. Case 1: the node fails after pod deployment. Case 2: the node fails before pod deployment.

The failure of a node that impacts the VNF availability will be mitigated once the failed pods have been restored on other nodes. The VNF is back to available even when the failed node undergoes a recovery process.

For the case of a VNF with 3 out-of- 3:G microservices, the VNF is available when all the pods are working. Since the load-balancing rule is applied to deploy the pods, the pods only work when all nodes work. As the time duration for repairing pods and nodes are vastly different, the two failure processes are assumed independent but contribute equally to the VNF failure.

Figure 4.3 is the Fault tree representation of VNF in this case. A VNF is available only when all nodes and pods are working.

We assume that these pods and nodes have similar behaviors, respectively. Then, the availability of the VNF composed of two 3 out-of- 3 microser-

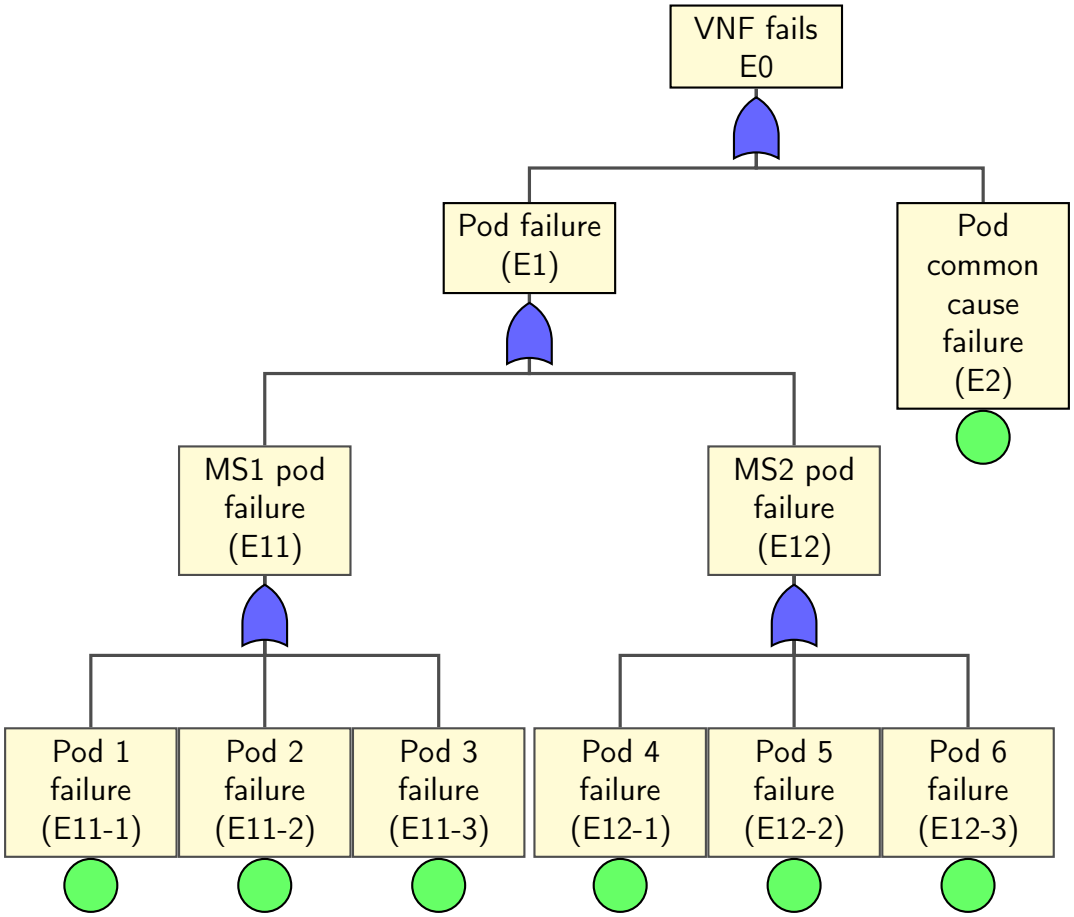


Figure 4.3: Fault Tree representation of the VNF including two Micro services.

vices can be approximated by:

$$A_{VNF} = (A_{pod})^6 \cdot (A_{node})^3 \tag{4.5}$$

4.2.3 . Simulation results

In the first situation, the effect of self-healing detection frequency, the heath check interval t_{HC}, on system availability is studied. We assume that the two microservices are from the same VNF supplier and are managed by the same Kubernetes Master (pod liveliness detection and node liveliness detection are synchronized and done at the same time). The numerical solution from analytical results is compared with the simulation results from CPN tools and a Python program based on SimPy platform.

We simulate the VNF behavior over 50 years. The average value of microservice uptime and VNF uptime over 20000 simulations is taken as the final result. After 20000 simulation iterations, both the results from CPN tools and the SimPy-based Python program converge well. It took half to two hours

(depending on detection interval) to run these 20000 simulations in CPN tools on a computer equipped with Windows 10, 2.10 GHz CPU, and 8 GB memory. Indeed, the computation time is proportional to the number of pods and inversely proportional to the health check interval (i.e. periodsecond). It took only a few minutes for the Python program.

The result is shown in Table 4.3 and Figure 4.4. We compare the overall VNF availability for health check interval varying from 0 to 10 seconds. The longer the probe periode, the lower the overall availability. The availability drops from 5 nines[1] to 4 nines[2] by changing immediate detection to 10 seconds. Thus, the telecommunication network can consume less energy while satisfying the availability requirement by wisely optimizing the health check interval if allowed, according to this result.

Table 4.3: Analytical and simulation results comparison.

Detection T_{HC}	0 (immediate)	2	5	10
Analytical	99.9992911%	99.9991494%	99.9989367%	99.9985823%
CPN tools	99.9993523%	99.9992258%	99.9990361%	99.9987198%
CPN tools LB[3]	99.9993520%	99.9992255%	99.9990358%	99.9987193%
CPN tools UB[4]	99.9993526%	99.9992261%	99.9990365%	99.9987202%
SimPy	99.9992911%	99.9991491%	99.9989364%	99.9985821%
SimPy LB	99.9992909%	99.9991488%	99.9989361%	99.9985817%
SimPy UB	99.9992913%	99.9991493%	99.9989368%	99.9985826%

The 95% confidence intervals of CPN tools and SimPy program are small, proving that the results converge well. There's not much difference in the results from the three solutions. Both CPN tools and the Python SimPy program are validated for modeling a 5G network. The Python program gives the closest result to the analytical result. The difference could come from the Monte Carlo simulation and the analytical solution approximation. However, the CPN tools simulation generates a larger difference to the analytical result. That is because the CPN tools program does not support the load balancing behavior when deploying a new pod, so it randomly chooses a node to deploy the pod. The time that a Node is empty will, therefore, be enlarged, causing a higher value of $\widetilde{MTTF_{node}}$. That explains why the estimated VNF availability is always higher than the one from other solutions. CPN tools may be a good

[1] availability above 99.999%
[2] availability above 99.99%
[3] Lower bounds of the 95% confidence interval
[4] Upper bounds of the 95% confidence interval

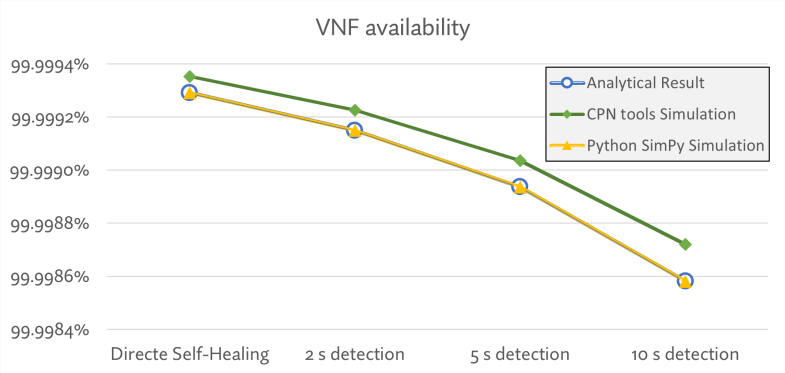


Figure 4.4: Analytical and simulation VNF availability results comparison of the first situation. It presents the availability of each microservice in the VNF under the scenarios of different k numbers.

solution when building the network model. However, its limitation and long simulation time make it not the best choice for a large 5G system modeling and simulation. The analytical solution can work for a simple system, but it becomes hard to solve when it gets more complex. In the thesis work, the Python program is the most promising solution for large 5G system modeling and performance estimation.

In the second situation, the value of k in the k-out-of-3: G microservice system varies. The performance of microservice is compared. The health check interval is set to immediate, i.e., a failure on a pod or node can be detected with no delay. Other parameters are unchanged. The result is obtained from CPN tools.

The results in Figure 4.5 show that if the desired replica quantity k is three (3-out-of-3 microservice), the total VNF availability, as in the previous situation, is 99.9993523%. The separated availability of a single microservice is 99.9996712% (5 nines). If the desired replica quantity k is one (1-out-of-3 microservice with two pods for redundancy), then the availability of this microservice can achieve up to 9 nines. The results directly present the importance of having redundancy in the subsystems.

For more details of the second situation, the reader can refer to Paper II [23].

4.3 . The effect of auto-scaling

4.3.1 . Scenario introduction

This section considers an E2E service that follows an SFC composed of 3 VNFs instead of only looking at one single VNF. The 5G network is assumed to

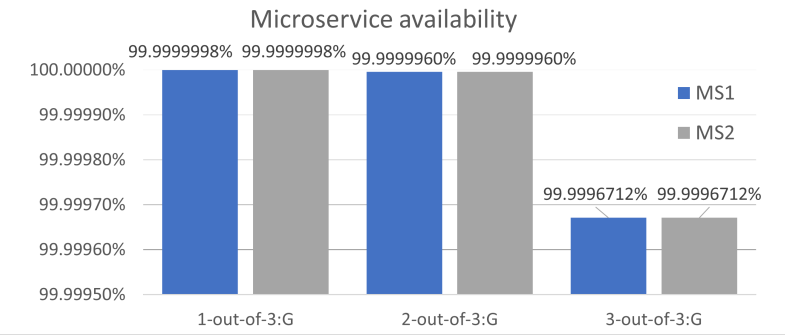


Figure 4.5: CPN tools simulation microservice availability results of the second situation. It presents the availability of each microservice in the VNF under the scenarios of different k numbers. Figure from Paper II [23] and Paper III [24].

be fully virtualized. The SFC containing the three VNFs is given in Figure 4.6. The service chain possesses two functions in the local RAN: DU and Centralized Unit. They are used to provide the connection to the CN. In the virtualized CN, the third function in the chain, the UPF, routes and forwards the packets to the internet. It is assumed that all the end-users in the network have the same SFC, and the packets are only in the uplink direction. Still, the TN is assumed to be fully reliable and capable of transmitting the package without congestion in the TN.

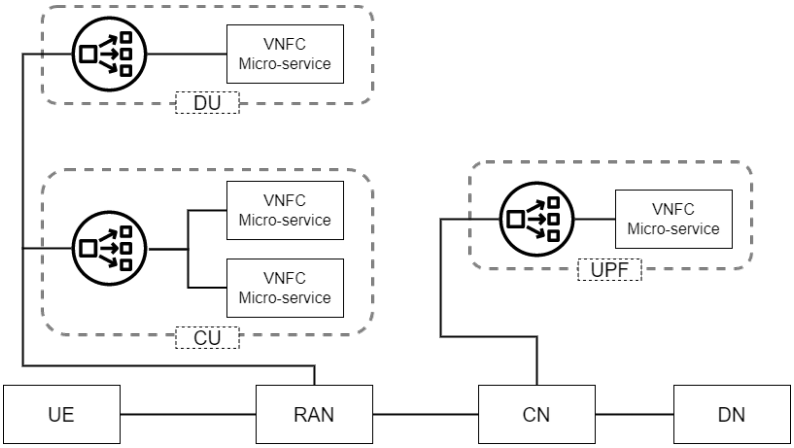


Figure 4.6: Service function chain including 3 VNFs. Figure from Paper V [26].

We look into the short-timescale resilience performance of the network. The failures of the elements are no longer considered threats to the network. The only resilience risk taken into account in this section is the traffic change.

The traffic variation brings many uncertainties to the configuration and makes it hard to prepare the system with an appropriate scale. 5G network is initially well configured for a given traffic forecast. 5G system can be dynam-

ically configured to fit the traffic using 5G NFV MANO when the environment changes. It tries to re-scale itself to save energy when there are few service requests. When the service requests grow, it increases its capacity.

A long timescale mobile traffic forecast can almost precisely anticipate the traffic change during a week or a day, as found in [206, 207]. However, in a short period, adverse events such as DDoS attacks, flash mobs, and some impromptu events could induce abnormal traffic that is hard to predict. A real example of network behavior during a football match is reviewed in [208]. A one-minute disruption would be tolerable for a smartphone user during an adverse event. However, it could be catastrophic for a reliable-sensitive use case and lead to severe consequences. For example, real-time applications, such as remote surgery, factory automation, and intelligent transportation, require reliable and precise information and feedback [209].

For the three-VNF SFC, when the end-users increase their traffic, a VNF may be congested if the number of packets arrived exceeds its capacity to treat them. Some packets will be delivered with a long delay, and some may be rejected due to the limited storage. If these pieces of important information are not completely delivered, the service loses its performance, becomes unavailable, and eventually causes serious accidents. Although short-term performance loss becomes critical in network resilience, few works have focused on a short-timescale traffic variation.

5G uses auto-scaling to realize an automatic scalability change according to a predefined strategy. In the modeling part, the Kubernetes HPA will be implemented in a 5G network management. It is modeled as in Subsection 3.2.2 to provide a scaling function.

This section presents two studies. Firstly, the scenario of a sudden increase in traffic from end-users is analyzed. Only service from one type of end-user is considered. The effect of different auto-scaling setups is discussed. Secondly, different kinds of traffic variations are injected into the model. Two types of end-users are considered. Different scaling strategies are compared.

4.3.2 . Network resilience performance under sudden traffic increase with auto-scaling

4.3.2.1 Traffic variation

In the first situation, the injected end-user traffic has a sudden increase pattern as depicted in Figure 4.7. The network has been initially well-scaled to meet the traffic rate of 1000 request packets per second. The packet arrival time (time difference between two packets arrival) follows an exponential distribution with parameter λ equals the inverse of the traffic rate. The request traffic arrival rate linearly increases from the time 10 s until 35 s, from 1200 to 4200 requests per second. Then, the traffic goes back to its normal state.

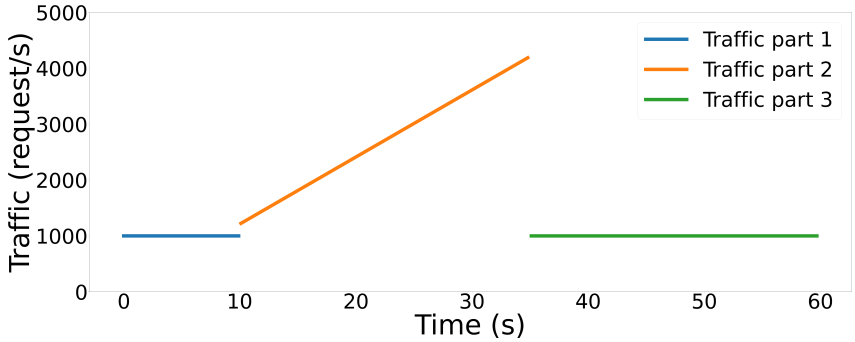


Figure 4.7: Traffic increase scenario. Traffic increased from 10 s to 35 s.

4.3.2.2 Resilience performance metrics

In order to evaluate the resilience performance, several resilience metrics are applied.

Latency is one of the critical indicators for network service. Especially in traffic variation, the congestion may vastly increase the latency. A long latency is often undesirable because it violates the Service-level agreement (SLA). The delayed packet becomes useless for the vertical service since it can no longer provide timely and useful information. In the proposed 5G system model, for a packet i, its packet delay d_i is composed of the transmission time in RAN $d_{trans,i}$, the sum of processing time at each VNF (a set of microservices) $d_{proc,i}$, and the sum of waiting time in the queue of each microservice $d_{wait,i}$ as in Equation (4.6). Other types of latency, such as time spent on a switch, are not considered.

$$d_i = d_{trans,i} + d_{proc,i} + d_{wait,i} \tag{4.6}$$

When we investigate the latency evolution for a couple of seconds, it seems impractical to examine the E2E latency, packet by packet. Indeed, it is preferable to look at the average delay of the service during a short time slot. The service latency is then discretized and is based on the average latency of the packets delivered in a time interval (0.1 seconds, for example, in the simulation of the thesis work). Equation (4.7) illustrates a way to calculate the service delay of one single time slot $]t, t+\Delta T]$ where it uses the average latency of all N delivered packets out of M transmitted packets during this time interval. d_i is the E2E delay of the $i-th$ packet. x_i is a binary variable, and it takes the value of 1 when the $i-th$ packet has arrived at its destination, and it takes the value of 0 when the target does not receive it.

$$\text{Service delay}(t) = \frac{\sum_{i=1}^{M} d_i \cdot x_i}{N}, \text{ where } N = \sum_{i=1}^{M} x_i \tag{4.7}$$

In the normal state, the packet average delay is around 0.035 s, including 34 ms processing delay, 1 ms transmission delay, and negligible waiting delay. However, the load on pods grows with the traffic, and they are soon congested. When a packet demands a microservice, there are no more available pods to serve it. The waiting delay increases, and when the waiting list is complete, the coming packets will be rejected.

Packet loss or packet acceptance rate is another indicator of the 5G networks' resilience. Based on assumptions, when a microservice queue is full, the arriving packets may not join the queue and then be rejected. The packet losses in the TN and the radio transmission are not taken into consideration. Sometimes, a network service can be very sensitive to packet loss since it impacts the quality of receiving data.

The packet loss is the number of rejected packets divided by the total sent packets. The packet acceptance rate is the number of packets N that arrive at its SFC destination divided by the total sent packets M. The sum of these two indicators is 100%. For ease of estimation, they can also be discretized over time intervals of 0.1 seconds. Equation (4.8) and Equation (4.9) show how packet acceptance (PA) and packet loss (PL) in the time slot $]t, t + \Delta T]$ are calculated.

$$\mathsf{PA}(t) = \frac{N}{M} \cdot 100\%. \tag{4.8}$$

$$\mathsf{PL}(t) = (1 - \frac{N}{M}) \cdot 100\%. \tag{4.9}$$

In a normal operation mode, the packet acceptance rate should be 100%, and the packet loss rate should be 0%. However, these indicators will not stay at a stable interval during some incidents. For example, in the case of traffic variation, congestion may occur at some microservices. As a result, packets may need to queue up for an available microservice pod and even be rejected if the queue is full. Then, the latency will increase, and the acceptance rate may decrease. Those packet losses can be fatal for vertical usages, such as the automatic control system, where continuous signals are indispensable.

4.3.2.3 Simulation setup

The network is managed by a threshold-based Kubernetes HPA, with a working algorithm as described in Algorithm 1. If the utilization rate of a microservice is outside the threshold interval, a new scale of the microservice will be calculated as follows:

$$\text{New scale} = \lceil \frac{\text{Current utilization}}{\text{Desired utilization}} \cdot \text{Current scale} \rceil \tag{4.10}$$

Transition time and processing time on VNF microservices follow an exponential distribution. The waiting list length for each microservice is 100 packets. When the queue length reaches 100 packets, a new packet will be rejected automatically. Other network parameters are given in Table 4.4.

Table 4.4: Single service function chain network parameters. Table based on Paper IV [25].

Parameter	**Value**
Number of VNF microservices	DU:1 MS, CU:2 MS, UPF:1 MS
Initial container/pod instances	3 pods for each MS
Transmission time	1 ms between RAN and CN
MS processing time for DU and CU	8 ms for each MS
MS processing time for UPF	10 ms for each MS
MS resource allocation for DU and CU	6 CPU units for each MS
MS resource allocation for UPF	12 CPU units for each MS
Packet processing resource	1 CPU unit for each MS
Queue length for process	100 packets for each MS
Number of nodes in RAN	4 nodesN
Number of nodes in CN	8 ndoes
Node capacity for RAN	18 CPU units per node
Node capacity for CN	36 CPU units per node
Desired CPU utilization rate	50%
Auto-scaling threshold	$\pm$ 30%
Pod starting time	50 ms
Pod termination time	30 s
Simulation run	1000 iterations

The simulation is carried out by a Python program based on the SimPy simulation framework. The auto-scaling module is activated. The effectiveness of auto-scaling is examined by changing different sync periods.

4.3.2.4 Simulation results

The service delay result is given in Figure 4.8. If there is no auto-scaling, the waiting delay increases up to 80 ms, and the overall delay will not decrease unless the traffic returns to normal. When we adopt a 15-second auto-scaling sync period, we find that few pods are scaled out at time 15 s, and more pods are scaled out at 30 s. These two scaling operations are not enough to immediately handle the congestion. In the 10-second sync period situation, the scaling-out decisions are taken at 20 and 30 s. The network service delay is shorter than the 15-second sync period case after 30 s. Finally, in the 5-second

sync period auto-scaling case, scaling decisions are taken more frequently, and the congestion time and service delay are significantly reduced.

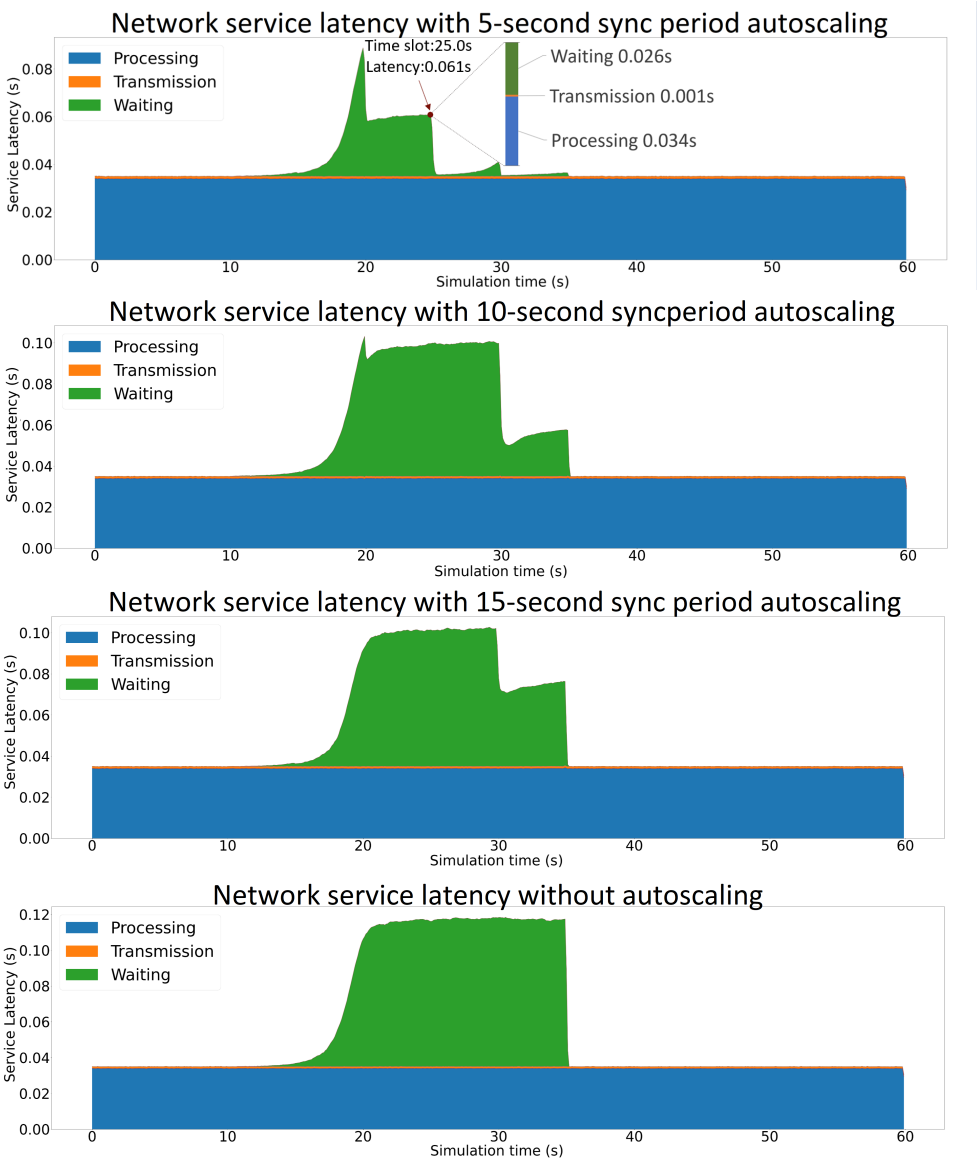


Figure 4.8: Network service latency with and without auto-scaling. Blue for processing delay, yellow for transmission delay, and green for waiting delay. Figure from Paper V [25].

Figure 4.9 gives the service acceptance rate result. Without auto-scaling, the acceptance rate may reduce up to almost 50%. With 5-second auto-scaling, both duration and packet rejection are largely reduced. The resilience is improved by shortening the time to adapt to the reverse event and better maintaining the performance. While for 10-second or 15-second auto-scaling, the

disturbance interval is not significantly reduced, the maximum packet acceptance degradation is about 40%. The acceptance rate is improved only after 30 s. However, the system is not fully recovered. It keeps suffering from the disturbance since the auto-scaling at 30 s is insufficient to cope with the continuously growing traffic.

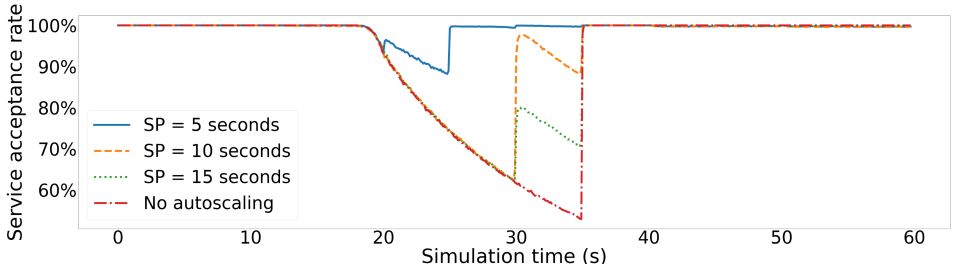


Figure 4.9: Network service acceptance rate with 15, 10, 5 seconds sync period auto-scaling and no auto-scaling. Figure from Paper V [25].

By comparing the acceptance rate performance, the 5-second performs the best in terms of service latency, system suffering time, performance degradation, and restoration time. However, frequently adjusting the scale of the 5G network may not be a wise choice. When doing scaling-in, it takes some time to terminate pods gracefully. The pod resources will not be released immediately. During this time, some of the resources become unavailable, and the system may not be able to scale out when the traffic immediately increases due to a lack of resources. Therefore, some more complicated algorithms can be further applied to set up scaling rules to adjust the system to the traffic load better.

4.3.3 . Network resilience performance under changing traffic conditions with different scalability strategies

The result from Subsection 4.3.2 shows that the proposed model is capable of estimating the network latency and acceptance rate under the presence of an adverse event. The usage of auto-scaling can mitigate the congestion of a sudden traffic increase.

A more complex case is considered in this second part of the scenario.

4.3.3.1 Duo service traffic variation

In the complex situation, the network still consists of one RAN and one CN. The 5G network we consider is fully virtualized. This network hosts two network services as shown in Table 4.5. Service 1 is a latency-sensitive type application with small-size packets. A slight congestion can cause a severe latency requirement violation. Service 2 is an IoT-type application. Its latency require-

Table 4.5: Network services characteristics

	Value	Remarks
Packet type		
Service 1	short packet	ping packet
Service 1	long packet	data message
Packets the mean inter-arrival time		
Service 1	high frequency	exponential distribution
Service 2	low frequency	exponential distribution
Latency requirement		
Service 1	10 ms	low latency, high priority
Service 2	50 ms	low priority

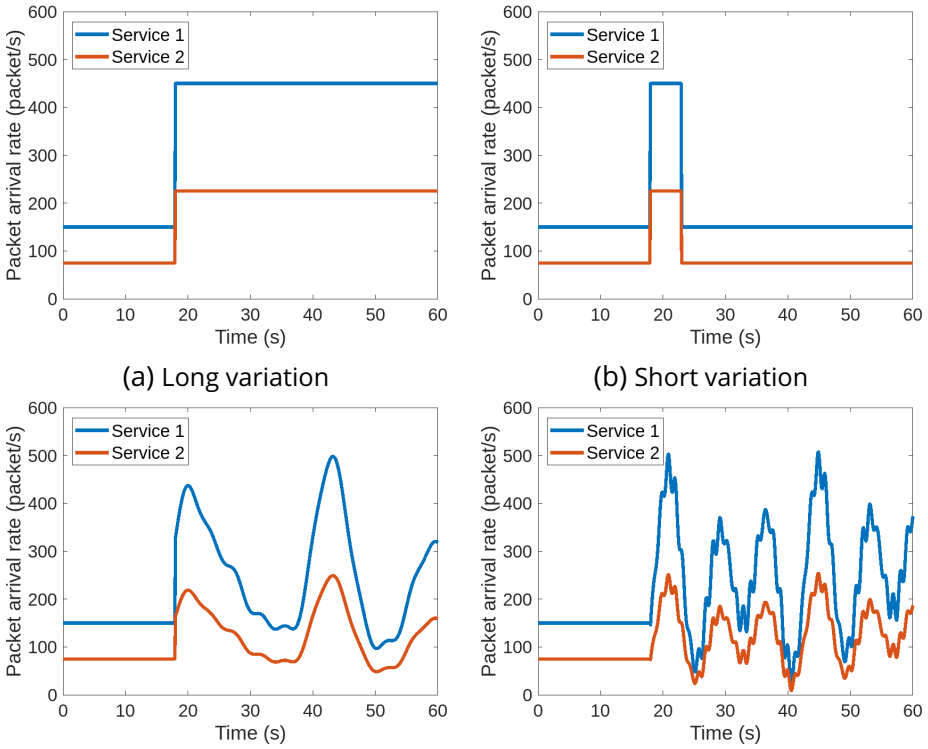
ment is relatively less strict. Both of these two services are considered uplink user-plane applications.

Four different traffic variations are injected as threats in the model: a short traffic change, a long-term traffic variation, and two fluctuating traffic changes. The traffic arrival follows an exponential distribution, and service 1 always has twice the traffic arrival rate as service 2, as shown in Figure 4.10. The irregularity of these traffic patterns, which is quantified by approximate entropy [210], increases one by one.

4.3.3.2 Auto-scaling strategies

The auto-scaling setup is given in Table 4.6. The pod graceful termination time is set to 15 s in order not to freeze the resource for a long time, allowing more frequent scaling actions. We compare different strategies: no auto-scaling (No AS), threshold-based basic Kubernetes built-in auto-scaling (Basic AS), and threshold-based basic auto-scaling combined with stabilization window (Win. AS) under four different traffic variations.

In the No AS strategy, no auto-scaling is performed. The 5G system will maintain the same scale during the traffic variation. In the Basic AS strategy, the Kubernetes HPA sends a probe to detect the CPU utilization rate of each microservice with a sync period of 5 seconds in this case. The scaling strategy is, by default, the one in Equation (4.10). If the new scale is greater than the current scale, a scaling-out decision is made to create more microservice instances. Otherwise, a scaling-in decision is made to remove some existing instances. In the Win. AS strategy, the HPA does not directly trigger a scaling action every 5 seconds. Instead, the decision is based on the resource utilization information during the stabilization window. In this case study, the window is 15 seconds. Therefore, a scaling-out decision is adopted if there are



(a) Long variation

(b) Short variation

(c) Sinusoidal variation 1

(d) Sinusoidal variation 2

Figure 4.10: Four traffic patterns with different arrival rate variation after t = 18 s. (a) Long-term constant variation pattern, approximate entropy: 0.0108. (b) Short-term constant variation pattern, approximate entropy: 0.0207. (c) Sinusoidal (superposition) variation pattern 1, approximate entropy: 0.1019. (d) Sinusoidal (superposition) variation pattern 2, approximate entropy: 0.3676. Figures from Paper VI [26].

Table 4.6: Network management parameters in traffic variation case. Table based on Paper V [26].

Parameter	**Value**	**Remarks**
Pod creation time	50 ms	exponential distribution
Pod termination time	15 s	fixed value
Auto-scaling interval	5 s	fixed value
Auto-scaling goal	50%	CPU utilization rate
Auto-scaling thresholds	30%&70%	down and up thresholds
Stabilization window	15 s	if applicable

three successive scaling-out proposals during the last 15 seconds, and it scales out to the smallest proposed scale. A scaling-in decision is triggered only after three successive scaling-in proposals and chooses the biggest estimated

scale.

4.3.3.3 Resilience performance metrics

New metrics are introduced for resilience evaluation.

Reliability in the context of network layer packet transmissions is the percentage value of the packets successfully delivered to a given system entity within the time constraint required by the targeted service out of all the packets transmitted [4]. It is a combined perspective of E2E latency and packet loss rate. Packet transmission reliability in one time slot is the percentage of the requests that are not rejected and whose delay is below the latency requirement. Equations (4.11) and (4.12) give the calculation of service packet transmission reliability (in this section, it is also called service reliability) SR. x_i and di are the same as in the 4.3.2, the binary value representing if packet arrives its destination and the latency of the packet, respectively.

$$\text{SR}(t) = (\frac{\sum_{i=1}^{M} x_i \cdot y_i}{M}) \cdot 100\%. \tag{4.11}$$

$$y_i = \begin{cases} 0, & \text{if } x_i = 0 \text{ or } d_i > \text{latency requirement} \\ 1, & \text{otherwise} \end{cases} \tag{4.12}$$

The resilience triangle [68] can be used to quantify the resilience concept. As the service packet transmission reliability considers both the acceptance and service latency, we adopt this metric as the performance function. Then, the **resilience loss** can be quantified by calculating the area of the degradation in the service reliability over time. The service packet transmission reliability is discretized based on a time slot $]t_k, t_k+\Delta T]$ in the proposed simulation model as shown in Figure 4.11.

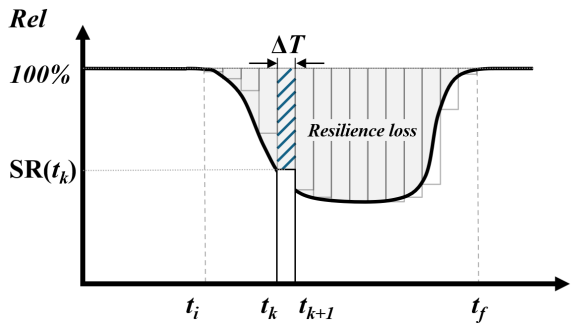


Figure 4.11: The resilience triangle. The incident takes place at t_i. The system recovers at t_f. The gray part represents the resilience loss of the k-th time slot. Figure from Paper V [26].

The estimated resilience loss of the network service under a certain incident is given as:

$$\mathrm{RL} = \int_{t_i}^{t_f} [1 - Rel(t)]dt = \sum_{t=t_1}^{t_K} [100\% - \mathrm{SR}(t)]\Delta T \tag{4.13}$$

In Equation (4.13), t_i is the time when the incident starts, and t_f is the time when the service is completely recovered. If we discretize the impacted duration into K time slots of length ΔT (the same slots as we calculate the performance metrics), the continuous integral of resilience loss equals the sum of $[100\% - \mathrm{SR}(t_k)]\Delta T$.

In addition to the service performance, network resource allocation is also a critical concern. Over-allocating CPU resources to network services improves resilience performance in the presence of adverse events. Nevertheless, the over-booked resources will not only charge an extra fee but also consume more energy. As shown in Table 4.7, it takes 20 CPU units of resources to run a pod of DU or CU microservice and 40 for a pod of UPF microservice. When Kubernetes takes charge of auto-scaling, it can adjust the number of pod instances according to the traffic congestion situation, thus resulting in changing the resource allocation. To quantify resource cost, the resource usage metric is introduced. We define in Equation (4.14) resource cost RC as the sum of the resource cost of each pod j in the 5G system, measured in CPU unit · second. For each pod, its resource utilization is the product of CPU resources that have been allocated to the pod and the pod lifetime ($t_{ej} - t_{0j}$). An ideal 5G system should have highly resilient performance while using fewer resources.

$$\mathrm{RC} = \sum_{j \in P} \mathrm{RC}_j = \sum_{j \in P} \mathrm{cpu}_j (t_{ej} - t_{0j}) \tag{4.14}$$

4.3.3.4 Simulation setup

The VNFs remain unchanged as in the first situation. The pods' capacity is modified to fit the considered problem. The network settings are given in Table 4.7. All parameters, including components of VNF, and their capacities eventually depend on the actual services suppliers provide.

The service packet in the 5G network generated by the user will be processed locally by the RAN microservices (in order), then transmitted to CN, processed again, and finally delivered to the internet. We adopt a higher RAN functional split [211]. Then, CU gathers more functions than DU, so it comprises more microservices. Since UPF is in the aggregated CN, each UPF pod allocates more CPU units to treat more packets in parallel. The processing time and transmission time are given in Table 4.8. The packet processing

time is proportional to the packet size, as we assume that one packet can be treated by one CPU unit only. With more resources allocated to VNFs in CN, UPF is capable of treating twice the packet than the VNFs in RAN, but all microservices process packets at the same rate. The variant part of packet delay is the service delay in the microservice queue. When a pod microservice is overloaded (congested), the arrival packets will queue up and wait for available resources. When the queue reaches the maximum length, the arriving packet will be rejected. The parameters of processing time and transmission time, in reality, may be associated with uncertainty as well. Since the major interest of this study is to estimate the network service resilience to congestion effects due to traffic variation, and the uncertainty of processing time is assumed to stay unchanged during adverse events, these parameters are considered fixed values.

Table 4.7: Service function chain composition. Table from Paper V [26].

	Number of instances	**Capacity**
VNFs in RAN		
DU	1 MS	infinite number of pods
MS of DU	initially 1 pod	20 CPU units per pod
CU	2 MS	infinite number of pods
MS of CU	initially 1 pod	20 CPU units per pod
VNF in CN		
UPF	1 MS	infinite number of pods
MS of UPF	initially 2 pod	40 CPU units per pod

To achieve an accurate result, the model is programmed in Python with the SimPy platform to run discrete event simulation. We take all iterations' average service latency, service reliability (packet transmission reliability), and service resilience values generated by Monte Carlo Simulation. We limit the time duration to 60 seconds in order to estimate the timely dynamic response of the 5G network. The simulations are run 2000 times to get a confident result.

4.3.3.5 Simulation results

The simulation results of the three strategies under these four different traffic patterns are presented in Figure 4.12, Figure 4.13, and Figure 4.14 and Table 4.9. In the simulation, the network suffers from abnormal traffic from 18 s. Some packets will be rejected during the overloaded situation due to the microservice queue length limit. Although some packets are not rejected, the packets

Table 4.8: Network processes parameters. Table from Paper V [26].

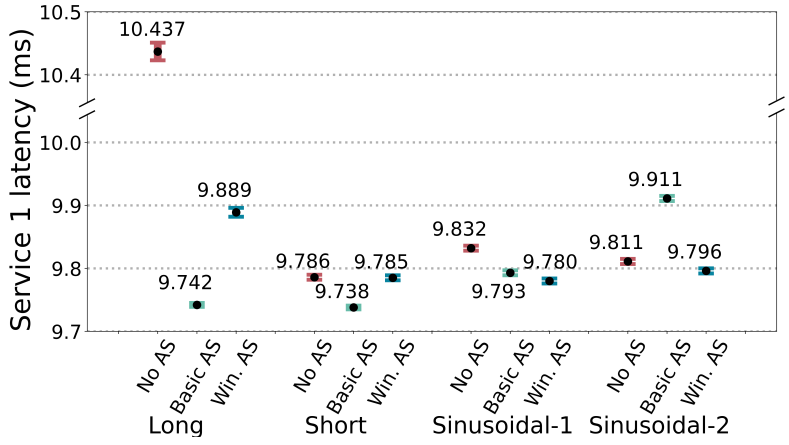
Parameter	Value	Remarks
Processing time		
Distributed Unit MS	short packet: 2 ms long packet: 4 ms	fixed time
Central Unit MSs	short packet: 2 ms long packet: 4 ms	fixed time
UPF MS	short packet: 2 ms long packet: 4 ms	fixed time
Transmission time		
Radio+transport	1.25 ms	fixed time
Service queue		
MS queue length	50 requests	first come first serve priority if applicable
Maximal waiting time	1000 ms	reject if time out

of the latency-sensitive service, service 1, can not afford a long waiting time during the congestion, and its delivery time exceeds the latency limit.

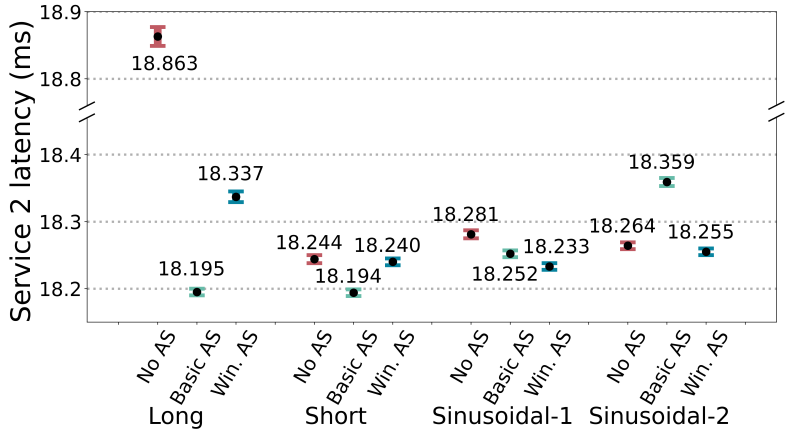
The service latency is estimated in the following way. The ΔT is 0.1 seconds. We collect the packet delay d_i of each packet x_i during this ΔT and compute the corresponding Delay(t) of each interval according to Equation (4.7). Service reliability also evolves with time. We obtain the y_i by verifying if the latency requirement is satisfied for each packet x_i during this ΔT interval and then compute the corresponding service reliability SR(t) of each interval according to Equation (4.11).

In the long traffic change, the Basic AS strategy immediately adds a necessary number of microservice instances to keep the network service load at an acceptable level at 20 s. The window-based strategy takes a relatively long time but eventually relieves the congestion. Not taking any scaling action results in a large resilience loss in the service, especially for service 1, since it is more sensitive to latency. The model captures the service latency and the resilience loss evolution, as presented in Figure 4.15.

For a short-term traffic variation, Win.AS and No AS perform almost the same since the scaling decision is neglected in the former, and no scaling action is required in the latter. This leads to a congestion of the network for about 5 seconds. However, due to the randomness of packet arrival rates, high resource utilization may occur occasionally and trigger window-based auto-scaling, causing a slightly higher resource cost than the No AS scenario.



(a) Service 1 latency

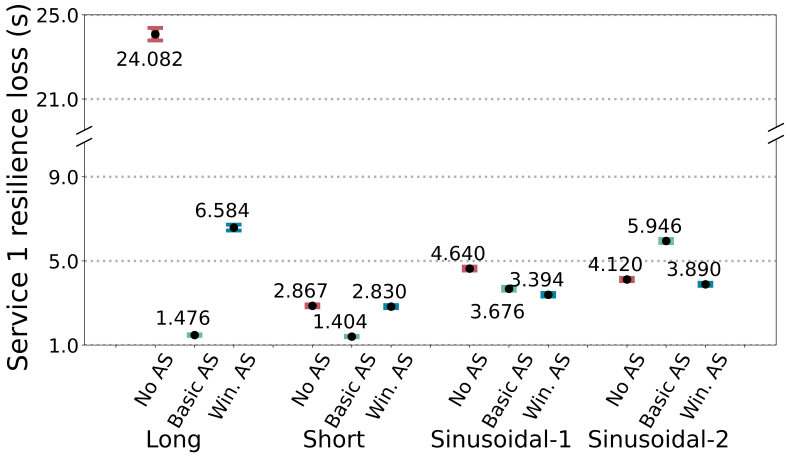


(b) Service 2 latency

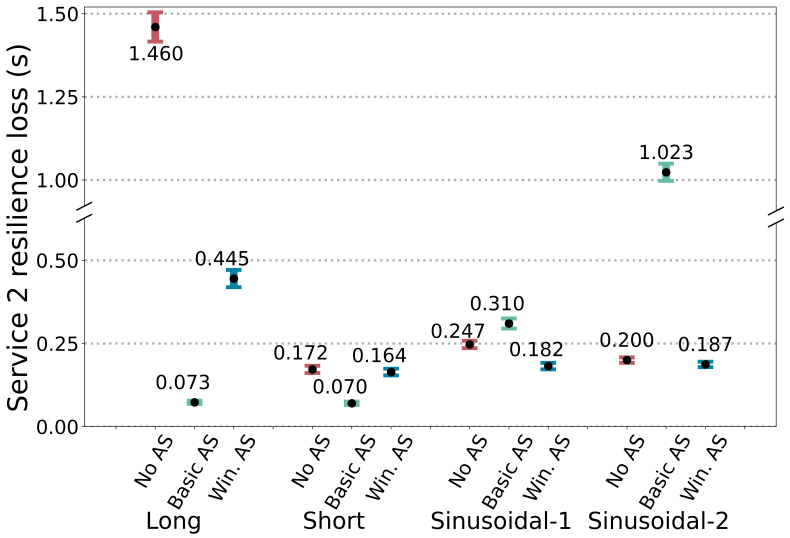
Figure 4.12: Service 1 (a) and Service 2 (b) latency values and confidential intervals in under different traffic. Figure from Paper V [26].

Basic AS reduces congestion time to two seconds. The resilience loss of both services is reduced, but it uses about a quarter more resources than other management strategies. The latency and reliability of the two services are compared in Figure 4.16.

For the less fluctuating sinusoidal superposition traffic variations, the Basic AS strategy makes a decision every 5 seconds to adapt to the traffic. Win.AS considers the traffic change during the last 15 seconds and is thus more "rigorous" to avoid frequent scaling in and out. The three strategies are compared in Figure 4.17. The resilience loss of Basic AS is less at the beginning of traffic variation, but it performs even worse than the No AS mechanism at the end of the simulation (at the third traffic peak). The resilience loss of Win.AS is almost the same as the No AS case initially, but it gradually performs better. The total resilience loss of the Win.AS is less than the Basic AS and the No



(a) Service 1 resilience



(b) Service 2 resilience

Figure 4.13: Service 1 (a) and Service 2 (b) resilience loss values and confidential intervals under different traffic variation. Figure from Paper V [26].

AS. Taking resource cost into consideration, Win.AS is the most economical solution to improve service resilience with a few additional costs.

In a more fluctuating traffic situation, the threshold-based Basic AS algorithm may not provide a satisfying solution. Indeed, the auto-scaling fails to make the correct decision as the expected scale at each decision moment changes. The Win.AS would prefer to decide not to change the scale during the fluctuation. As shown in Figure 4.18, the differences in resource cost and resilience loss for the scenarios Win.AS and No AS are not much. The resilience of Basic AS is worse than No AS, and it costs the most. Basic AS takes the hazard of scaling out and in quickly but fails to provide enough ser-

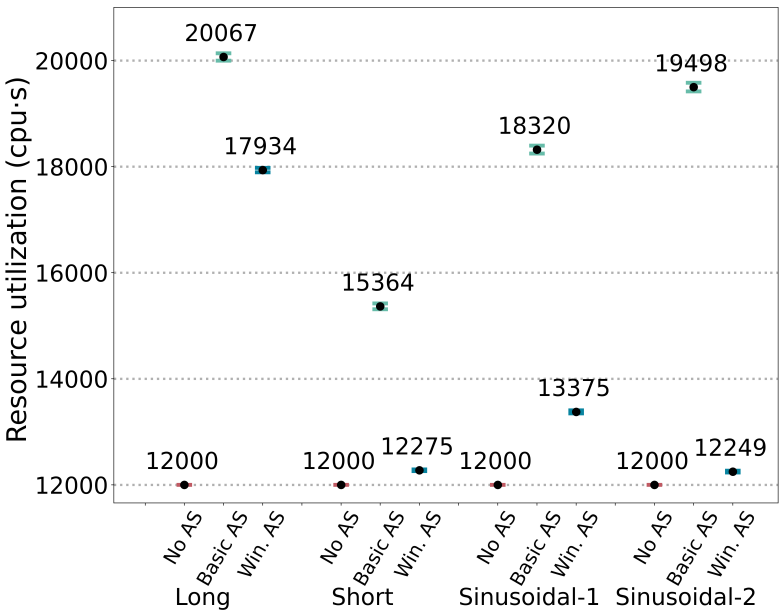
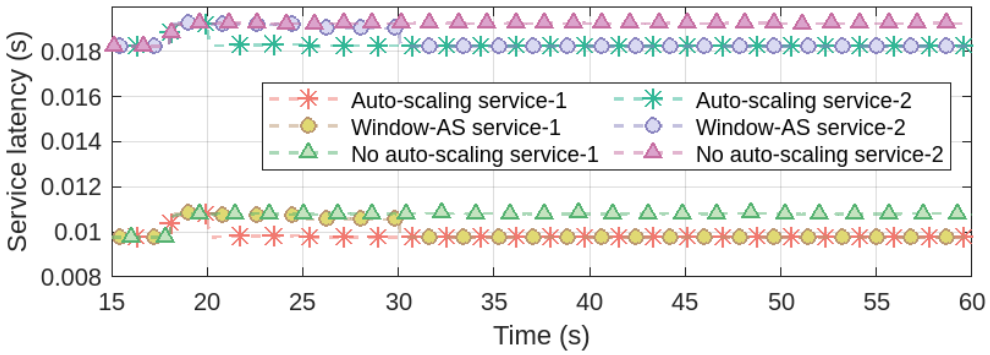


Figure 4.14: Resource cost values and confidential intervals under different traffic variations. Figure from Paper V [26].

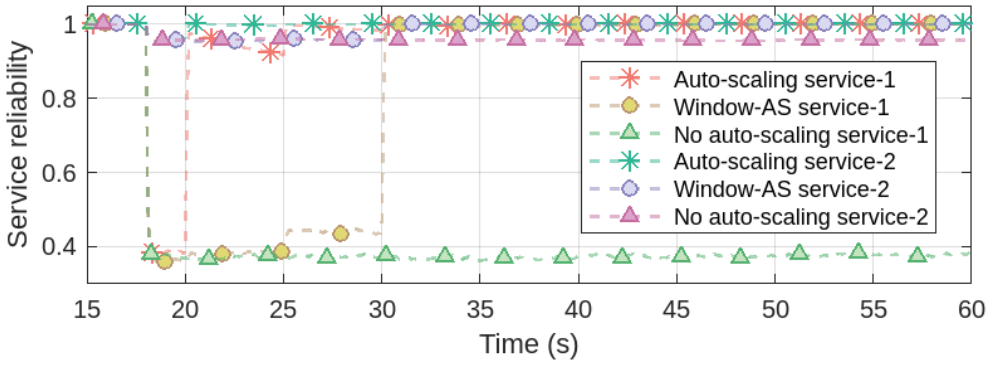
vice instances if there is a traffic increase just after a scaling-in triggered by a short-sighted decision. In fact, a scaling-in action would freeze the removed instance's resource for a while before being entirely killed to ensure all packet treatments are done before removing the instance. This results in a large resource cost and reduces the total available resources in the shared server that other microservices can allocate. In this scenario, Win.AS performs the best in resilience but is close to the No As situation. Basic AS has the lowest resilience and the highest resource cost. If the fluctuation or irregularity of the traffic keeps increasing, it is possible that the Win.AS performs worse than No AS, as it may not always provide a suitable scale.

These strategies seem to perform differently under different traffic environments. Indeed, it is possible to implement artificial intelligence in Kubernetes so that the HPA parameters can be optimized according to the real-time traffic for better service performance. In our model, Kubernetes is assumed to be reliable throughout the simulation. However, in actual network installation, if Kubernetes fails, the HPA function becomes unavailable. In such a scenario, the Basic AS and Win.AS will perform the same as No AS.

Although this study focuses on short-timescale traffic variation, it can be extended to evaluate network service resilience under a long-timescale traffic variation. The long-timescale traffic variation can be seen as slices of short-timescale traffic variation, but the traffic often fluctuates less in each time slot. Therefore, the auto-scaling can better adjust to the traffic, and the network

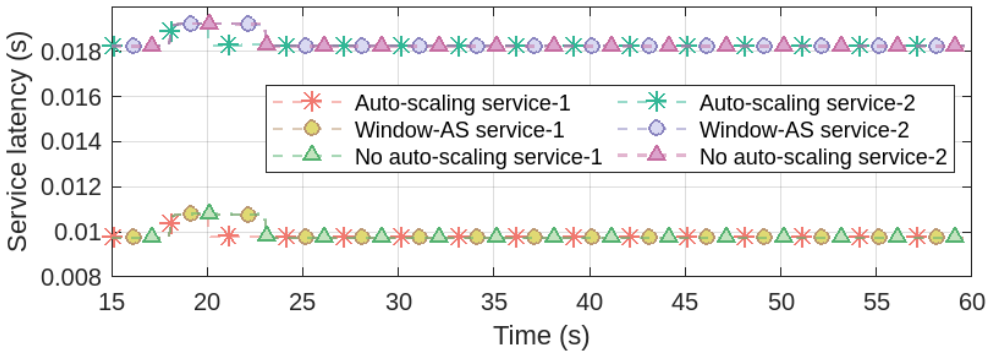


(a) Service latency

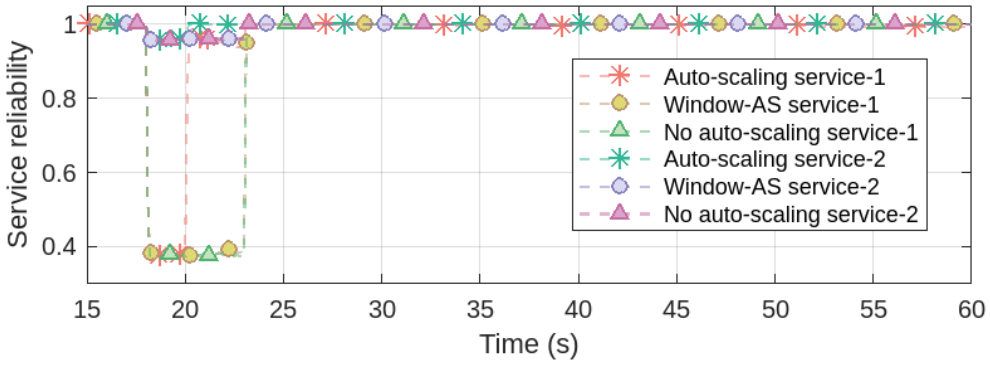


(b) Service reliability

Figure 4.15: Service latency and reliability under a long-term traffic variation (pattern a) with different management strategies (auto-scaling, stabilization window-based auto-scaling, and no auto-scaling). Figure from Paper V [26].

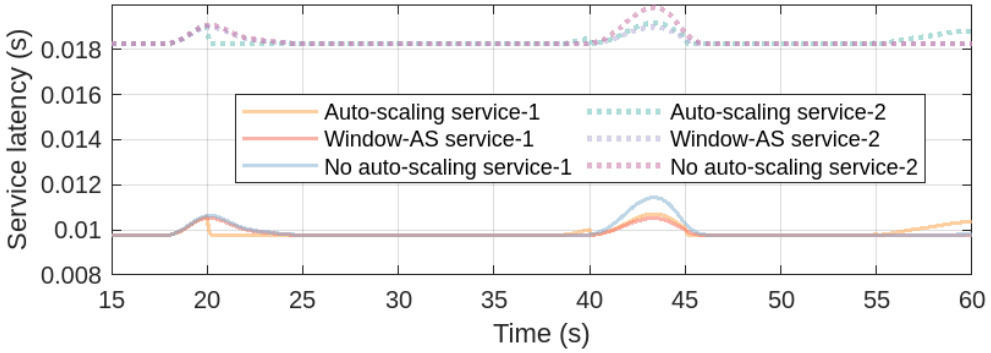


(a) Service latency

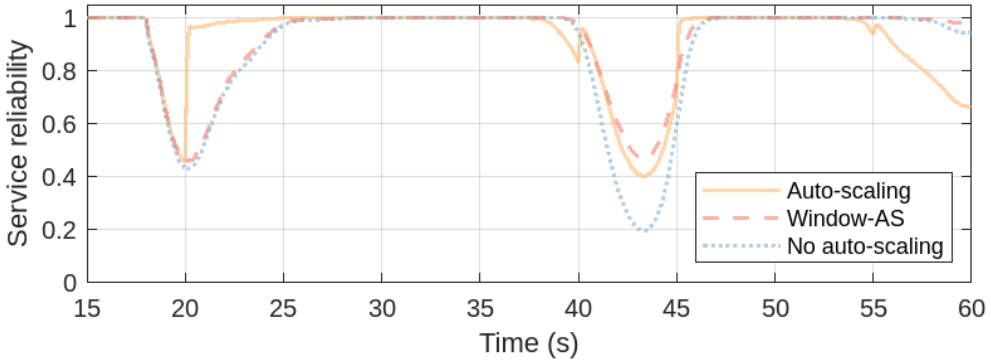


(b) Service reliability

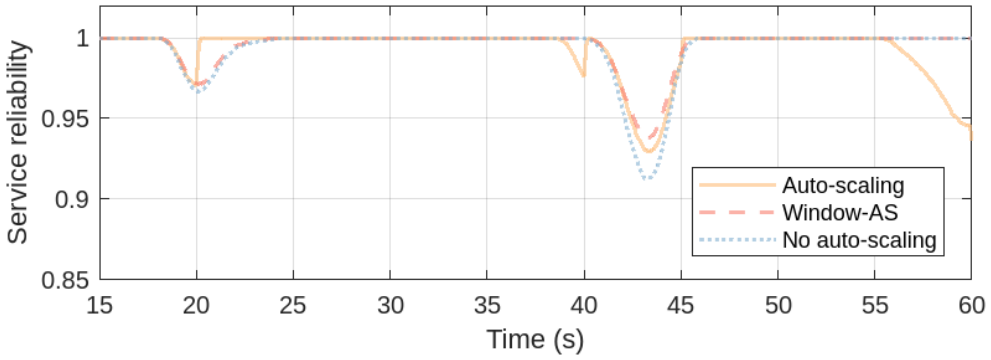
Figure 4.16: Service latency and reliability under a short-term traffic variation (pattern b) with different management strategies (auto-scaling, stabilization window-based auto-scaling, and no auto-scaling). Figure from Paper V [26].



(a) Service latency

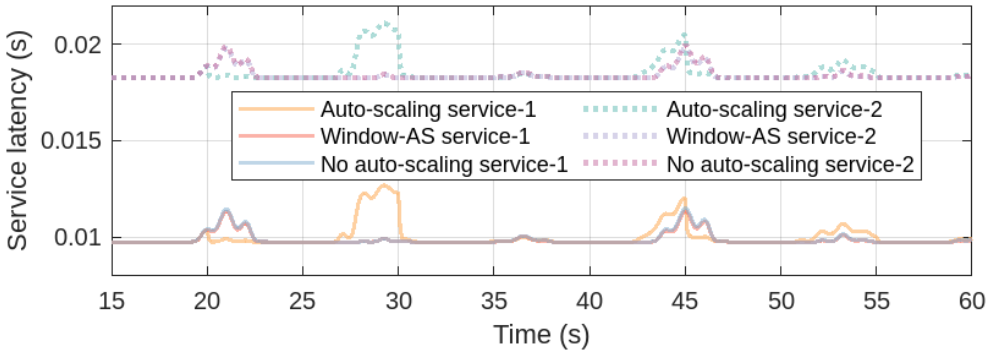


(b) Service 1 reliability

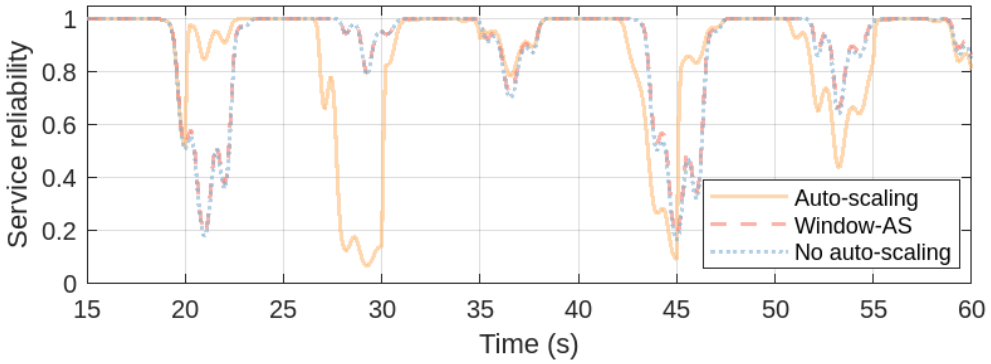


(c) Service 2 reliability

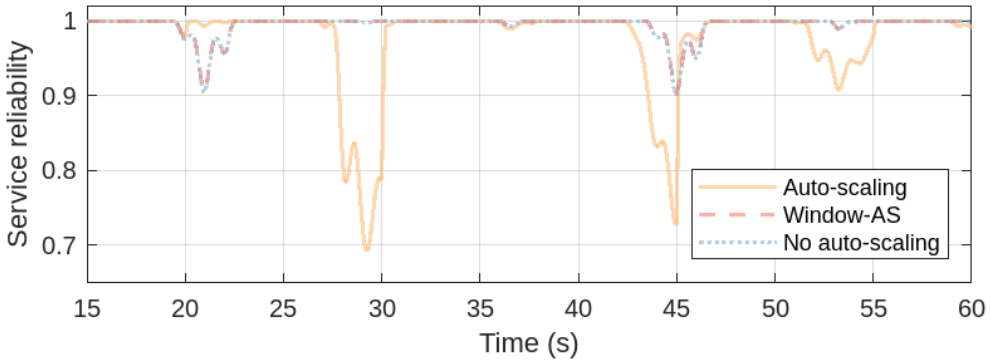
Figure 4.17: Service latency and reliability under sinusoidal superposition traffic variation (pattern c) with different management strategies (auto-scaling, stabilization window-based auto-scaling, and no auto-scaling). Figure from Paper V [26].



(a) Service latency



(b) Service 1 reliability



(c) Service 2 reliability

Figure 4.18: Service latency and reliability under sinusoidal superposition traffic variation (pattern d) with different management strategies (auto-scaling, stabilization window-based auto-scaling, and no auto-scaling). Figure from Paper V [26].

Table 4.9: Simulation result of automatic management under variant traffic.

Strategy		Average latency [ms]	Resilience loss [second]	Resource cost [CPU·s]
Long variation				
No AS	Serv. 1	10.437 (±0.014)	24.082 (±0.300)	12000
	Serv. 2	18.863 (±0.014)	1.460 (±0.044)	
Basic AS	Serv. 1	9.742 (±0.003)	1.476 (±0.037)	20067 (±70)
	Serv. 2	18.195 (±0.005)	0.073 (±0.005)	
Win. AS	Serv. 1	9.889 (±0.007)	6.584 (±0.141)	17934 (±42)
	Serv. 2	18.337 (±0.008)	0.445 (±0.026)	
Short variation				
No AS	Serv. 1	9.786 (±0.004)	2.867 (±0.070)	12000
	Serv. 2	18.244 (±0.006)	0.172 (±0.011)	
Basic AS	Serv. 1	9.738 (±0.003)	1.404 (±0.033)	15364 (±55)
	Serv. 2	18.194 (±0.005)	0.070 (±0.005)	
Win. AS	Serv. 1	9.785 (±0.004)	2.830 (±0.070)	12275 (±16)
	Serv. 2	18.240 (±0.005)	0.164 (±0.010)	
Sinusoidal superposition 1				
No AS	Serv. 1	9.832 (±0.004)	4.640 (±0.092)	12000
	Serv. 2	18.281 (±0.006)	0.247 (±0.011)	
Basic AS	Serv. 1	9.793 (±0.004)	3.676 (±0.092)	18320 (±74)
	Serv. 2	18.252 (±0.005)	0.310 (±0.015)	
Win. AS	Serv. 1	9.780 (±0.004)	3.394 (±0.088)	13375 (±28)
	Serv. 2	18.233 (±0.005)	0.182 (±0.010)	
Sinusoidal superposition 2				
No AS	Serv. 1	9.811 (±0.004)	4.120 (±0.077)	12000
	Serv. 2	18.264 (±0.005)	0.200 (±0.008)	
Basic AS	Serv. 1	9.911 (±0.004)	5.946 (±0.093)	19498 (±81)
	Serv. 2	18.359 (±0.006)	1.023 (±0.026)	
Win. AS	Serv. 1	9.796 (±0.037)	3.890 (±0.074)	12249 (±18)
	Serv. 2	18.255 (±0.005)	0.187 (±0.008)	

Note: 95% confidence interval is given after the metric value.

service is thus more resilient to a long timescale traffic variation.

4.4 . The effect of service isolation and prioritization

4.4.1 . Congestion propagation

Network congestion often occurs locally due to increasing traffic demand or insufficient network resources. However, the congestion can also lead to a Domino effect, causing a propagation of the undesired congestion in the following parts of the network.

4.4.1.1 Network layout

The considered network is similar to the case in Subsection 4.3.2. Four local RAN networks cover the whole area. The DU and CU collocate in the local RAN, and UPF is located in a centralized CN as depicted in Figure 4.19. Only one type of service is considered in this case. The end-users are equally distributed in the four local RAN zones. Different DU and CU instances are assigned to the UE in different zones according to geographical locations. Therefore, these UE's packets are isolated in DU and CU but not for UPF. A traffic variation in one zone will first congest DU and CU. Then, it can probably propagate to UPF, which is initially set up with more redundancy than local VNFs. The packets from other zones with no local congestion will be delayed due to the congestion happening in shared UPF. In this example, the auto-scaling sync period is set to 10 seconds. Other parameters are the same as in Subsection 4.3.2. The abnormal traffic in zone 1 is the same pattern as in Subsection 4.3.2. The latency and acceptance rate for packets starting from different zones are presented in Figure 4.20 and Figure 4.21.

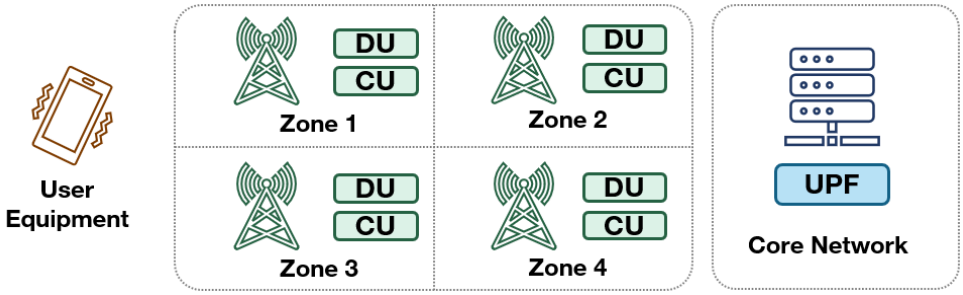


Figure 4.19: Network layout with four local RANs. Figure from Paper IV [25].

4.4.1.2 Simulation result

The traffic change from zone 1 congests not only the local VNFs DU and CU but also propagates to UPF.

Since the UPF is initially scaled for four radio network zones, it has a bigger capacity than DU and CU. The traffic congestion on UPF is less severe than in the case in Subsection 4.3.2. The packet waiting delay in zone 1 increases to 70 ms. The packet waiting delay in other zones is about 15 ms, majorly caused by

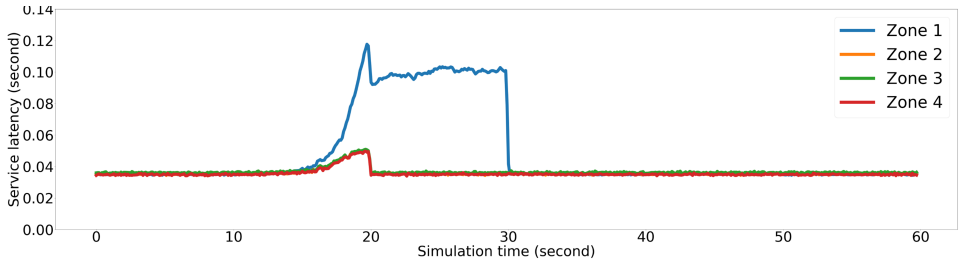


Figure 4.20: Network service latency of users from different zones. Figure from Paper IV [25].

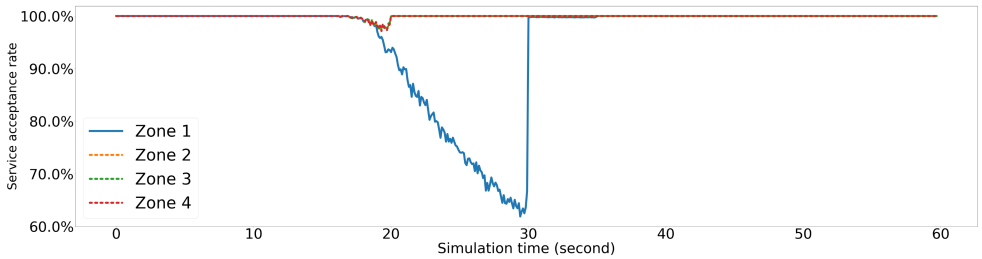


Figure 4.21: Network service acceptance rate of users from different zones. Figure from Paper IV [25].

UPF congestion. At 20 s, a scaling-out decision is taken by Kubernetes. It creates the maximum number of pod replicas that the scaling algorithm allows. For the UPF microservice, one single scaling-out action is enough. The congestion is then released, and the packets in zones 2-4 are no longer queuing for UPF. However, in zone 1, local DU and CU microservices are still congested after one auto-scaling action.

The result shows that a traffic change can cause congestion on VNF, which can propagate from RAN to CN. Adopting a local RAN isolation, only zone 1 is largely impacted by the traffic change. The network services of other zones are less impacted, with less than 3% packet loss, proving the effectiveness of network isolation in improving network service resilience.

Although we are limited in physical or geographical network isolation in this case, this result can still be meaningful since it may be extended to a virtual isolation case for 5G QoS or network slicing.

This case only has one type of service so that the network treats the packets from all users equally. Two services are considered in the following Subsection 4.4.2. Different service-based network management methods are compared.

4.4.2 . High resilience performance with network prioritization

and isolation

4.4.2.1 Prioritization and isolation

Priority can also improve network resilience. When an adverse event occurs, the service with higher priority will be able to allocate more resources. The prioritized service will be processed first in case of congestion. Prioritizing one critical vertical service can largely reduce service latency and packet loss in case of congestion but may also penalize a lot on the less prioritized services.

Without full isolation, all network services will share the network resources. Higher-priority services may even take resources from lower-priority services. By introducing new notions, such as network slicing, network resources are sliced and assigned to different usages so that different services use the customized VNFs belonging to their slice. When the end-user starts a communication, the PDU session establishment is informed of which VNF instances are used when delivering data packets. With the help of network slicing, the network management becomes more efficient. The VNF instances are adapted to different usages, which creates a virtual separation among the verticals. When end-user traffic demand changes, only the corresponding VNF instance will be re-scaled and the rest of the network stays unchanged.

To verify the model's support for network prioritization and isolation and to compare the resilience performance, we consider a no-auto-scaling 5G system composed of four identical distributed local RANs (for zones 1-4) and a centralized CN.

The network setup is similar to Subsection 4.3.3. Two kinds of services are considered. In zone 1, only service 1 end-users are connected and always generate regular traffic. In zones 2, 3, and 4, only service 2 end-users are connected, and they start to change the traffic arrival rate by triple (short traffic variation for 10 seconds). If no network slice is applied, in RAN, each service has its own VNF since they use different physical infrastructure geographically. They share the same UPF instance in the centralized CN. If priority is applied, the latency-sensitive service 1 packets are prioritized in the shared VNF. If slicing is applied, then in CN, each service has its UPF instance, and they are managed separately. These UPF instances are assigned to end-users when building PDU sessions for the connection between the user and the network.

Four scenarios are compared: no slicing or priority network scenario, prioritization network scenario, and two sliced network scenarios. We consider two slicing partitions. The first partition is to create two separate UPF instances for services 1 and 2, each using the same amount of resources as in the shared UPF. Therefore, we double the initial resource. The second partition is to create two different-sized UPF instances with different resource allocations according to the initial service traffic. The total resource usage of the two UPFs equals the single shared UPF.

4.4.2.2 Simulation result

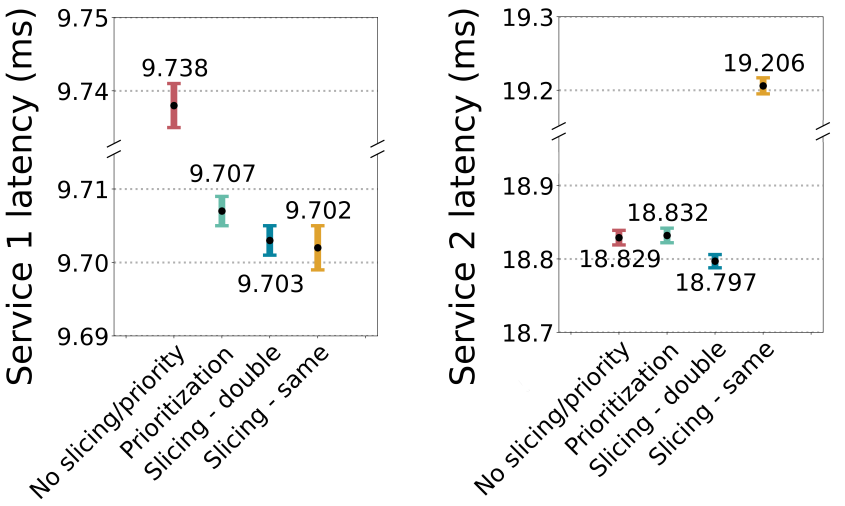
Figure 4.22 and Table 4.10 show the simulation results. Prioritization helps vastly reduce critical service resilience loss without the need for allocating more resources as it treats the latency-sensitive packets first so that most of them do not exceed the time limit. On the other side, the less-prioritized will suffer some resilience loss and increase slightly its average latency. In this case, The resilience of the critical service is improved, and the resilience of other services is decreased in exchange.

Table 4.10: Simulation result of the network isolation case.

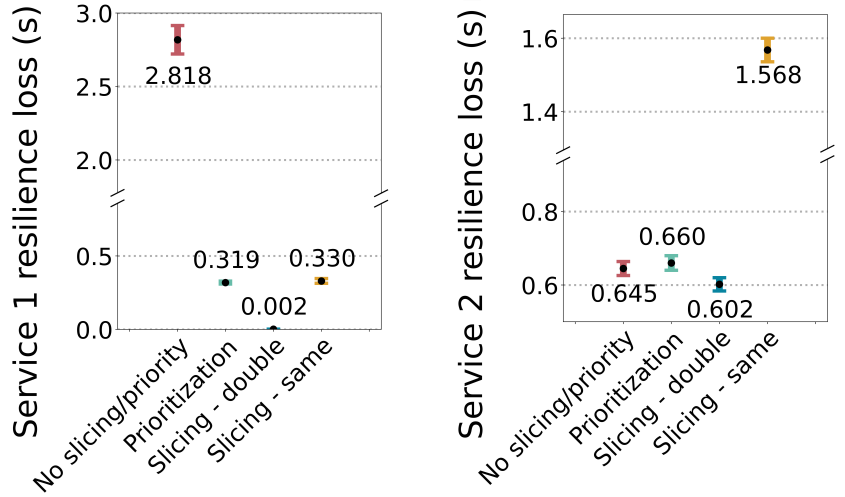
Management prioritization	Average latency [ms]	Resilience loss [second]	Resource cost [CPU·s]
No slicing/priority			
Serv. 1	9.738 (±0.003)	2.818 (±0.097)	19200
Serv. 2	18.829 (±0.010)	0.645 (±0.019)	
Prioritization			
Serv. 1 - prio.	9.707 (±0.002)	0.319 (±0.010)	19200
Serv. 2	18.832 (±0.010)	0.660 (±0.020)	
Slicing - doubled initial resource			
Serv. 1	9.703 (±0.002)	0.002 (±0.0001)	24000
Serv. 2	18.797 (±0.009)	0.602 (±0.018)	
Slicing - same total initial resource			
Serv. 1	9.702 (±0.003)	0.330 (±0.016)	19200
Serv. 2	19.206 (±0.011)	1.568 (±0.032)	

Note: 95% confidence interval is given after the metric value.

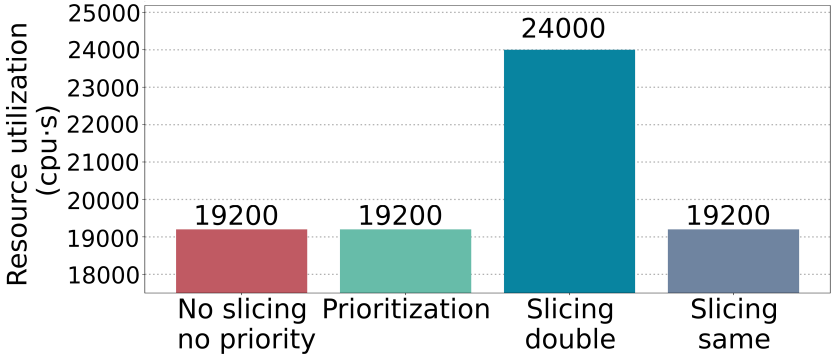
Dedicated slices also keep the latency-sensitive service from anomalies. When failure is injected into service 2 end-users, service 1 is protected by virtual isolation. The two isolation solutions work differently. If each service has its own dedicated UPF instance the same size as the shared one, then the performance of both services is better than without slicing. However, it takes relatively more overall resources (about a quarter in this case). If we keep the overall resource the same, each service only has a limited dedicated resource. For each service, the allocated resource in normal operation mode is less than in a shared network. Although not disturbed by other services, service 1 has more chance to overload the slice by the randomness of the packet arrival due to the reduced resource allocation. This explains a more significant service 1 resilience loss than the doubled initial resource slicing. For service 2, as the resource margin is reduced, it is more likely to get congested than the no-slicing scenario during traffic variation, resulting in a greater resilience loss.



(a) Service 1 and Service 2 latency



(b) Service 1 and Service 2 resilience loss



(c) Resource cost

Figure 4.22: Service latency (a), reliability (b), and resource cost (c) in the network isolation case. Figure from Paper V [26].

According to these results, with a generous budget, the doubled initial resource slicing is preferred during a traffic variation. Otherwise, prioritization is favored as it maximizes the global network service resilience according to the SLA.

4.5 . Conclusion

In this chapter, different 5G network scenarios are introduced. Different resilience threats are injected into the model for different scenarios.

Three major threat aspects are considered. The network internal failure is studied in the first scenario, where long-term simulation generates the average of the system's available time. From the external 5G network, the traffic variation (from users or by attack) is the second aspect considered in the next two scenarios. The simulation results show how the network dynamically adapts to the congestion caused by traffic variation. The latency, reliability, and acceptance rates are computed. Some of these indicators are also used as performance indicators for resilience loss calculation.

In terms of resilience performance improvement, various aspects have been discussed as well. In the operational phase, the NFV MANO plays an important role in resilience performance. The self-healing mechanism reduces network failure time and thus improves availability indicators. The auto-scaling mechanism avoids network congestion. It also reduces network latency and packet loss. Therefore, the network reliability indicator can be improved. In the design phase, different network layout designs, as well as isolation strategies, can impact the overall network reliability and availability.

The analytical methods can be used for a simplified case at a local subsystem. In comparison, the simulation results showcase how the proposed model can estimate a large network's resilience. They also give insights into a first-step network optimization to improve network resilience.

Since network resilience can be estimated through the proposed model, a future extension of the thesis is to investigate how AI or Machine Learning can be introduced to help manage 5GB networks by wisely choosing parameters and dynamically adjusting to the environment to improve resilience.

5 - 5G resilience estimation for vertical applications

Based on the resilience insights acquired from the model, two vertical use cases, one in the electric distribution network and one in the railway, are studied in this chapter.

5.1 . Tele-action use case

The first use case is based on an electric power distribution network. 5G is planned to be introduced to the energy domain to increase safety, reliability, and efficiency and reduce capital and operational expenditures. The partner company, EDF, and its affiliate plan to implement 5G in Tele-action to enhance fault management in their smart grid.

5.1.1 . Tele-action for electric distribution network

5.1.1.1 Tele-action mechanism

The Tele-action use case originally proposed by EDF in the 5G EVE project [212] refers to remote decoupling protections for the distributed generation power stations in the electric grid. The use case is presented in Figure 5.1. The integration of distributed generators into distribution networks affects the requirements and the performance of conventional protection schemes. It may cause problems such as unwanted islanding of feeders or unwanted disconnection of distributed energy resources during high voltage faults or wide area disturbance, which may cause damages [213]. This problem can be solved thanks to Tele-action.

In Figure 5.1, the considered network includes four distributed energy generation units and one customer load. They connect to the high-voltage network via two feeders.

The whole process of the scenario during a network fault is given as follows:

- At the beginning, the distribution network is in the normal operation mode.
- A fault situation occurs, leading feeder 1 to open the protection.
- The unwanted islanding operation will be avoided by remote decoupling through signaling to the distributed generation stations connected to the segment that feeder 1 controls.

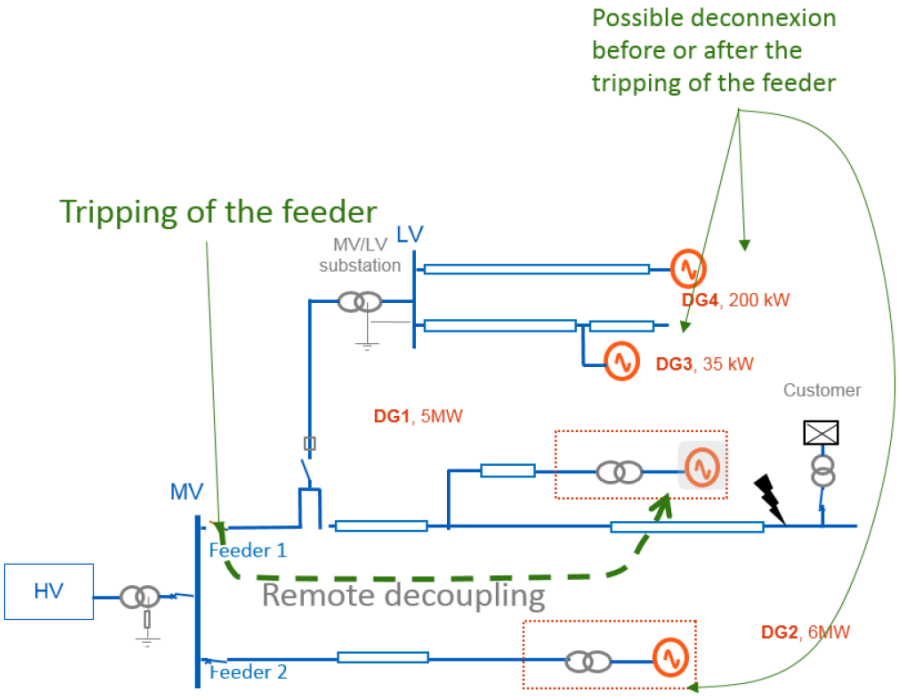


Figure 5.1: The Tele-action use case presentation [212].

5.1.1.2 Tele-action using 5G

Currently, the message signaling of Tele-action is realized through fiber connectivity. As more and more distributed generators, such as solar parks and wind farms, are installed, 5G can replace the fiber wires to reduce deployment costs and possible to ensure a certain level of reliability.

Some KPIs have been targeted in the 5G Tele-action use case:

- The one-way latency should be less than 50 ms.
- The network availability should be above five nines, i.e., 99.999%.

5.1.2 . Resilience evaluation for Tele-action service

The two KPIs from Sub-subsection 5.1.1.2 are addressed. To simplify the case, we only consider the up-link service delivery between a UE, the Tele-action device, and the 5G CN. Then, the signaling message for Tele-action will be sent to the target device by a similar process using a down-link. Thus, half of the total process of Tele-action signal will be considered in this Subsection.

5.1.2.1 Latency estimation

In the normal operation mode, the latency of the Tele-action service can be estimated following the method presented in Subsection 2.4.3.

Since no detailed parameters in the network architecture are currently precised, the work from Section 4.3 can be used as a reference without loss of generality. In the absence of adverse events, the service processing latency is less than 40 ms. The service transmission latency will depend on the distance between the device and the core network. Indeed, the transmission latency only contributes to a small part of the latency, as a distance of 100 km will lead to a latency of 1 ms, according to Subsection 2.4.3. The requirement of 50 ms latency can be satisfied for the device to the 5G CN part service delivery.

Actually, the processing time depends on the capacity of the computation resource. A more powerful server can reduce the service processing latency. If only one UPF function in CN is needed in the device-to-device service delivery, and all servers are high-performance, one-way service delivery latency of less than 50ms can be obtained.

The work from Section 4.4 can bring insights into service protection in the presence of adverse events. Tele-action itself does not generate a great amount of traffic. The Tele-action devices only send signaling orders to others in case of the feeder opening; otherwise, they only send intermittent ping messages to test the connection. The main threats come from outside the service. The results from Section 4.4 and Paper [26] show that a dedicated network slice can be assigned to the Tele-action service to guarantee a low-latency and high-reliability service.

5.1.2.2 Availability estimation

As in the previous scenario presented in Figure 4.6, the SFC we consider for the Tele-action use case only takes into account the essential functions, i.e. the User Plane (UP) part of the network. The Control Plane (CP) part will be neglected as it is considered only serve for session setups phase. The comprises four microservices. The availability of each microservice takes the simulation results from Subsection 4.2.3. Three pods are activated for each microservice, which needs at least one pod to serve the Tele-action packet. Assuming each microservice is independent, it has a dedicated physical resource. The failure can be managed by immediate self-healing.

The availability of these three VNFs can be computed by Equation (5.1).

$$A_{VNFs} = \prod_{i=1}^{4} A_{MS,i} \tag{5.1}$$

The Tele-action service availability should also take into account the availability of the Radio Unit and TN (only taken into account in Tele-action use case) when the service is deployed. In TN, it comprises transport fiber and switches. The essential 5G functions for Tele-action service is represented by Figure 5.2. The availability of the Tele-action service is given by Equation (5.2). The availability-related parameters are given in Table 5.1.

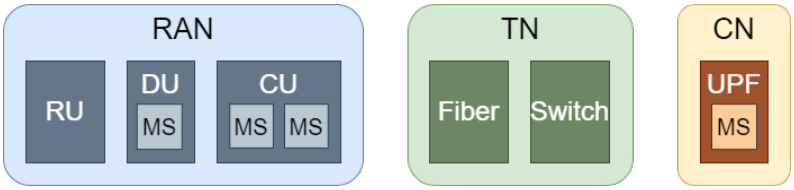


Figure 5.2: The 5G network for a Tele-action use case.

$$A_{Tele-action} = A_{RU} \cdot A_{VNFs} \cdot A_{Fiber} \cdot A_{Switch} \tag{5.2}$$

Table 5.1: Components availability for Tele-action use case. Information collected from Orange internal experts.

Component	Availability	Remarks
RU	99.97317352%	Downtime: 141 minutes per year
Microservice	99.99999983%	Result from Paper II [23]
Fiber (200 km)	99.90487062%	Downtime (per 10 km): 25 minutes per year
Switch	99.99790715%	Downtime: 11 minutes per year

Table 5.2: Tele-action service availability under different network design.

Design	Radio Unit	Fiber	Switch	Availability
1	Single	Single	Single	99.87597870%
2	Single	Single	Double	99.87806895%
3	Single	Double	Single	99.97099009%
4	Single	Double	Double	99.97308233%
5	Double	Single	Single	99.90277191%
6	Double	Single	Double	99.90486272%
7	Double	Double	Single	99.99780879%
8	Double	Double	Double	99.99990159%

The single Radio, fiber line and switch are insufficient to obtain a highly available Tele-action service. They should be reinforced by doubling the equip-

ment in parallel. The availability comparison of different network designs for Tele-action use cases is summarized in Table 5.2. The only case that meets the requirement of five nines is when the Radio, fiber line and switch are all doubled, with the availability of 99.99990159%. The choice of doubled Radio is consistent with the 2-cell parallel connection envisaged by EDF. The doubled fiber and switch deployment is consistent with Orange's transport network design.

5.2 . Railway use case

Most industries will benefit from 5G networks from a static perspective. In a normal operational state, the vertical services stick to the same UP connection using the same network components. While for transportation vertical users, the UP connection is no longer an independent issue. Indeed, to guarantee the UP connection during mobility, the CP becomes indispensable. In this section, the railway telecommunication scenario is introduced. A decomposition method is proposed for simplifying the complex network structure modeling. The resilience perspectives from the network operator and service user are addressed. An analytical method is presented for network perspective resilience evaluation. A simulation framework is developed for network and service perspective resilience evaluation.

5.2.1 . Current and future railway communication systems

5.2.1.1 railway communication services and challenges

For more than 20 years, ground-to-train communication has relied on the GSM-R system based on 2G. The International Union of Railway (UIC) decides to launch a new system, the Future Railway Mobile Communication System (FRMCS), to replace it. As pointed out by [214], the goal is to usher in 5G for rail networks. GSM-R, often reinforced with redundancy in the application, has been, so far, one of the most reliable systems ([215]). Although GSM-R is still a universal solution for the communication between the train and control center, there are many reasons to upgrade this system, such as the end of the GSM-R system life cycle and the need to improve the quality of service and quality of experience ([216]).

5G is undoubtedly the most advanced telecommunication system that will enhance the quality of railway services. The 5G NR extends to a higher spectrum band ([217]), enabling a higher data transfer rate. The 5G Core will be fully virtualized ([218]), providing a flexible and tailored network to train services.

Nevertheless, just as GSM needs to be upgraded with further enhancements specific to the requirements to become GSM-R, 5G networks need to be carefully implemented and designed to adjust to the specific requirements

of railroad operation.

According to [6], seamless communication is crucial for train control service as it conveys important signals guaranteeing the operation of trains. Onboard, seamless communication is also required to provide high-quality services.

However, communication in the high-speed railway scenario faces many challenges. As discussed by [219], most of these challenges could be grouped under four categories: accurate channel estimation, advanced signal processing, optimized network deployment, and effective mobility management. As this work addresses reliability-related issues, we focus mainly on network deployment and mobility management. The failure of the network facility is one of the main reasons a train loses its communication service since it would need to connect to different base stations during its movement. The faster a train moves, the faster it needs to change the anchoring base stations, thus the more network elements it uses during a given time. In network management, the Handover (HO) procedure can be another crucial reliability challenge. As 5G networks introduce a high spectrum band, the dense small-cell ([220]) layout increases HO frequency for high mobility end-users. HO signaling procedure reliability becomes thus more important for providing a seamless connection to high-speed trains.

Some works have addressed the 5G reliability problem, considering low-mobility or non-mobility users ([221, 222, 223]). Some works have investigated the HO process management under high mobility and sought to find a better way to avoid wrong HO, failed HO, or missed HO ([224, 225, 226, 227]). Nevertheless, little attention has been paid to the impact of network infrastructure failure and HO procedure failure on the reliability and availability of high-speed train communication service.

5.2.1.2 Resilience expectations on railway communication services

For now, only "Radio sol-train" [1] relies on GSM-R in the French rail network. In the future, new train services will benefit from the new telecommunication network. Although 5G has not yet officially been put into service in the domain, 3GPP [6] has already launched rail communication normative service requirements for 5G. A summary of communication service performance requirements for rail-bound mass transit is listed in Table 5.3.

These communication services will rely on a 5G network. There exists a different perspective on Communication service performance requirements for rail services. The mission of the communication service provider, the telecommunication network operator, is to provide a resilient service to all trains using

[1] In English: Ground-train radio, a telecommunications system used on part of the French rail network to provide a link between traffic management center and train drivers.

Table 5.3: Communication service performance requirements for rail-bound mass transit. Table adapted from [6].

Use case	Communi-cation service availability	Communi-cation service reliability (MTTF)	End-to-end latency
Control of automated train	99.999%	<1 year but »1 month	<100 ms
CCTV communication automated train surveillance cameras	99.99%	~1 week	<500 ms
Emergency voice call	99.99%	~1 day	<200 ms
Train coupling	99.9999%	~1 year	<100 ms

the rail lines covered by the 5G network. Therefore, the service must be able to be delivered everywhere on the rail lines independent of the position of a train. The demand of the communication service consumer, the train operating company, is to have the trains connected throughout the journey. The risk of failure from the communication network will be diversified as a train is only concerned with a particular part of the entire network. The two different perspectives are important for resilience assessment. The angle from the telecommunication operator has a more global idea of the risks arriving at the network. The angle from the train operator reflects the dynamic behavior of the service for a moving user.

According to Table 5.3, latency is less demanding than in other scenarios discussed. Thus, communication service availability and communication service reliability are selected as the main resilience-related indicators.

5.2.1.3 Challenges dealing with high-speed end-users

When a 5G network is applied to high-mobility scenarios, some network components may need multiple instances and be distributed along the railway track due to the radio coverage distance constraints and service latency requirements. A train will only connect to the RAN components covering it, as shown in Figure 5.3. Therefore, at a given position, the train only establishes an E2E connection via the reachable local RAN components. The local RAN is connected to an aggregated CN. The CN components are often located in a data center far from the RANs. Sometimes, a train runs in an overlapping, reachable by multiple RAN components, like in Zone 2 in Figure 5.3. Sometimes, the train runs in a zone covered by only one RAN radio antenna, as

Zone 1 in Figure 5.3.

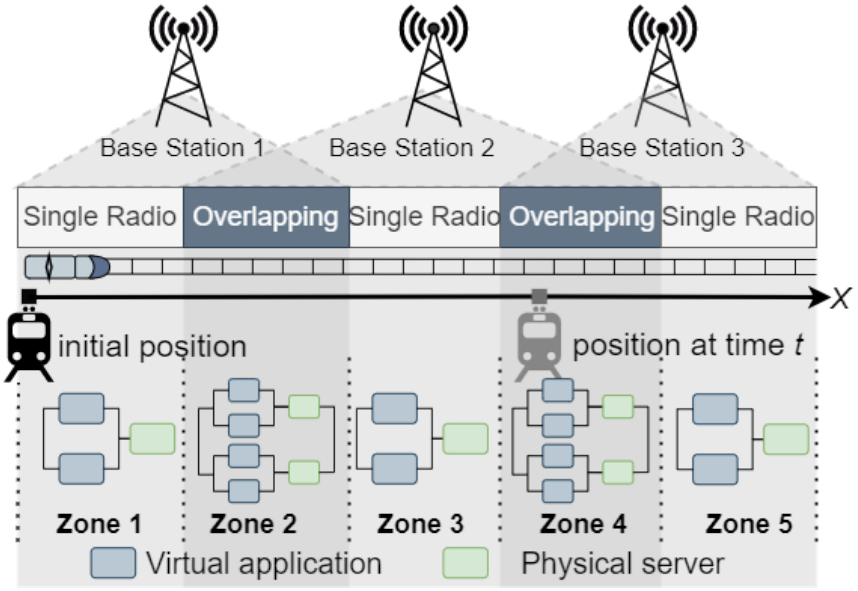


Figure 5.3: Example of 5G network along a railway track. Figure adapted from Paper VII [28].

The 5G network is supposed to provide various high-availability and high-reliability railway communication services, including voice and data communication. These services are based on UP E2E communication. This E2E communication requires all UP network components to operate correctly.

5.2.2 . Analytical model for a complex communication subsystem

5.2.2.1 Network regrouping

The entire system has a hierarchical topology from the edge of the user side to the aggregated CN. At the bottom layer lie the Base stations, mainly composed of a RU (antenna) and a DU. Then, the Central Units, at the second layer, are relatively far from the end-user side, connecting to RUs. Finally, the top layer is the CN.

This tremendous communication system is built for trains operating in a great region. Indeed, a train is usually only connected to one Base Station at a time. Each base station has a particular coverage capacity for a given position so that the train can connect to a limited number of Base Stations. Each Base station only connects to one CU, and each CU connects only to one CN (they can connect to others, but this could generate a latency issue). Thus, a train can only connect to a subset of the entire system at a given position and will connect to a subset of this subset.

Since not all network components are usable for the train at a given position and time, it is possible to simplify the 5G system by considering different

subsystems when assessing the network service availability and reliability.

We set the length of the considered railway as S. Alongside this rail line, N Base Stations $BS = \{bs_1, bs_2, ..., bs_N\}$ are evenly distributed from the start $x = 0$ to the end $x = S$ of the line. Each base station bs_n can effectively transmit radio signals to end-users in a zone with a radius r_n. We divide this rail line into M zones $Z = \{z_1, z_2, ...z_M\}$, such that in zones z_i and z_j, $\forall i, j \in \{1, 2, ..., M\}$, for their the corresponding effective covering Base Station ensembles, $c_i, c_j \subseteq BS$, we have $c_i \neq c_j$. This division ensures that each zone has a unique Base station cover situation. If in zone z_i, $card(c_i) = 1$, it is called a single covering zone. If $card(c_i) \geq 2$, it is called an overlapping zone.

Figure 5.3 shows how zones reconstitute a telecommunication network. The whole system comprises virtual and physical components for RAN and CN. At the moment t, the train is in the middle of the train line of zone z_4 and is reachable to the second and third Radio Base Stations. The covering Base Stations are $c_4 = \{bs_2, bs_3\}$. Each Base Station can establish an E2E connection by using a series-parallel network function system composed of two virtual applications and one physical server (a simplified demonstrative example). The series-parallel of $\{bs_2\}$ is the same as in zone z_4. The series-parallel of $\{bs_3\}$ is the same as in zone z_5. These two series-parallel systems are also in parallel and form a subsystem for zone z_4. When the train enters zone z_5, the only effective covering Base Station is $c_5 = \{bs_3\}$. The subsystem for zone z_5 consists of two virtual components and one physical component. With M zones, the entire system can be regrouped into M subsystems.

The availability of a train network service is the average percentage of available time that the train can connect to the DN via at least one subsystem. The reliability of a train network service is the capacity to provide an E2E connection without failure, which is characterized by the MTBF of the service in the present study.

The availability and reliability of one subsystem provide the availability and reliability for the communication service of a train running at this specific zone. Although some components could belong to multiple subsystems by this regrouping, the failures of these subsystems are assumed to be independent, i.e., the failures in one subsystem will not cause failures in other subsystems. For the train use cases, these subsystems are temporally and spatially independent. At a given moment t, the train is located only at one position and connects to only one subsystem. The train service's available time can be computed as the sum (superposition) of the available time of those subsystems it passes.

$$A_{train} = \frac{\sum_{i=0}^{M} A_{subnet_i} \cdot T_i}{T_{total}} \times 100\% \tag{5.3}$$

Equation (5.3) calculates the service availability. A_{subnet_i} is the availability of

the i-th subsystem. T_i is the train passing time at zone z_i. The train network service availability shows the percentage of time the train can use the E2E communication during the trip.

The number of failures of the train service for a given duration that the train stays in the subsystem can be deduced from the reliability of the subsystem. Then, by assuming these subsystems are independent, it is possible to extract the MTBF for the overall train service.

$$MTBF_{train} = \frac{T_{total}}{\sum_{i=0}^{M} \frac{T_i}{MTTF_i + MTTR_i}} \tag{5.4}$$

In Equation (5.4), the sum of $MTTF_i$ and $MTTR_i$[1] is the MTBF of the i-th subsystem. T_i is the train passing time at zone z_i. The passing time divided by MTBF at zone z_i is the number of failure occurrences when the train passes zone z_i. The train network service reliability indeed describes how often an E2E service interruption may happen during the trip.

5.2.2.2 State space Markov model

A first assumption to simplify the considered subsystems model is that all components, whatever their nature, physical or virtual, their failure processes follow the exponential law, and so do their repair processes. For the virtual elements, for instance, software, their major failures are caused by bugs, often assumed with a constant failure rate [228]. As for the physical server, some research has also simplified the situation by adopting a constant failure rate as in [229] and [230].

Based on this assumption, we create a state space model of the subsystems. The m-th subsystem $S_m = \{e_1, e_2, ..., e_{m_k}\}$ is an ensemble of m_k components. There will be 2^{m_k} states in total, as each element can be either at a working or failed state. Normally, the single Radio Base Station-covered zone subsystem will be less complex than the overlapping zone subsystem since fewer components exist.

An example of a subsystem with three elements (two identical virtual component instances and one server) is the subsystem in Zone 1. The two virtual functions are in parallel to provide redundancy. If one virtual component fails, the other keeps the subsystem's virtual part alive. The server and the virtual functions are in series. The whole subsystem fails under two situations, either the only server or the parallel virtual part fails.

In this subsystem, each component is either in the state "Working" or "Failed". TABLE 5.4 gives the entire eight subsystem states. The reliability of a repairable series-parallel system can not simply be solved using tools like the Reliability Bloc Diagram. The state space models are preferred. We build

[1]The failure detection or diagnose time is included in $MTTR_i$.

a Markov chain [231] with this list of possible states. The possible transition paths are shown in the figure of CTMC in Figure 5.4. λ and μ represent the element failure and repair rate, respectively. With the help of the transition rate matrix of the CTMC, the stationary distribution of the CTMC, π, can be obtained. We can then deduce the subsystem's availability.

Table 5.4: States of the subsystem containing two virtual components and one server. Table from Paper VII [28].

Chain State	**Component state**			**System state**
	Virtual 1	***Virtual 2***	***Server***	
1 (1,1,1)	Working	Working	Working	Working
2 (0,1,1)	Failed	Working	Working	Working
3 (1,0,1)	Working	Failed	Working	Working
4 (1,1,0)	Working	Working	Failed	Failed
5 (0,0,1)	Failed	Failed	Working	Failed
6 (1,0,0)	Working	Failed	Failed	Failed
7 (0,1,0)	Failed	Working	Failed	Failed
8 (0,0,0)	Failed	Failed	Failed	Failed

The set of chain states $CS \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ corresponds to the combination of component states in the subsystem. Simulation results show that the subsystem stays short at a transient state and moves fast to a steady state. A detailed example will be given in Section 5.2.2.3. Supposing $p(i, t), i \in \{1, 2, 3, 4, 5, 6, 7, 8\}$ represents the probability of the subsystem being at state i at time t. In steady-state SS, the probability of the subsystem at states $\{1, 2, 3\}$ is $p(SS = \text{“Working"}) = \lim_{t\to+\infty} p(\{1, 2, 3\}, t)$. This distribution gives us a rapid answer to compute subsystem availability, equivalent to the sum of the first three items of π (sum of the working state probabilities).

The stationary distribution of the subsystem states $\pi = \{p_{1\infty}, p_{2\infty}, ..., p_{8\infty}\}$ can be directly computed from the transition matrix of the CTMC. The availability of the series-parallel subsystem by adding the stationary distribution of all “Working" states is:

$$A_{subnet} = \sum_{i=1,2,3} p_{i\infty} \tag{5.5}$$

However, it could be more complicated when computing the subsystem's reliability. Instead of looking at all changes of states, we consider two Discrete-Time Markov processes: the failure process and the repair process.

For the failure process, we consider the transitions inside the “Working" states and from the “Working" states to the “Failure" states. This process

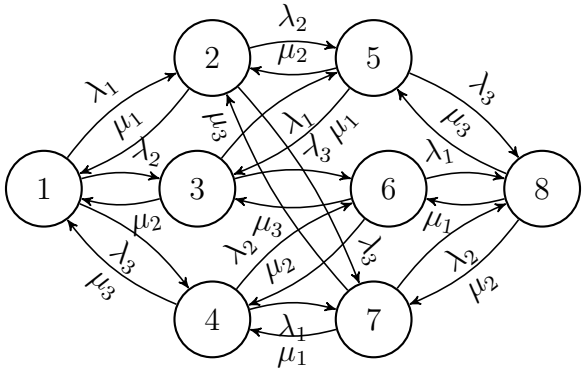


Figure 5.4: Subsystem represented by a Continuous-Time Markov Chain. Figure from Paper VII [28].

starts from the subsystem's recovery and ends with the state changed to $CS \in \{4, 5, 6, 7\}$, represented by recurrent states, as shown in Figure 5.5. The transition probability of this DTMC is deduced from the CTMC. It shows how the subsystem definitively changes from one state to another. Each transition corresponds to a state-changing event. For example, the state-changing probability from state 1 to state 2 in Figure 5.5 is the probability of moving to state 2 after the first state-changing event from state 1. We use variables τ_1, τ_2, τ_3 to represent the failure time of component 1, 2 and 3. The state-changing probability from state 1 to state 2 is Prob$\{\min\{\tau_1, \tau_2, \tau_3\} = \tau_1\} = \frac{\lambda 1}{\lambda 1+\lambda 2+\lambda 3}$.

We describe this Markov Chain of failure process by a stochastic transition matrix P_F. The initial state is the state of the very moment that the subsystem is repaired to the "Working" state. The chain has a final state as the failure process always ends with the connection becoming unavailable. That is the state of the very moment that the subsystem for the first time goes into the "Failure" state. Since state 8 is not a direct "Failure" state and is only reachable from another "Failure" state $CS \in \{4, 5, 6, 7\}$, state 8 is not engaged during this process. As a result, we define the process initial state distribution π_F^0 and final state distribution π_F^∞. We get the following relations:

$$\pi_F^0 = [p_{F1}^0, p_{F2}^0, p_{F3}^0, p_{F4}^0, p_{F5}^0, p_{F6}^0, p_{F7}^0, p_{F8}^0] \tag{5.6}$$

$$\pi_F^\infty = [p_{F1}^\infty, p_{F2}^\infty, p_{F3}^\infty, p_{F4}^\infty, p_{F5}^\infty, p_{F6}^\infty, p_{F7}^\infty, p_{F8}^\infty] \tag{5.7}$$

$$\lim_{k \to +\infty} \pi_F^0 \times {P_F}^k = \pi_F^\infty \tag{5.8}$$

where: $\sum_{i=1}^{8} p_{Fi}^0 = 1$, and $p_{Fi}^0 = 0$ for $i \in \{4, 5, 6, 7, 8\}$

$\sum_{i=1}^{8} p_{Fi}^\infty = 1$, and $p_{Fi}^\infty = 0$ for $i \in \{1, 2, 3, 8\}$

For the repair process, we consider the opposite. All start from the "Failure" states $CS \in \{4, 5, 6, 7\}$. This process ends by reaching the states $CS \in$

$\{1, 2, 3\}$, represented by recurrent states, as shown in Figure 5.6. The transition probability is also deduced from the CTMC of the subsystem.

We describe this Markov Chain by a transition matrix P_R. The initial state is the state of the very moment that the subsystem failed to a “Failure" state. The final state is the state of the very moment that the subsystem, for the first time, goes into a “Working" state. Since state 8 is only reachable from another “Failure" state $CS \in \{4, 5, 6, 7\}$, the initial “Failure" state can not start from the state 8. As a result, we define the process initial state π_R^0 and final state π_R^∞.

We get the following relations:

$$\pi_R^0 = [p_{R1}^0, p_{R2}^0, p_{R3}^0, p_{R4}^0, p_{R5}^0, p_{R6}^0, p_{R7}^0, p_{R8}^0] \tag{5.9}$$

$$\pi_R^\infty = [p_{R1}^\infty, p_{R2}^\infty, p_{R3}^\infty, p_{R4}^\infty, p_{R5}^\infty, p_{R6}^\infty, p_{R7}^\infty, p_{R8}^\infty] \tag{5.10}$$

$$\lim_{k \to +\infty} \pi_R^0 \times P_R^k = \pi_R^\infty \tag{5.11}$$

where:
$\sum_{i=1}^{8} p_{Ri}^0 = 1$, and $p_{Ri}^0 = 0$ for $i \in \{1, 2, 3, 8\}$
$\sum_{i=1}^{8} p_{Ri}^\infty = 1$, and $p_{Ri}^\infty = 0$ for $i \in \{4, 5, 6, 7, 8\}$
When the subsystem is in its steady state, we have the following relations:

$$\pi_F^\infty = \pi_R^0 \tag{5.12}$$

$$\pi_R^\infty = \pi_F^0 \tag{5.13}$$

By solving Equations 5.6 - 5.13, we obtain the initial and state distribution of “Working" and “Failure" states.

Each transition step corresponds to a sojourn time in the DTMC. We define the state sojourn time T_i^s as the mean time between the subsystem entering state i and leaving the state i in the CTMC. We also define the mean state failure time T_i^F as the mean time between the subsystem entering the “Working" state i and the first time entering a “Failure" state. It is the sum of a set of transition steps in the DTMC for the failure process. The MTTF we intend to compute is the mean state failure time of all “Working" states.

$$MTTF = \sum_{i \in \{1,2,3\}} \frac{p_{Fi}^0}{\sum_{j \in \{1,2,3\}} p_{Fj}^0} \cdot T_i^F \tag{5.14}$$

Note that for this case, $\sum_{j \in \{1,2,3\}} p_{Fj}^0 = 1$.

The Markov Chain of the failure process in Figure 5.5 gives the following relations:

$$T_1^F = T_1^s + \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3} \cdot T_2^F + \frac{\lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \cdot T_3^F \tag{5.15}$$

$$T_2^F = T_2^s + \frac{\mu_1}{\mu_1 + \lambda_2 + \lambda_3} \cdot T_1^F \tag{5.16}$$

$$T_3^F = T_3^s + \frac{\mu_2}{\lambda_1 + \mu_2 + \lambda_3} \cdot T_1^F \tag{5.17}$$

For Equation (5.15), the average failure time T_1^F includes the time spent in different steps. In the first step, the subsystem leaves state 1 and spends time T_1^s, the CTMC sojourn time in state 1. According to the state transition probabilities, from state 1, the subsystem may change to state 2, 3, or 4. In the next step, the process ends if the subsystem directly fails to state 4. Otherwise, it will spend time T_2^F and T_3^F accordingly for the rest of the failure process. The average failure time of state 2 and 3 can be represented similarly in Equations 5.16, 5.17. Finally, the MTTF is obtained by solving Equations 5.14 - 5.17.

The MTTR is the total transition time of a subsystem being repaired during failure. We define the mean state repair time T_i^R as the average time between the subsystem entering a specific “Failure" state i and the first time entering a “Working" state. Therefore, the MTTR is the mean sojourn time of all “Failure" states.

$$MTTR = \sum_{i \in \{4,5,6,7,8\}} \frac{p_{Ri}^0}{\sum_{j \in \{4,5,6,7,8\}} p_{Rj}^0} \cdot T_i^R \tag{5.18}$$

Note that for this case, $\sum_{j \in \{4,5,6,7,8\}} p_{Rj}^0 = 1$ and $p_{R8}^0 = 0$.

The Markov Chain of the repair process in Figure 5.6 gives the following relations:

$$T_4^R = T_4^s + \frac{\lambda_1}{\lambda_1 + \lambda_2 + \mu_3} \cdot T_7^R + \frac{\lambda_2}{\lambda_1 + \lambda_2 + \mu_3} \cdot T_6^R \tag{5.19}$$

$$T_5^R = T_5^s + \frac{\lambda_3}{\mu_1 + \mu_2 + \lambda_3} \cdot T_8^R \tag{5.20}$$

$$T_6^R = T_6^s + \frac{\mu_2}{\lambda_1 + \mu_2 + \mu_3} \cdot T_4^R + \frac{\lambda_1}{\lambda_1 + \mu_2 + \mu_3} \cdot T_8^R \tag{5.21}$$

$$T_7^R = T_7^s + \frac{\mu_1}{\mu_1 + \lambda_2 + \mu_3} \cdot T_4^R + \frac{\lambda_2}{\mu_1 + \lambda_2 + \mu_3} \cdot T_8^R \tag{5.22}$$

$$\begin{aligned} T_8^R = T_8^s + \frac{\mu_1}{\mu_1 + \mu_2 + \mu_3} \cdot T_6^R + \frac{\mu_2}{\mu_1 + \mu_2 + \mu_3} \cdot T_7^R \\ + \frac{\mu_3}{\mu_1 + \mu_2 + \mu_3} \cdot T_5^R \end{aligned} \tag{5.23}$$

For Equation (5.19), the average repair time T_4^R includes the time spent in different steps. In the first step, the subsystem leaves state 4 and spends time T_4^s, the CTMC sojourn time in state 4. According to the state transition probabilities, from state 4, the subsystem may change to state 1, 6, or 7. In the next step, the process ends if the subsystem is directly repaired to state 1. Otherwise, it will spend time T_6^F and T_7^F accordingly for the rest of the failure process. The average failure time of states 5, 6, 7, and 8 can be represented similarly. Finally, the MTTR is obtained by solving Equations 5.18 - 5.23.

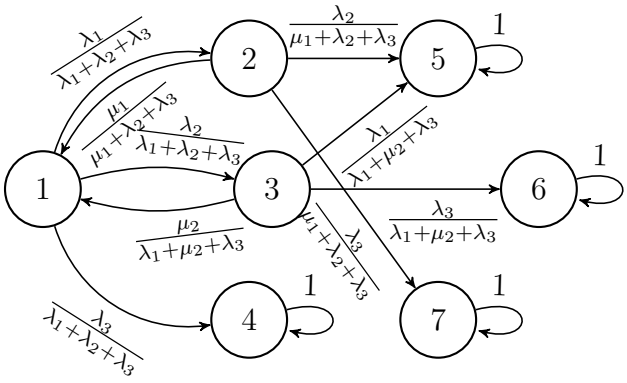


Figure 5.5: Failure process represented by a Discrete-Time Markov Chain. Figure from Paper VII [28].

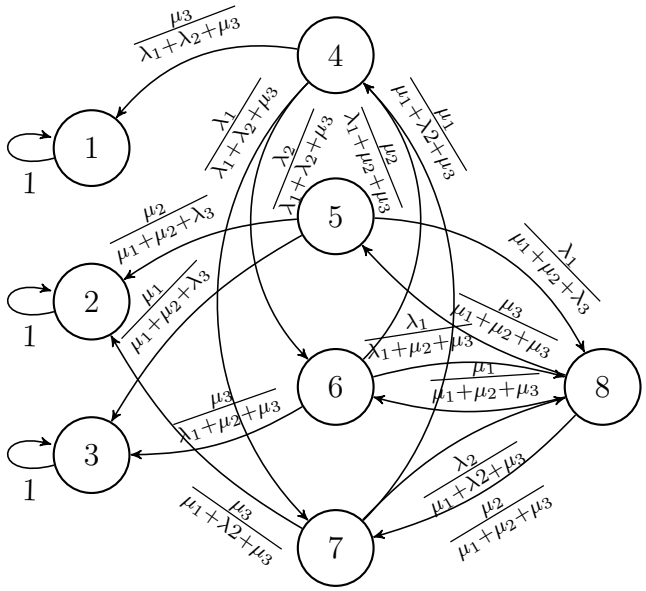


Figure 5.6: Repair process represented by a Discrete-Time Markov Chain. Figure from Paper VII [28].

Although only a three-element system is demonstrated in the example, the proposed method can also be applied to any series-parallel system. However, the state space will increase exponentially with the number of considered components.

5.2.2.3 Availability and reliability estimation

Inspired from [232], under the context of 5G for high-mobility trains, the network service availability can be defined as the probability that the E2E connection is available at any instant. Reliability is often used to characterize if a system is appropriately working during a specific period of time [233]. The 5G network we consider is a repairable system. We use MTTF, the average time the E2E connection lasts, and MTTR, the average time to repair the E2E connection, to estimate the network service reliability. When the time moment in the availability definition t tends to infinity, the steady-state availability equals MTTF/(MTTF + MTTR) [234].

Example of a three-element subsystem

Now we consider a system with two virtual components, #1 and #2, and one physical component, #3. The virtual components are the applications that are often threatened by operational failures. The physical component often refers to a physical server where the applications are hosted, which is less likely to fail. Repairing a virtual component takes only a few seconds by restarting the application. However, when a physical server fails, it must be repaired manually. TABLE 5.5 shows the failure and repair rates.

Table 5.5: Failure and repair rates of components. Table from Paper VII [28].

Failure process			
Component	Symbol	Rate [hour^{-1}]	MTTF
1 - virtual	λ_1	0.005	200 hours
2 - virtual	λ_2	0.005	200 hours
3 - physical	λ_3	0.0002	5000 hours
Repair process			
Component	Symbol	Rate [hour^{-1}]	MTTR
1 - virtual	μ_1	360	10 seconds
2 - virtual	μ_2	360	10 seconds
3 - physical	μ_3	1	1 hour

After building the CTMC model and the transition rate matrix, we calculate the transient availability of such system as shown in Figure 5.7. Initially, the brand new subsystem has 100% availability. After a few hours, it gradually drops to the stationary availability around 99.98%. The steady state of this CTMC also gives us a similar result as shown in TABLE 5.6. The availability of

the subsystem is $A_{subsystem} = p^1_\infty + p^2_\infty + p^3_\infty = 99.9800\%$. This result shows that, at a stationary state, 99.9800% of the time, this subsystem is available to provide application service to the end-user.

Table 5.6: Stationary state distribution of the subsystem. Table from Paper VII [28].

State	Probability	State	Probability
1	9.99772e−1	5	1.92857e−10
2	1.38857e−5	6	2.77715e−9
3	1.38857e−5	7	2.77715e−9
4	1.99954e−4	8	3.85715e−14

As for reliability, two DTMCs are built for failure and repair processes. The subsystem reparation processes are assumed to be parallel, i.e., each component can fail or be repaired independently. Equations 5.6 - 5.13 give the initial and final states of failure and repair processes as shown in Table 5.7.

Table 5.7: Initial and final state distribution of the subsystem. Table from Paper VII [28].

Failure process		Repair process	
p^0_{F1}	9.99278e−1	p^∞_{R1}	9.99278e−1
p^0_{F2}	3.60850e−4	p^∞_{R2}	3.60850e−4
p^0_{F3}	3.60850e−4	p^∞_{R3}	3.60850e−4
p^∞_{F4}	9.99278e−1	p^0_{R4}	9.99278e−1
p^∞_{F5}	6.93943e−4	p^0_{R5}	6.93943e−4
p^∞_{F6}	1.38789e−5	p^0_{R6}	1.38789e−5
p^∞_{F7}	1.38789e−5	p^0_{R7}	1.38789e−5

Using Equations 5.14 - 5.23, we obtain the MTTF and MTTR of the subsystem. MTTF of the subsystem is 4996.53 hours, and MTTR is 0.999307 hours. The physical server failure primarily dominates the subsystem failure time, and the repair time is also dominated by physical server repair because, unlike the virtual components, the physical component is not designed with redundancy in the subsystem. It shows that a possible way to improve the subsystem availability is to reduce physical component failure and repair time.

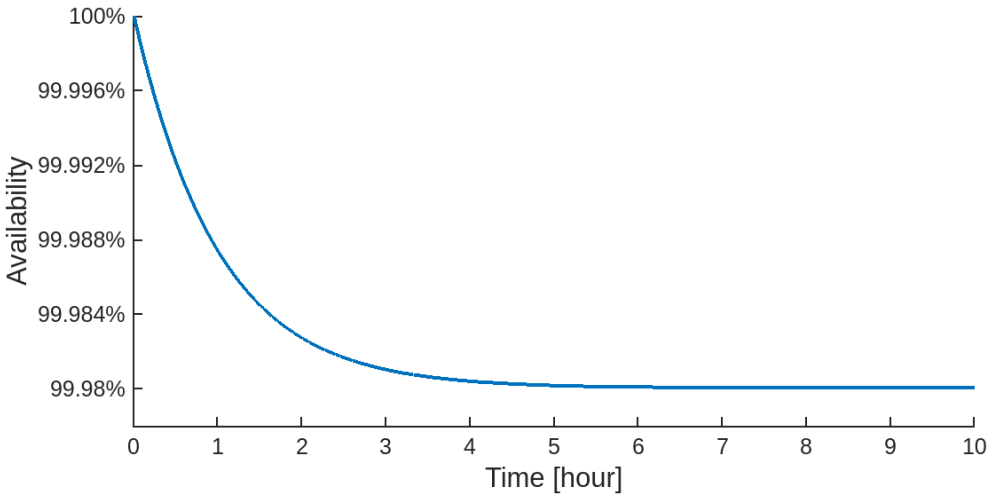


Figure 5.7: Transient availability of the subsystem. Figure from Paper VII [28].

From subsystem to the whole system

The considered railway is 100 km long. A train runs at a constant speed, 200 km per hour. We assume the trains are well-timed and always pass the zone at a fixed time. The information of each zone is given in TABLE 5.8.

Table 5.8: Subsystem characteristics of each zone. Figure from Paper VII [28].

Zone	Expected passing time [min]	Availability	MTBF
1	3.6	99.980004%	4997.5 hours
2	6.6	99.999996%	1426.5 years
3	2.4	99.980004%	4997.5 hours
4	4.5	99.999996%	4997.5 hours
5	5.5	99.980004%	1426.5 years
6	6.0	99.999996%	4997.5 hours
7	3.6	99.980004%	1426.5 years

It is ideally assumed that along the railway line, all Radio Base Stations with their connected components in Figure 5.3 have a similar system structure. Each of them forms a subnetwork as the one in Section 5.2.2.3. In Zone 1, 3, 5, and 7, the subsystem is the same as in Section 5.2.2.3. While in Zone 2, 4, and 6, the subsystems are in the form of two subnetworks of Section 5.2.2.3 working in parallel. The reliability and availability of these subsystems are computed following the proposed method.

The average available network service time and average number of network service failures when a train passes each of the seven zones are given in TABLE 5.9. By summing up the result in the subsystems, the total mean network service available time of the 100 km route is 29.99742 minutes out of a 30-minute ride. The network service availability is 99.9914%. The total mean number of generated failures is $4.30441\text{e}{-5}$. In other words, there will be one failure about every 11616 running hours, which is one failure every 16 months if the train keeps running on this railroad section 24 hours per day.

Table 5.9: Mean service available time and mean service failures. Table from Paper VII [28].

Zone	Available time [min]	Failures
1	3.5992801	$1.20059\text{e}{-5}$
2	6.5999997	$8.80260\text{e}{-9}$
3	2.3995201	$8.00395\text{e}{-6}$
4	4.4999998	$6.00177\text{e}{-9}$
5	3.2993401	$1.10054\text{e}{-5}$
6	5.9999998	$8.00236\text{e}{-9}$
7	3.5992801	$1.20059\text{e}{-5}$

In order to achieve a highly resilient train service, some potential improvements can be made. The first improvement is adding parallel virtual components to each unitary subsystem connected to the Base Station. Instead of 2 virtual components, the unitary subsystem has been upgraded to 3. The second improvement could be adding a redundant parallel physical server to the unitary subsystem. The service availability and reliability comparison is showcased in Figure 5.8. Adding parallel virtual components has less impact on availability and reliability since the virtual element is already redundant. Adding a redundant physical component can vastly improve both availability and reliability. On average, the train can connect to network service for over 10 thousand months without interruption. The availability improved to more than seven nines, largely above the requirement.

5.2.3 . Simulation model for high-speed communication service resilience analysis

If all the subsystems have a similar size, the analytical solution can be scaled to a large railway telecommunication network. However, the assumption of having only two virtual elements and one physical element is ideal. Each VNF can be an ensemble of multiple virtual pods and physical nodes. Besides, if the VNFs in the UP of a CN are also considered, the state space

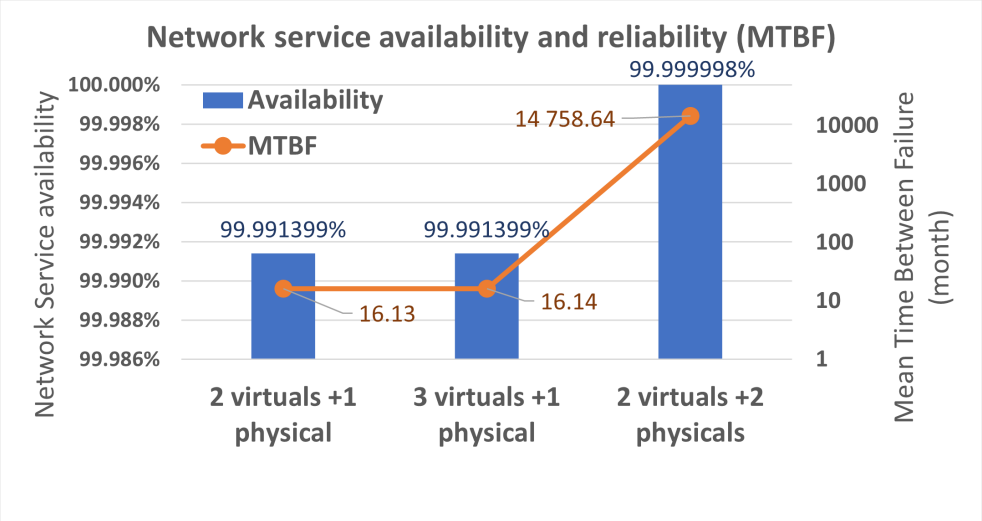


Figure 5.8: Service availability and reliability comparison. The reliability is described by MTBF. Figure from Paper VII [28].

of a subnetwork becomes tremendous. Therefore, the analytical model is no longer the most practical way to estimate network resilience precisely. Running simulations can help generate large-scale failure scenarios and estimate resilience close to real value.

5.2.3.1 Railway-telecommunication network model

We consider a generic 5G network composed of the RAN and the CN. The network architecture is presented in Figure 5.9. RAN, which transmits, receives, converts, and processes the signal, comprises a set of gNodeB (gNB) base stations (5G radio base station), and each is composed of RUs, DUs, and CUs. The CN, consisting of different VNFs that take charge of aggregation, authentication, service control, etc., is divided into the UP with User plane Function (UPF) and the CP including VNFs such as Access and Mobility Management Function (AMF), Session Management Function (SMF)), Unified Data Management (UDM),Authentication Server Function (AUSF), etc. As an end-user, a train will connect to the RU with the best signal that covers the area it passes via a 5G NR air interface. Once the train is registered to the network, it will request a PDU session to start an E2E UP connectivity between the UE and DN. This connectivity is supported by UP, that is, RU, DU, CU-UP, UPF, and the links between them.

The main problem addressed in this work is the reliability and availability-related challenges of communication services applied to high-speed trains. More precisely, a train is considered connected to the internet if the user is registered to the network and it has initiated a PDU session and the whole UP

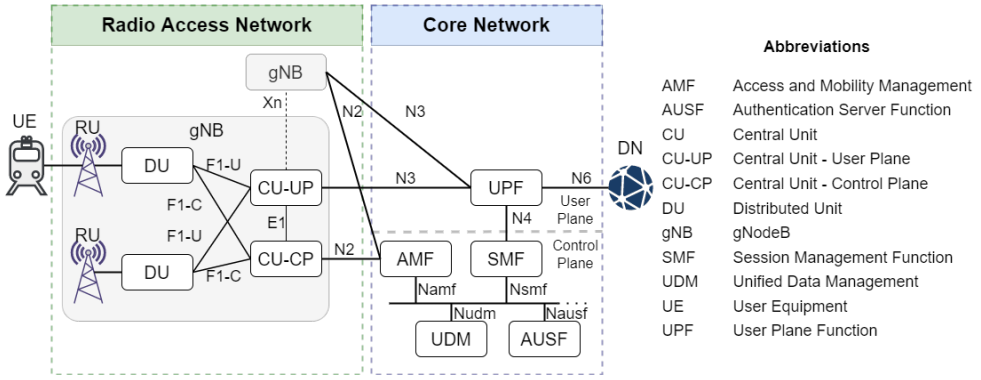


Figure 5.9: A representative 5G network architecture for railway. The 5G network comprises multiple gNB and one CN. The gNB used by the train is detailed with its components. Figure from Paper VI [27].

allocated by the PDU session is reachable and available to the train. We distinguish in the dissertation two kinds of connection failure: the failure related to UP failure and the failure related to reachability.

User Plane failure

When a train starts to travel on the railway, we assume that it is already registered to the network. While the train is running, failures from different parts of the network will impact the communication service in different ways:

- If the gNB[1] facility (including RU, DU, and CU-UP) fails, the train directly loses the connection to DN. There are two possible solutions to reconnect to the DN. If there is another available gNB covering the train, then the train will try to re-establish the connection via this available gNB by a re-establishment procedure. Otherwise, the train becomes unconnected and untraceable. Communication service is stopped. The train will wait until the gNB is repaired or until it enters an available gNB coverage area.

- If the UP in CN fails, i.e., UPF-UP fails, the E2E communication service is interrupted, yet the train is still attached to the gNB. The communication service resumes after the recovery of CN UP.

The Re-establishment procedure ([235]) is simplified by considering the call flow involving only the RU, DU, CU, AMF, and UPF.

Reachability failure

[1]A gNB can be a Base Station as in the Section 5.2.2 or an extended Base Station according to its scale.

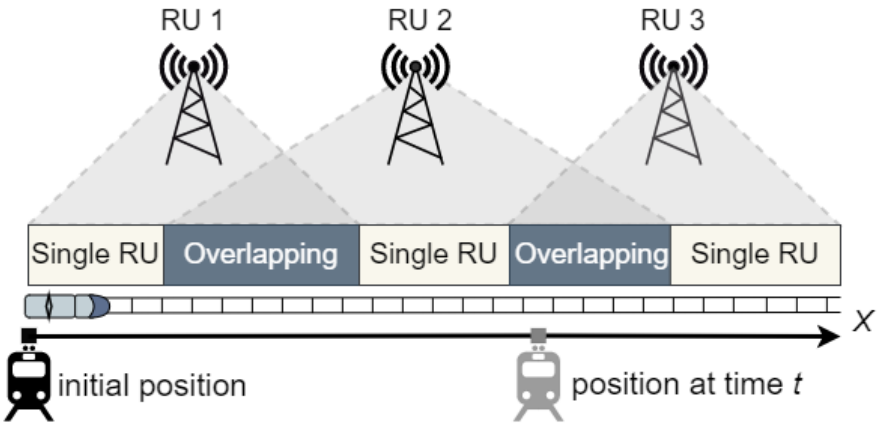


Figure 5.10: An example of 5G gNB RU layout along a section of railway. Figure adapted from Paper VI [27].

Since the train is in high mobility, the RU to which it connects can only serve a specific area, as shown in the radio layout example in Figure 5.10. To guarantee a seamless connection, the train regularly changes the connected RU by HO process at the overlapping covered by multiple RUs. There are different types of HO regarding the implementation and layout of 5G ([235]). In the scope of this work, we consider two of them:

- Inter gNB-DU and Intra gNB-CU Handover: In this HO procedure, the new and old gNB-DUs are connected to the same CU. The signaling message will not necessarily be sent to CN. This procedure will involve messaging over the source and target RUs, DUs, and their CU.
- Inter gNB-CU Handover: In this HO procedure, the signaling will involve messaging over the source and target gNBs (including RUs, DUs, CU), AMF, and UPF.

If the HO procedure fails, the train stays connected to the previous RU. When the RU is no longer reachable to the train, the train will be disconnected from the network and need to re-establish the connection to resume the communication service.

We divide the entire 5G network into different sections as represented in Figure 5.10. Each section has a different layout and can either be a single RU area or an overlapping area. For each section, it is composed of a set of network elements. The UP comprises the UP functions in RAN and UPF-UP in CN. The CP comprises the CP functions in RAN and CN, such as AMF, SMF, and UPF-CP. These functions are a set of physical servers and virtual applications (software). We assume they all have similar behavior as shown in Figure 5.11. They all start from a working state (W) and may fall into a failed state (F) due to

software and hardware reasons. This failure will be detected and identified (N). Finally, it will be either fixed automatically for software and application issues or repaired manually (R). When the element is not in the state (W), all end-users relying on this element fail to use the element, leading to a service connection or a signaling procedure (re-establishment or HO) failure.

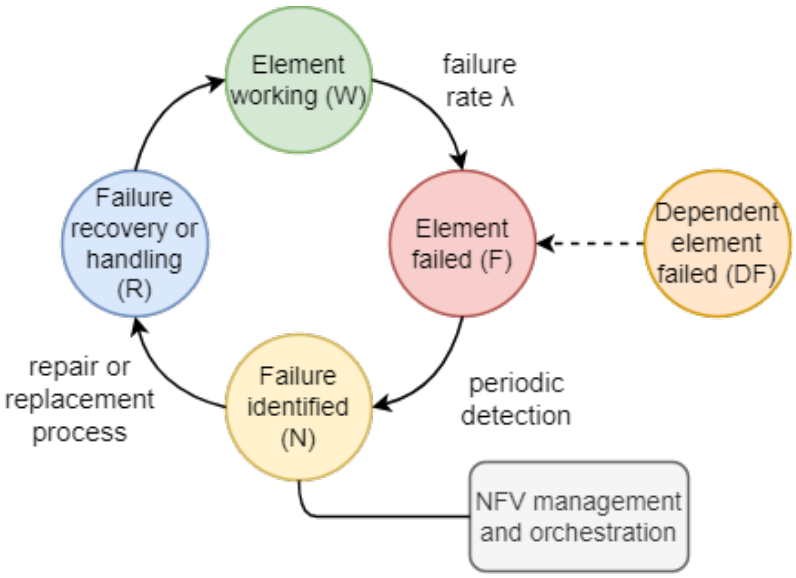


Figure 5.11: Network element life cycle model. Figure from Paper VI [27].

From an end-user's perspective, the train is always in a moving situation. We divide the train's mission into a series of rounds. Each round is represented by Figure 5.12. A round starts from the state where the train is initially connected to i^{th} RU.

If the train runs into a Single RU area, it will stay at the connected state unless the connection fails (some of the network elements it uses are in states (F)). If the failure is due to UPF-UP, the train can return to the connected state when UPF-UP is repaired. If the gNB fails, the train will try to re-establish the connection to i^{th} RU if the failed gNB is repaired, and the train then goes back to the connected state. If the train fails to re-establish the connection, it will remain disconnected until a successful re-establishment to j^{th} RU when entering an overlapping zone, where $j \neq i$.

If the train runs into an overlapping area, it can request HO when a better signal is found. If the HO procedure succeeds, the train will connect to j^{th} RU, where $j \neq i$. If the HO procedure fails, the train will retry HO until the train runs outside of overlapping zone. Then, the train will loose connection and re-establish the connection instead of requiring HO. In this overlapping area, the connection is also at risk of facility failure. As another RU existed in the overlapping zone, should the i^{th} RU fails, it would immediately try to re-establish the connection to the other RU, j^{th} RU, where $j \neq i$.

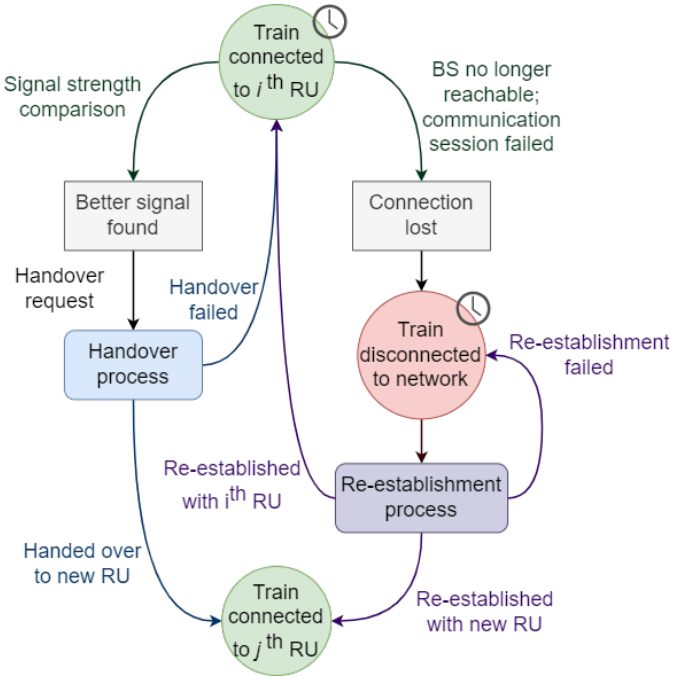


Figure 5.12: High-speed train model.

Both the HO and re-establishment processes change the state of a train by generating a call flow. The re-establishment process changes a train from a non-connected state to a connected state. The HO process allows a connected train to be handed over to another available RU. The train remains connected throughout the HO process.

5.2.3.2 Different perspectives on availability and reliability from telecommunication and railway operators

To analyze the reliability challenges, the reliability-related terms should be defined. For the considered network, we define the availability and reliability from both network and high-speed train communication service perspectives:

- We define network availability as the percentage value of the amount of time the network operator can provide E2E service and response to CP signaling messages everywhere by using the 5G network deployed in a considered area, divided by the total considered time.

- We define network reliability as the ability of the 5G network to provide E2E connection and response to CP signaling messages everywhere in a considered area. We measure network reliability using the MTTF of the considered network system.

- We define network communication service availability as the percentage value of the amount of time the E2E communication service is delivered, divided by the amount of time the train network communication service is expected to be delivered.

- We define network communication service reliability as the ability of the communication service to perform as required for a given time interval under given conditions. We describe network communication reliability using MTTF of the train communication service.

To summarize, the telecommunication operator's perspective focuses on providing a resilient network for whoever uses the network, wherever the user is, and whenever the user needs the network. They should provide more than a user may consume. While the railway operator is the service consumer, it looks into whether the train can use the network at a given moment and location. Indeed, the second perspective diversifies the risks in the entire network by concentrating on a specific subnetwork that the train is able to connect to.

5.2.3.3 Communication network availability and reliability estimation

We still investigate how network layout would impact network and service performance, similar to the Sub-subsection 5.2.2.3. However, this time, we are not only looking at the UP network performance but also the CP. Therefore, the network availability and reliability should include both E2E service and CP signaling messages. Apart from network operator's perspectives, the train users' perspectives are also considered in the estimation.

We implement the proposed models in Sub-subsection 5.2.3.1 with the SimPy environment. We consider a railway line of 100 km (about the distance Paris $\leftrightarrow$ Amiens) with locally distributed RAN and one aggregated CN. The gNBs in RAN consist of co-located RUs and DUs at the edge data center and one aggregated CU at the gNB level data center. RUs are assumed to be purely physical equipment and are equally spaced alongside this 100 km line. Throughout the simulation, one train runs every hour from the start to the end of the line at a fixed speed of 200 km/h. The simulation covers 100,000 hours, equivalent to about 11 years. The train is assumed to run at a speed of 200 km/h. The total journey of the train is therefore 30 minutes. We assume that one train will run from the beginning to the end of the journey. 20 simulations are run in order to generate a large number of scenarios. The first RU is at the starting point of the railway, and the last RU is at the endpoint. The RUs in this scenario can cover an area with a radius of 5 km using the spectrum it can provide. The failure process of the network system is given in Table 5.10, according to the data provided by the network service suppliers.

Table 5.10: Failure processes of network system for the train use case. Table from Paper VI [27].

Item	MTTF	repair time
RU	50 years constant failure rate	1 hour fixed repair time
Virtual application (container)	52 days constant failure rate	10 s $U(0, 10)$ continuous uniform distributions
Server	1 year constant failure rate	1 hour fixed repair time

Table 5.11: Components of network system for the train use case. Table from Paper VI [27].

Items	Instances	Description
RU	Variable	Physical equipment
DU	1 for 1 RU	1 app and 1 server
CU	1 pair for 8 DUs	2 apps and redundant servers
UPF	1 in total	2 apps and redundant servers
AMF	1 in total	1 app and redundant servers

All network links in this study are assumed ultra-reliable. The composition of our envisioned 5G network is given in Table 5.11.

Unreliable Radio Unit

In the first case, we simplified the network elements to explain better the different perspectives from the network and the train. Only RUs will fail in the network, and the rest of the system is highly reliable. For the network operator, the network availability and reliability are strictly defined by considering the capability to provide E2E connection and signaling message response at every position (including both single RU zones and overlapping zones) in the considered area. If at least one zone is not covered by any working RU, the network is considered unavailable. For a high-mobility user, the train, the system is changing between a single RU system and an overlapping system dynamically as it travels. If no working RU can be reached, the service is considered failed.

We investigate how the density of radio installations may impact the network and service communication reliability. The number of overlapping (double RU) zones and single covering zones is shown in Figure 5.13.

Via Monte-Carlo simulation, we compared the impact of different num-

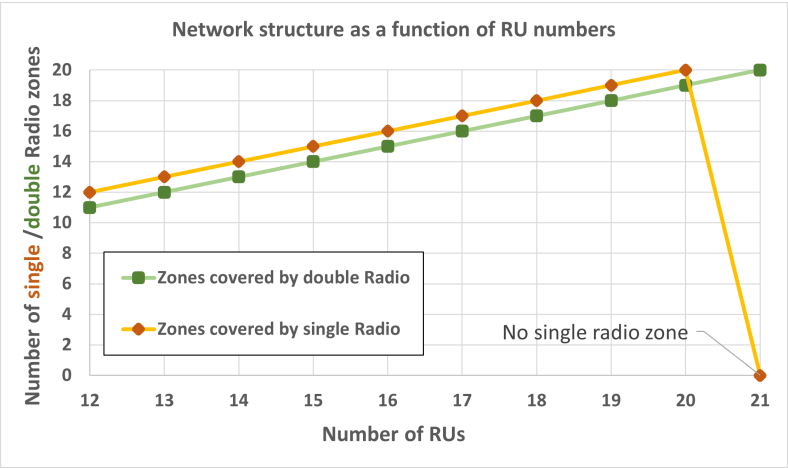


Figure 5.13: Number of single and double covering zones in function with number of RUs.

bers of RUs, varying from 12 to more than 20. Figure 5.14 and Figure 5.15, Table 5.12 and Table 5.13 show the availability and reliability metric MTTF for network and service. A direct computation of the series-parallel system helps us validate this result.

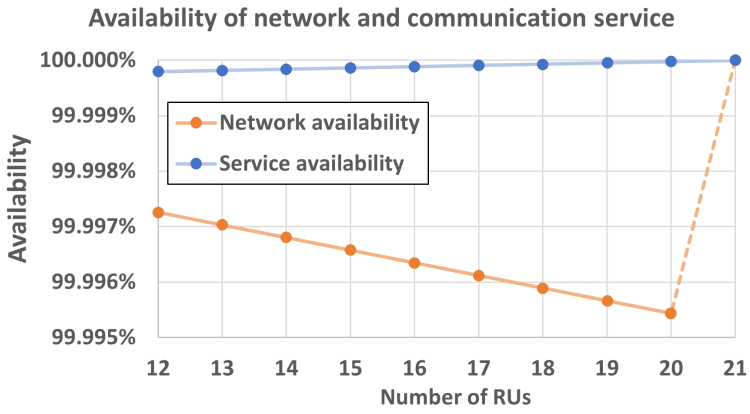


Figure 5.14: Impact of number of RUs on network and service availability.

Obviously, both availability and reliability from these two perspectives are different. For operators, when the number of RUs is below 20, some parts of the railway are always covered by a single RU. The more RU is densely installed, the more single RU zones there will be[1]. The network availability

[1]For example, unless the distance between two RUs is greater than the covering radius, if N RUs are installed evenly along the rail line, there will be N single radio cover zones and $N-1$ double radio cover zones between the first RU and the last RU.

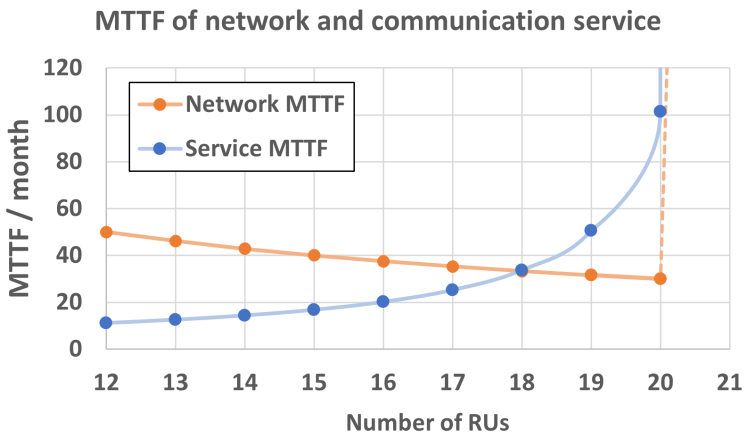


Figure 5.15: Impact of number of RUs on network and service MTTF.

Table 5.12: Network performance in function of RU from network operator's perspective.

Number of RUs	Availability	MTTF [hour]
12	99.86058%	54.66902
13	99.84895%	52.11449
14	99.83789%	49.75021
15	99.82612%	47.56593
16	99.81485%	45.59728
17	99.80219%	43.76260
18	99.79151%	42.08143
19	99.78031%	40.54534
20	99.76875%	39.07220

Table 5.13: Network performance in function of RU from train user's perspective.

Number of RUs	Availability	MTTF [hour]
12	99.99456%	358.79129
13	99.99512%	344.02964
14	99.99571%	332.85248
15	99.99628%	319.44676
16	99.99686%	308.44795
17	99.99742%	297.61343
18	99.99801%	288.29191
19	99.99858%	279.12699
20	99.99917%	270.34631

and MTTF thus decrease with the number of RUs. However, if the number of RUs exceeds 20, there is a sudden jump. In fact, the RU setup is considered fully redundant[1] everywhere, covered by at least two RUs (this redundant layout, in reality, is often not affordable for a network operator). The network service availability obtains nine nines (99.9999999%), and the MTTF is largely improved.

For train service, it only considers the RUs it can connect to at its position. A failed RU far from where the train is would not impact E2E service delivery for the train. At the overlapping zone, the re-establishment procedure helps the train to resume the connection if one of the RU in the overlapping zone fails. Therefore, the more RUs installed, the less time the train spends in a single RU area, and the more service can be guaranteed by at least two RUs in the overlapping area. Therefore, the train communication service availability increases with the density of RU installation. With more than 20 RUs in the railway, the communication service availability reaches even 11 nines. However, the RU is expensive and difficult to carry maintenance as they are often distributed. With a limited budget, one of the possible solutions could be deploying RUs according to geographical information of the train route and upgrading the existing 3G/4G facility.

If we consider a more frequent passage of trains, there are more chances that at least one train in the entire rail line encounters a network failure during its journey. For one single journey, the availability and reliability are the same. However, when considering all trains running in the line, the service reliability and availability will decrease and even close to (in the case of at least one train in every zone at any moment) network's reliability and availability when the train frequency increases.

Random failures

In the second scenario, we remove the assumption of high reliability on the rest of the network. All elements in gNBs and the CN can fail. Then, the system becomes more complex.

Still, we compare different RU densities alongside the railway. The simulation time is 100,000 hours to generate enough failure in the system.

For the network operators, the system is considered available when all network elements work as initially expected to provide E2E service, the re-establishment request, and the HO request anywhere in the considered railway network. The time to fail is the time from when at least one network element fails to when all the failed network elements are repaired.

For the high-speed train, the service is considered available when its connection is established, and all the UP functions it uses work. HO procedure

[1]If there are 21 RUs along the 100 km line, the distance between two RUs will be 5 km which equals the radius of the RU. Then everywhere along these 100 km rail line will be covered by two RUs.

provides seamless connection as it induces no service interruption and thus enhances service reliability. On the other hand, the re-establishment procedure helps an end-user reconnect to the network from either UP or HO failure. Re-establishment can not maintain a connection and always comes with a service interruption. Therefore, unlike HO, the re-establishment procedure can only enhance service availability but does not contribute to service reliability.

Table 5.14: Network and service performance with random failures in the communication network for railway service. Table from Paper VI [27].

Number of RUs	Network availability	Network MTTF [hour]	Service availability	Service MTTF [hour]
12	99.86058%	55	99.99456%	359
13	99.84895%	52	99.99512%	344
14	99.83789%	50	99.99571%	333
15	99.82612%	48	99.99628%	319
16	99.81485%	46	99.99686%	308
17	99.80219%	44	99.99742%	298
18	99.79151%	42	99.99801%	288
19	99.78031%	41	99.99859%	279
20	99.76875%	39	99.99917%	270

The estimated reliability and availability for the network and service from the simulation are shown in Table 5.14. Similar to the previous scenario, while we increase the number of RUs, the network availability and reliability decrease. However, for communication service, there are more failures during a train's mission, especially minor failures when the number of RUs increases. The re-establishment procedure can guarantee availability since the overlapping area gets larger. Nevertheless, as the number of failures still increases, the MTTF gets shorter, resulting in less reliable communication service. A possible solution for enhancing reliability could be adding redundant items, which may be energy-consuming and expensive for train and network operators.

In order to visualize how different train operating conditions and telecommunication system conditions affect the performance of communication services, an interactive interface has been integrated into this simulation program. This extended simulation program is explained in the Appendix A.

5.3 . Conclusion

This chapter showcased how the proposed model can be applied to assess resilience and the related parameters in vertical use cases.

The Tele-action use case is a general vertical use case, and its latency and availability estimation can be extended to various similar use cases, for example, smart city and remote medication. The UE stays at a fixed point, and it communicates with the Internet or other devices. If the vertical service is critical in the use case, most of the equipment and elements of the internet should be reinforced, and a dedicated network slice can be a good solution to protect the service from outside adverse events.

The railway use case can be generalized to other use cases where users are in mobility, such as autonomous vehicles and other transportation systems. In these use cases, UP E2E service delivery data traffic and CP signaling processes must be considered. The signaling process becomes essential to guarantee the E2E service delivery and thus indirectly impacts the resilience, especially for the users frequently change their position . Availability, reliability, and service continuity can be closely related to how the radio cells are deployed and how the handover process is triggered. The resilience can be improved by studying, case by case, the behavior of the moving user and the schedule of its route planning.

6 - Conclusion

6.1 . 5G network complexity in modeling

6.1.1 . Contribution summary

In Section 2.2, the complexity of the 5G and beyond network is addressed, based on Paper I [22]. A main contribution of Paper I [22] is highlighting that the main feature that differentiates 5G from its predecessors is virtualization, which makes the network more complex and dynamic than before. From an E2E service delivery point of view, the network has a linear logical composition, which can be seen as a set of NFs. From a multi-layer perspective, each NF comprises a virtual layer and a physical infrastructure layer. In addition to this, network management is also an indispensable part of modeling 5G networks. This conclusion is essential for selecting which element should be included and which should not.

Different network modeling and evaluation methods are compared in Section 2.4. In the literature review, many researchers have addressed the issue of network performance assessment. The existing simulation tools have their limitations and inconveniences to adapt to the thesis subject. In this dissertation, a state-space model, Petri Net, is selected for 5G network modeling. Then, it is implemented in a discrete-event simulation platform for network resilience evaluation.

In Paper II [23] and Paper III [24], a Petri Net models a 5G VNF. The network self-healing process is considered. The simulation environment is validated by analytically solving the simple use case. Paper IV [25] and Paper V [26] apply the Petri Net-based discrete-event simulation environment to the traffic variation scenarios. Network auto-scaling and isolation are considered, which can be potentially extended to the Tele-action use case. Paper VI [27] applies the simulation environment to the high mobility use case. Both the UP and CP are considered. Finally, in Paper VII [28], the Markov process models a subsystem of the UP for a railway service. It validates the simulation methods and shows that Markov processes are practical for a small state space. However, when considering the UP and the CP together, the problem will become very complex to solve.

6.1.2 . Perspective

The 5G Radio air interface is not included in the model yet. Indeed, it is possible that the network failure is caused neither by physical nor virtual elements. Instead, the interference or the meteorological condition on the air interface also leads to an interruption in the E2E connection. Plus, Radio air interface can also be an important cause for HO failure in a high-mobility

scenario. This part could be integrated by introducing radio modules, for instance, from libraries in MATLAB.

New models, such as the Multi-agent model, and new approaches in a broad sense, such as the Digital Twin (DT), are new trends for network performance evaluation. It is possible to connect our environment with those platforms so that the simulation environment can not only estimate network resilience by running offline simulations but also gather real-time information and contribute to online decision-making.

6.2 . Resilience threats to 5G verticals

6.2.1 . Contribution summary

Section 2.3 introduces various 5G industrial verticals. 5G is the enabler of realizing verticals' novel services, especially with the arrival of Industry 4.0. The threats and resilience concerns can vary with the selection of vertical use cases. Two main types of threats, internal failure, and external adverse events, have been identified.

Network resilience can be evaluated from two main categories. When looking at frequent system failures, network availability and network reliability will be resilience-related indicators to evaluate the network performance in the long term. For a specific major adverse event, the resilience estimation can be done by estimating the resilience loss before, during, and after the adverse event in the short term.

The choice of scenarios is based on the industrial partners in the chair RRCS [236]. For each scenario, different indicators are investigated:

- Paper II [23] and Paper III [24] propose a single 5G VNF scenario. VNF, an indispensable part of E2E service delivery, is first studied before looking into a large system. In this scenario, only system failure is considered a threat. VNF availability and microservice availability are used as indicator to evaluate the performance.

- In Paper IV [25] and Paper V [26], a complete E2E service delivery is considered. The service suffers from traffic variation, an external threat that can cause severe network congestion and even a propagation of the congestion. E2E service latency and packet loss are estimated. A packet transmission reliability indicator, combining both latency and packet loss, is used as a resilience performance indicator to calculate resilience loss during traffic variation. This scenario can be applied to the Tele-action service of electric networks, where the communication service for Tele-action can suffer from the traffic variation of other service users sharing the same network.

- Paper VI [27] and Paper VII [28] focus on a railway end-users. The system failures are taken into account for the service delivery. Plus, the mobility of end-users requires a more frequent HO process, which can impact the continuity of the communication service. In this scenario, network availability, network reliability, service availability, and service reliability are used for long-term performance evaluation. This scenario can be applied to various railway use cases, for example, autonomous train control.

These works consider internal and external threats and even the propagation of threats. It covers most of the possible scenarios for vertical industries. The short-term performance estimation gives resilience loss information during one specific event. In comparison, long-term performance estimation gives the availability and reliability of the 5G network or the vertical service, which are valuable information in the system design.

6.2.2 . Perspective

As communication technology continues to advance, new industries and scenarios will be introduced. Therefore, new risks should be included in resilience assessment. They can be security and privacy-related risks. The next challenge will be how to take them into consideration and explain them in the proposed model.

The elements considered in the 5G system in the thesis dissertation have a constant failure rate. This assumption holds for virtual elements. However, for some physical elements, such as the servers, a degradation model can be considered to get a more realistic result.

Another potential extension of communication system risk analysis and resilience estimation is considering the impact of degraded or failed 5G networks on actual vertical domain service. A degraded communication network may or may not impact the vertical service. Taking the railway use case as an example, losing the network connection for the train control system will immediately stop the train. However, a few seconds of network loss for passenger entertainment services can be acceptable. This extended study requires a thorough understanding of the use case and deep cooperation with the vertical industries.

6.3 . 5G resilience estimation and optimization

6.3.1 . Contribution Summary

From the aspect of performance evaluation, the scenarios selected in Chapter 4 estimate a group of resilience-related indicators. For random threats, for instance, system failure, 5G resilience is evaluated by availability and reliability for a long enough period of observation time. When focusing on a major

threat, the resilience is evaluated by resilience loss, which is based on a performance indicator. The evolution of this indicator represents how a 5G network or service absorbs, recovers from, and adapts to the threat.

In 5G networks, various methods help to improve resilience.

At the design phase, redundancy has a significant impact on improving availability, as presented in Section 4.2, based on Paper II [23] and Paper III [24]. Besides, in Section 4.4, the use case from Paper IV [25] and Paper V [26] show how network isolation helps reduce the resilience loss due to a major threat, such as traffic change. Finally, network layout or structure can be important factors for availability and reliability, especially for high mobility users, as presented in Section 5.2, based on Paper VI [27] and Paper VII [28].

At the operational phase, two automatic actions that improve resilience are mainly discussed. First, the self-healing mechanism, presented in Section 4.2 by use cases from Paper II [23] and Paper III [24], terminates the failed virtual elements and replaces them by starting and deploying new ones. This action reduces network downtime so that the availability is increased. The other action, the auto-scaling mechanism, presented in Section 4.3 by use cases from Paper IV [25] and Paper V [26], adapts the network to the traffic load. The impact of threats that potentially congest the network can be mitigated, and the performance degradation can be avoided.

In Chapter 5 the resilience for vertical use cases are discussed. The proposed model would need to be modified to adapt to different use cases. The resilience metrics may also be different according to the focus of the use case and the considered threats. The Tele-action use case can benefit from a dedicated network to guarantee its performance when other users may likely to congest the network. For a railway communication use case, the layout and structure of the radio base station impacts the reliability and availability. Nevertheless, in both cases, the energy consumption and deployment cost should also be considered when looking at network resilience.

6.3.2 . Perspective

With the rapid development of machine learning, future telecommunication network design and management can benefit from advice provided by AI. In the design phase, AI can use geographical information, equipment parameters, and service requirements to find the optimal network layout design. In the operational phase, AI can analyze and predict the network situation by collecting real-time information. Then, management actions will be taken to anticipate or mitigate an undesirable situation.

Sustainability is a focus for the future telecommunication network. Resilience assessment should also include the sustainability aspect by considering the impact on energy, society, and economy at both network design and operational phases.

A - Interactive railway use case performance estimation program

A demonstrative program has been developed based on the railway use case described in Section 5.2. Due to various modifiable parameters, the service performance and sensitivity analysis can be complicated to carry out. An interactive program can help, especially for the railway operator, estimate the average or transitory performance during an incident.

A.1 . Introduction

The program is adapted from the Python program for network model and simulation. The major contribution of this program is the railway system and network system coupling. Both network and train behaviors are targeted to be simulated.

The interface of the program is presented in Figure A.1. The program simulates a train mission from Paris to Amiens. This trajectory is approximately 100 km. A train will start from Paris and run at high speed to Amiens. During its mission, the train is targeted to be connected all the time. To achieve this goal, the network facility for the user planes has to be highly available in order to provide E2E connection. At the same time, the train needs to hand over the communication session while it changes its location, which requires a highly available network control plane.

The resilience threats for the service may come from the layout of the network deployment. In fact, by changing how the network, especially for RAN, is organized, the network's redundancy may change. Some places will be overlapped and have a doubled radio access. Some places will have only a single radio unit cover, which is fragile to failure. The system failure of the network can also greatly impact network resilience. The availability and reliability of the components in RAN and CN are also critical to service and network performance.

The architecture diagram of the program is shown in Figure A.2. The program has the following functions:

- Communication network layout parameter configuration: it allows to change how RU, DU, and CU are aggregated. They are linearly distributed along the rail line from Paris to Amiens.

- Communication network component parameter configuration: it allows the change of the failure times of physical equipment and the microser-

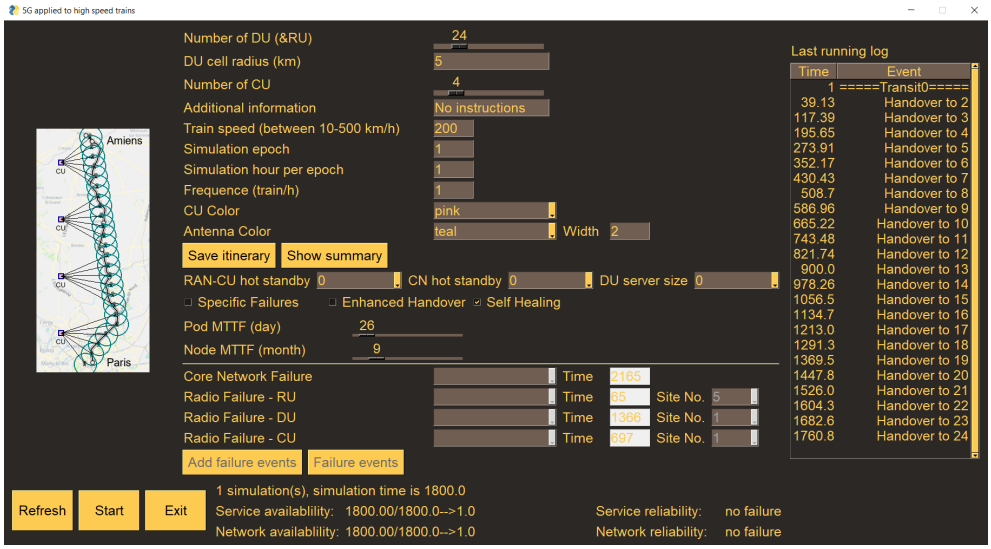


Figure A.1: Interface of the railway use case performance estimation program.

vices. It is also possible to change the redundancy at the microservice level.

- Railway network parameter configuration: it allows the change in the train speed and the train frequency per hour.
- Communication network fault injection: it allows injecting specific predefined network faults into different network components.

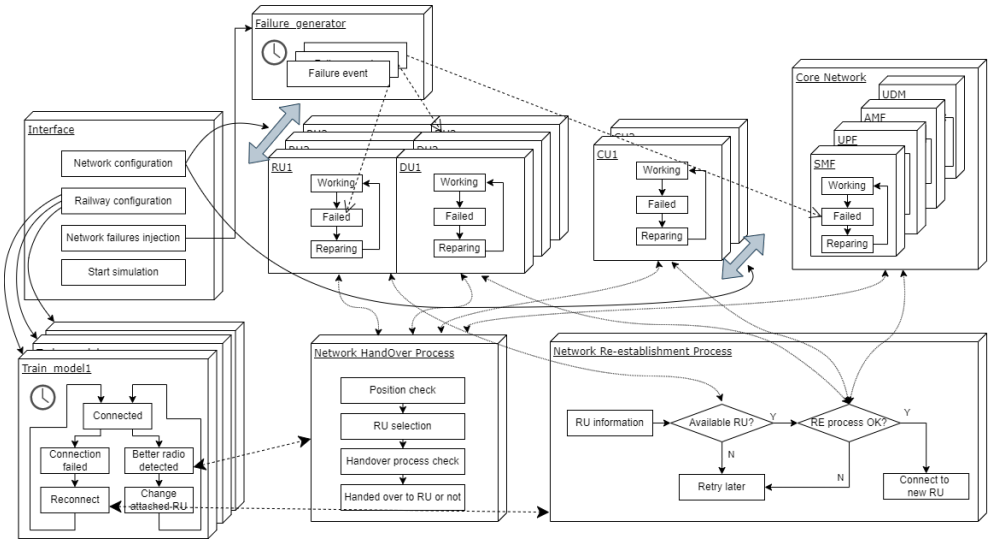


Figure A.2: Architecture diagram of the railway use case performance estimation program.

A.2 . Configure communication and train networks parameters

As shown in Figure A.1, on the top of the interface lays the sector for communication network layout configuration. The RU and DU are placed together to form a cell, meaning that there is always one DU connects to one RU. These RUs or DUs are evenly distributed. Depending on the frequency of the radio spectrum, the effective coverage distance is different. Regardless of how the network structure is set up, the layout must ensure that every location on the railway line is covered by at least one radio cell. When setting up CUs, the number of CUs should be less or equal to the number of DUs. For the VNFs in CN, they are in one centralized site. Some examples of the communication network layout are given in Figure A.3.

The program also provides a more detailed network component configuration. The sites of CN and CUs are relatively large and comprise several physical servers. For the DU, however, it contains by default one server and may possess limited additional servers. The RU is physical equipment and does not have redundancy. For the VNFs, it is also possible to choose the number of standby instances to increase the redundancy of the service. The failure rate for pods and nodes can be adjusted according to the type of software and server. The self-healing process can be enabled with a 5-second intermittent healthiness check.

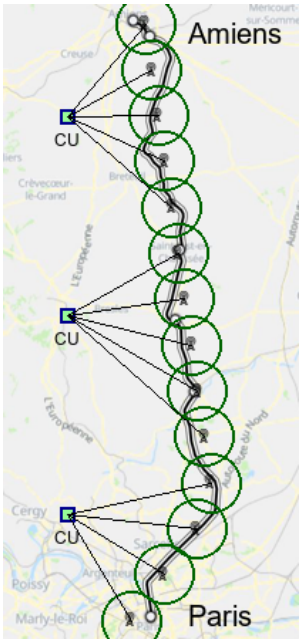
For railway network configuration, the train speed and frequency are modifiable. The train speed can vary from 10 to 500 km/h. Thus, the time the train spends on the rail line will differ. The frequency of the train may impact the time the train passes a radio cell.

A.3 . Average performance estimation

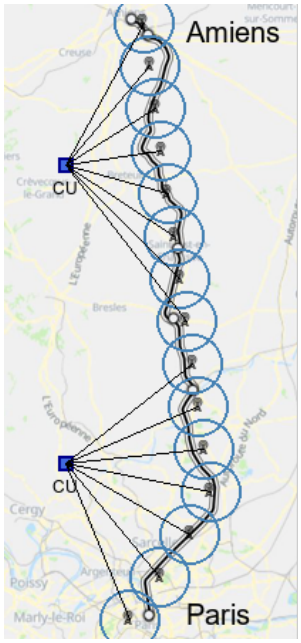
To estimate the impact of system failure on the communication network and train service, the Monte Carlo method will be applied. It will generate large samples and approximate the performance using the mean value.

In the result example in Figure A.4, the simulation has been run 1000 times. The simulation will run each round for one train journey of 1800 s. With the default network configuration, the train communication service will achieve an availability of 99.99656465% and generate four times of failure, which is equivalent to an MTBF of 450 000 s. The 5G network will achieve an availability of 99.93423122% and generate 15 times of failure, which is equivalent to an MTBF of 120 000 s.

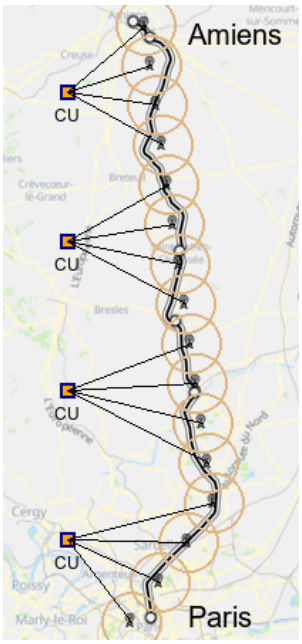
A.4 . Performance estimation by injecting failures



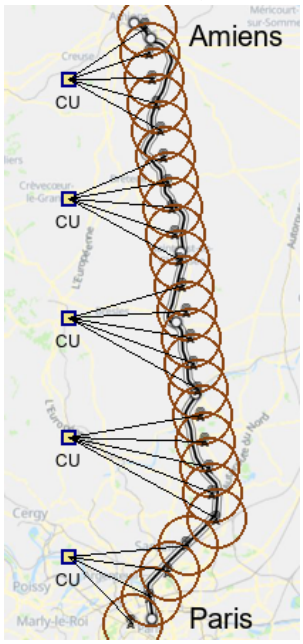
(a) Network layout 1



(b) Network layout 2



(c) Network layout 3



(d) Network layout 4

Figure A.3: Four communication network layouts. (a) Layout 1: 14 RUs + 14 DUs + 3 CUs. (b) Layout 2: 15 RUs + 15 DUs + 2 CUs. (c) Layout 3: 16 RUs + 16 DUs + 4 CUs. (d)Layout 4: 24 RUs + 24 DUs + 5 CUs.

By adding specific failure events, the program can also inject predefined defaults into the network to evaluate the performance under a certain sce-

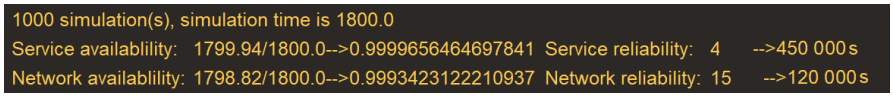


Figure A.4: Computation result example of the railway use case performance estimation program.

nario. For example, in Figure A.5, various failures are injected into the model. These failures will be triggered during the simulation.

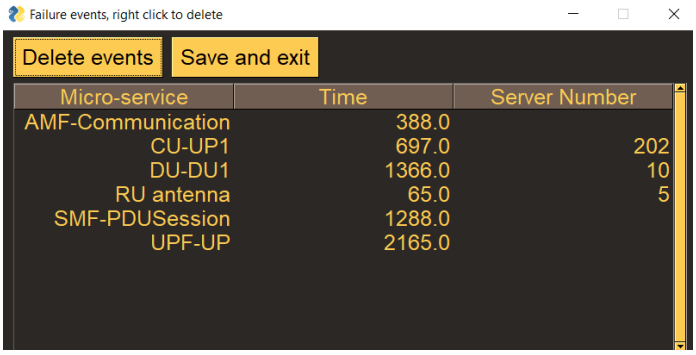


Figure A.5: Failure events summary window in the railway use case performance estimation program.

When the simulation is launched only once, we can trace the log for both the communication network and railway service. Figure A.6 summarizes the event log during a train service. The handover processes are all recorded in the log. The DU #12 failure happened at 1366 s, which led to a connection failure. The train re-established the connection using another DU #11 that is reachable. When the DU #12 is self-healed, the train could then hand over the session back to it. The connection loss time, in this case, is 40 ms for re-establishing the connection.

A.5 . Conclusion

This work aims to provide an interactive simulation platform. Before implementing a wireless communication network for railway service, this simulation program gives a practical solution to estimate network and service performance by manipulating the deployment of the communication network and other parameters. It also showcases how network and service would react and adapt to a given failure, which could give insights into failure anticipation.

This program will be extended in the future to simulate multiple train lines simultaneously. Plus, it can also be combined with the train schedules. This allows the network to be designed and operated more energy-efficiently and flexibly, depending on train operations and schedules.

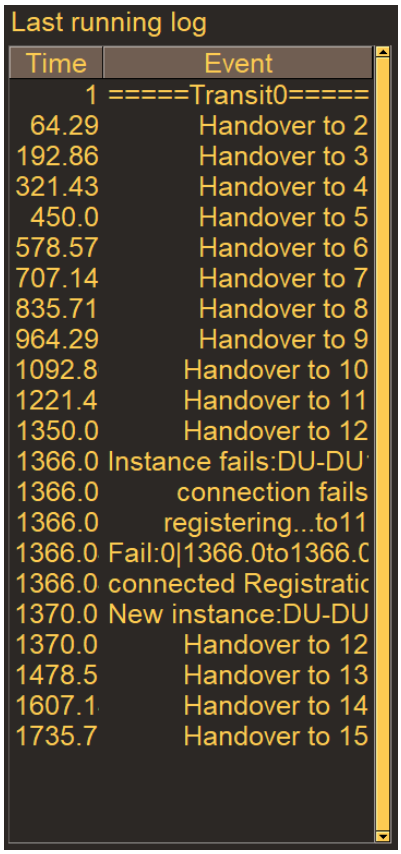


Figure A.6: The train service event log of the railway use case performance estimation program.

B - Reinforcement-based auto-scaling

B.1 . Introduction

As previously discussed and investigated in Section 4.3, as part of MANO functions, auto-scaling is critical for the correct operation of NFV-based networks. Different scaling strategies have been compared in Section 4.3. However, the traffic pattern can sometimes be irregular and hard to predict, making it difficult to adapt the network to the traffic. Plus, the concern about energy efficiency becomes essential for creating an environmentally friendly and sustainable network. Therefore, we start to find a balance between resource cost and fulfilling the performance requirement of the services. It is hard for traditional strategies, such as threshold-based, to take up these challenges.

Some researchers have focused on applying Machine Learning methods to solve the scaling problem. They mainly proposed strategies focused on predictive scaling, exploiting the historical data by using time-series forecasting [237]. Among these Machine Learning methods, Reinforcement Learning (RL) has recently attracted attention. It seems to provide potential solutions to address network management and orchestration challenges in 5G and beyond networks [238].

B.1.1 . Reinforcement learning

Reinforcement Learning is a machine learning paradigm where an agent learns to interact with an environment, making a sequence of decisions to maximize a numerical reward signal. It is distinguished from other computational approaches since it emphasizes learning by an agent from direct interaction with the environment, without requiring exemplary supervision or complete models of the environment [239]. This learn-from-interaction mechanism can be applied to network management for the auto-scaling problem.

B.1.2 . Deep Q-learning

We can apply a value-based method for RL in the auto-scaling problem by estimating the action-value function, known as the Q-function, which assigns a value to each action taken in a particular state. Traditional RL methods, such as Q-Learning, are effective but limited in their applicability when dealing with high-dimensional state spaces. In the network scaling problem, many network indicators can be taken into account when deciding the scaling action so the state space can be huge. Deep Q-learning (DQN) emerged as a groundbreaking innovation in RL by introducing deep neural networks as function approximators to estimate the Q-function. The core idea behind DQN is to use a deep neural network to approximate the Q-function. This neu-

ral network takes the environment's state as input and outputs the estimated Q-values for all possible actions.

B.2 . Implementation DQN model

We use the OpenAI Gym toolkit [240] to simulate a one-VNF network system. We build a network gym environment that behaves as a real VNF containing one single microservice.

The network environment we considered is a class with three essential functions. The first function will initialize a class and set the initial state of our RL problem. The second function is the step function, which takes an action and returns the system state, the reward, a sign of the end of the running episode. Finally, the reset function resets the state of the environment.

The state for this one-VNF network is a three-dimensional vector $\{U, T, N\}$, U is the current CPU resource usage, T is the current traffic arrival rate, N is the current number of available pods. The packet arrival rate is imposed from outside the environment. The number of available pods is modified by actions taken at each step. The CPU usage changes with the number of pods and the traffic arrival rate.

The action space is discretized into five values:

- (1) Scaling in by removing two pods;
- (2) Scaling in by removing one pod;
- (3) No change;
- (4) Scaling out by adding one pod;
- (5) Scaling out by adding two pods.

An agent will interact with the environment by making an action and sending it to the step function of the environment. After each step, it will receive a reward. The agent will act as the Kubernetes autoscaler, and it takes a decision every 5 seconds.

An episode terminates if (1) the agent removes all the current pods, i.e., there is no pod in the environment, or (2) the environment has more than 20 pods, i.e., the resource is overused, or (3) the episode length is greater than a given time (1200 s during training).

The agent will receive a reward at each step based on its action. This reward comprises three parts: the latency penalty, the rejection penalty, and the resource usage reward. The latency penalty is 50 times the average packet waiting time between the current step and the next step in the VNF queue due to the congestion. The rejection penalty is calculated as the total rejected packets between two steps due to congestion divided by 1000. The resource

usage reward is -2 times the difference between the current and expected (to keep the CPU usage at the desired level: 60%) number of pods if the average CPU usage between two steps is below 30%. It takes the value of -1 if the average CPU usage is between 30% and 40%. It takes the value of $+5$ if the average CPU usage is between 40% and 80%. It takes the value of -1 if the average CPU usage is between 80% and 99%. It takes the value of -2 if the average CPU usage is above 99%.

To solve the problem, a DQN agent is created. It first uses the Bellman Equation to form a Q-function $Q_t(x, a)$ to quantify the expected discounted future rewards for each possible action a for a given state x and a given step t. This agent then uses a Boltzmann exploration strategy to explore new options at each step to get a greater reward. This BoltzmannQpolicy is intended for a discrete action space. It uses a soft-max function to convert the Q values of each possible action into a distribution as follows:

$$Pr_x(A_t = a) = \frac{e^{Q_t(x,a)}}{\sum_{b=1}^{k} e^{Q_t(x,b)}} \tag{B.1}$$

In this way, the DQN agent selects an action based on the probability generated by soft-maxing Q values. Figure B.1 gives an overview of the system we consider in this work.

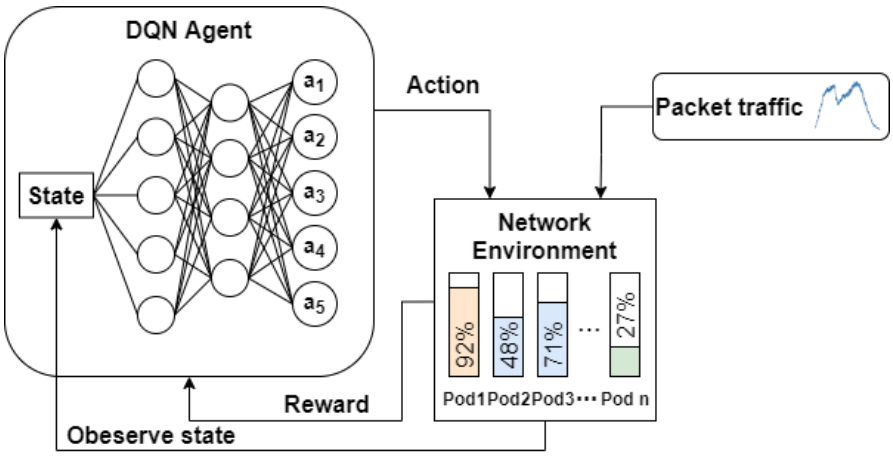


Figure B.1: Network auto-scaling reinforcement learning scheme.

The performance of the DQN agent will be compared with three other agents: Random actions agent, No action agent, and Threshold-based agent.

For the Random actions agent, at each step, it will randomly select one of the five actions. For No action agent, the pod number will not change. The threshold-based agent will take action based on Algorithm 1 and is not limited to a maximum of two pods that can be scaled up or down at a step.

B.3 . Experiments and results

During the training, the network will receive packets with a random traffic arrival rate varying between 400 and 4500 packets per second. The traffic arrival changes every second by randomly multiplying by a value in $[0.75, 1.25[$. The maximum training episode is 1200 steps.

The RL model is trained using the Google Colab platform. For the DQN architecture, we build a three-layer network with 821 parameters. The discounted factor is 0.99 by default. The learning rate for training is set to $5e^{-4}$. The model has been trained for $1e^{5}$ steps to achieve a satisfying reward.

The trained model is applied to test with different traffic patterns, similar to those in Subsection 4.3.3. The test episode will only last for 60 seconds this time.

B.3.1 . Traffic 1: Long variation

Traffic 1 packet arrival rate is shown in Figure B.2. The traffic arrival rate increases suddenly from 18 s. The CPU usage and pod number results of the four agents are shown in Figure B.3 and Figure B.4.

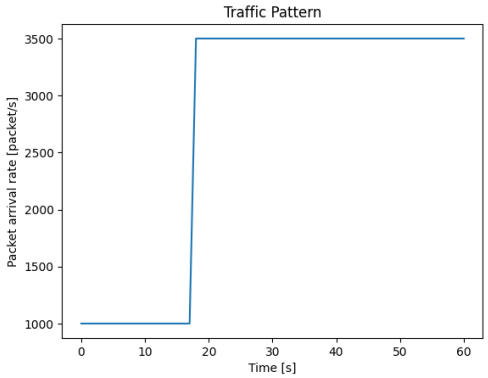


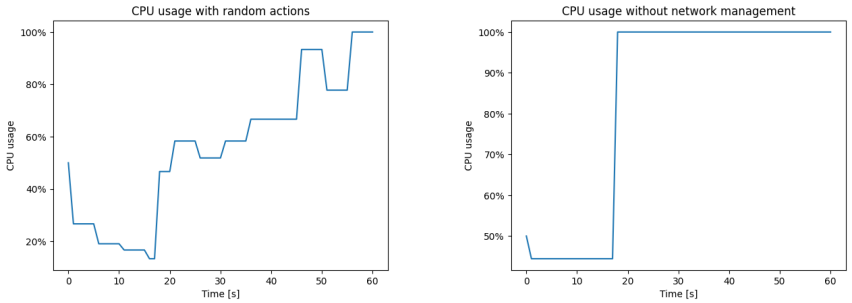
Figure B.2: Packet arrival rate of traffic pattern 1: long traffic variation.

When sampling the random strategy agent, we found too many scaling-out actions in the beginning, and the resources were over-allocated even before the traffic increase arrived. In the end, the scaling-in actions resulted in the pod being overused.

For the no action agent, the network suffered from an overloaded situation with CPU usage of 100% from 18 s until the end of the episode.

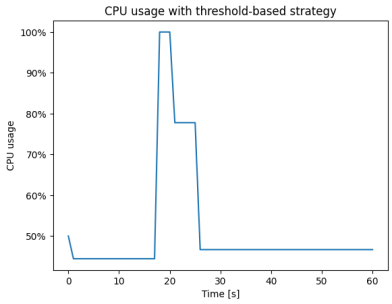
The threshold-based agent made immediate and efficient scaling-outs after the traffic increase. The first scaling-out action added three pods and reduced CPU usage from 100% to about 80%. The second scaling-out action added four pods and reduced CPU usage from 80% to about 45%.

For the RL DQN agent, as the scaling action is limited to 2 pods at a time, it can not act like the threshold-based agent to scale out more than two pods

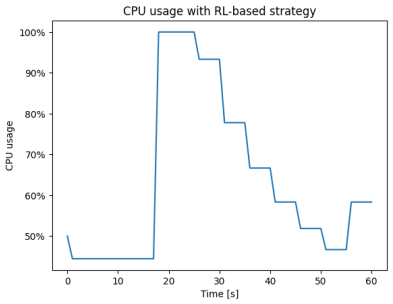


(a) Random actions

(b) No action



(c) Threshold-based strategy



(d) RL strategy

Figure B.3: CPU utilization rates performance of different agents for Traffic 1. (a) Random actions (b) No action (c) Threshold-based strategy (d) RL strategy.

at one step. It behaved more carefully when taking action in the experiment and added only one pod when noticing the traffic increase. Therefore, CPU usage drops slowly to a desired level.

B.3.2 . Traffic 2: Short variation

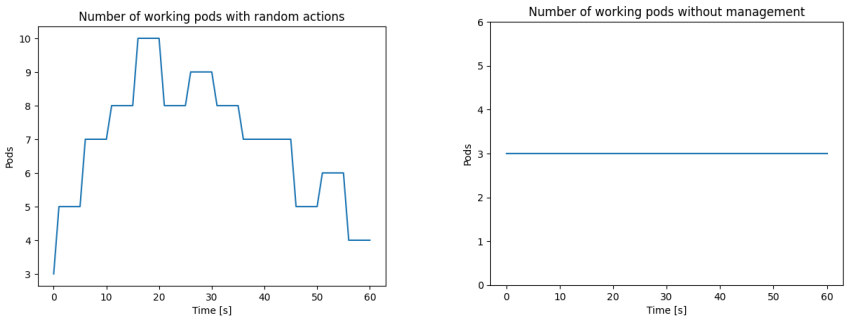
Traffic 2 packet arrival rate is shown in Figure B.5. The traffic arrival rate increases from 18 s to 23 s. The CPU usage and pod number results of the four agents are shown in Figure B.6 and Figure B.7.

When sampling the random strategy agent, it initially keeps a very small quantity of pods, which does not allow the network to cope with the sudden traffic increase between 18 s and 23 s. The CPU usage was 100%. In the end, too many scaling-out actions resulted in pod resources being over-allocated.

For the no action agent, the network suffered from an overloaded situation with CPU usage of 100% during the whole traffic peak time.

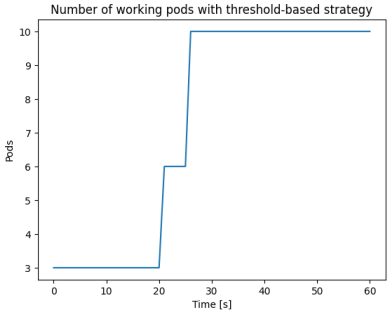
The threshold-based agent scaled out three more pods right after the traffic increase. The CPU usage reduced from 100% to about 80%. Then, it noticed the traffic decrease and took a scaling-in action to remove the over-allocated resource.

For the RL DQN agent, it only scaled out one pod after traffic increased. It was not enough to mitigate the congestion. Then, as the traffic decreased to

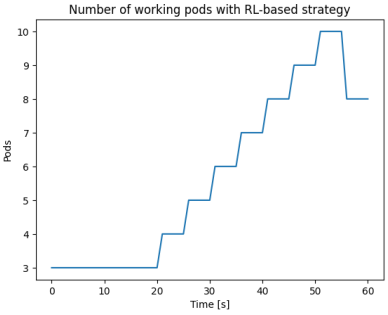


(a) Random actions

(b) No action



(c) Threshold-based strategy



(d) RL strategy

Figure B.4: Pod usage of different agents for Traffic 1. (a) Random actions (b) No action (c) Threshold-based strategy (d) RL strategy.

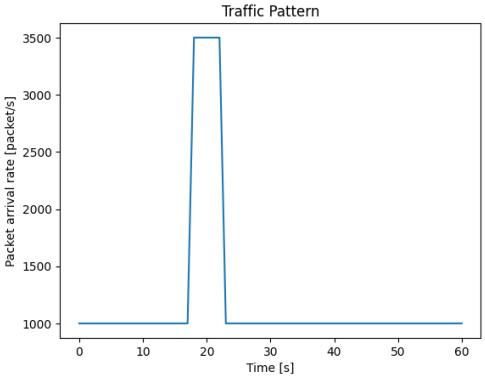


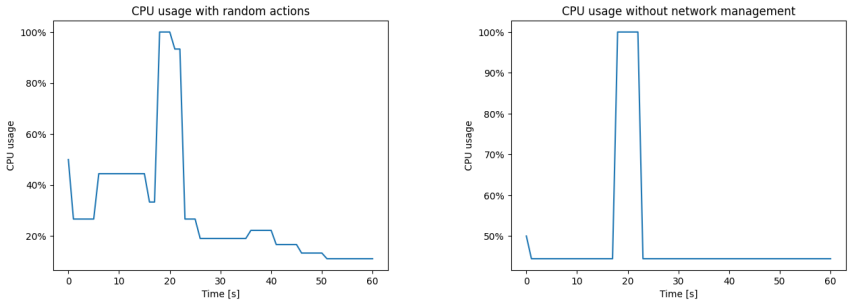
Figure B.5: Packet arrival rate of traffic pattern 2: short traffic variation.

normal, a scaling-in action was taken to reduce the allocated resources.

B.3.3 . Traffic 3: Sinusoidal variation 1

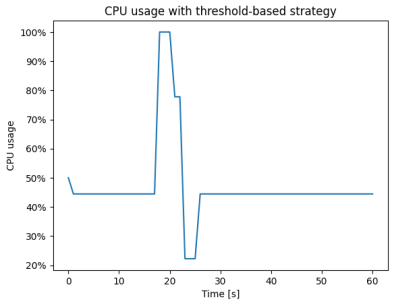
Traffic 3 packet arrival rate is shown in Figure B.8. The traffic arrival rate has a sinusoidal pattern. The CPU usage and pod number results of the four agents are shown in Figure B.9 and Figure B.10.

At this time, when sampling the random strategy agent, it tends to scale

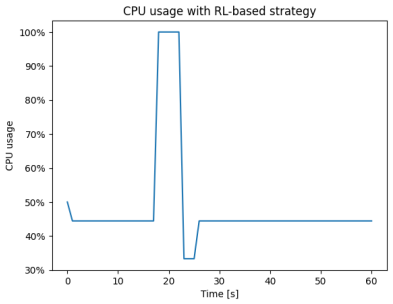


(a) Random actions

(b) No action



(c) Threshold-based strategy



(d) RL strategy

Figure B.6: CPU utilization rates performance of different agents for Traffic 2. (a) Random actions (b) No action (c) Threshold-based strategy (d) RL strategy.

out the number of pods gradually. Therefore, the network was more likely to get congested when the traffic increased in the beginning with CPU usage is 100%. In the second half of the episode, as there were more pods in the network, the CPU usage was not too high, even during peak traffic. However, the pod resource was over-allocated during traffic non-peak time.

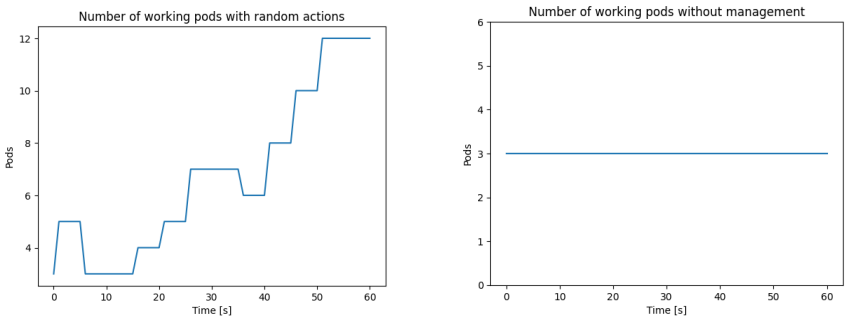
For the no action agent, the network suffered from an overloaded situation during all traffic peaks of the episode.

The threshold-based agent tried to follow the traffic change trends. However, it could not predict the traffic change in the future. The CPU usage sometimes reached 100% during traffic peaks. The actions were not perfectly adapted to the sinusoidal traffic variation.

For the RL DQN agent, the actions seemed consistent with traffic trends. The traffic pattern has a peak at 25 s and a peak at 45 s. The network managed by RL DQN agent had the most pods during two intervals from 25 s to 30 s and 45 s to 50 s. The high CPU usage appeared less frequent during the episode.

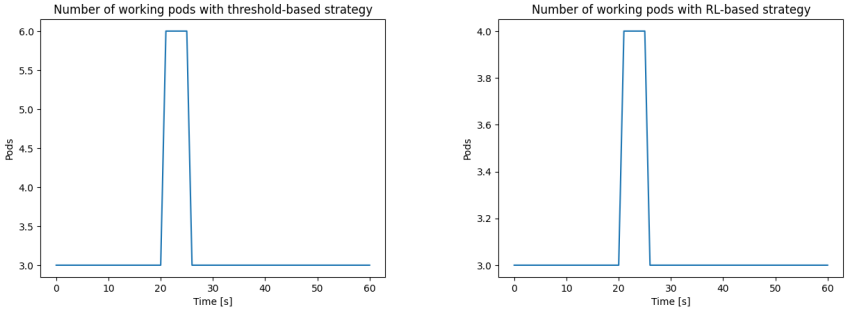
B.3.4 . Traffic 4: Sinusoidal variation 2

Traffic 4 packet arrival rate is shown in Figure B.11. The traffic arrival rate has a sinusoidal pattern with a larger approximate entropy than Traffic 3. The four agents' CPU usage and pod number results are shown in Figure B.12 and



(a) Random actions

(b) No action



(c) Threshold-based strategy

(d) RL strategy

Figure B.7: Pod usage of different agents for Traffic 2. (a) Random actions (b) No action (c) Threshold-based strategy (d) RL strategy.

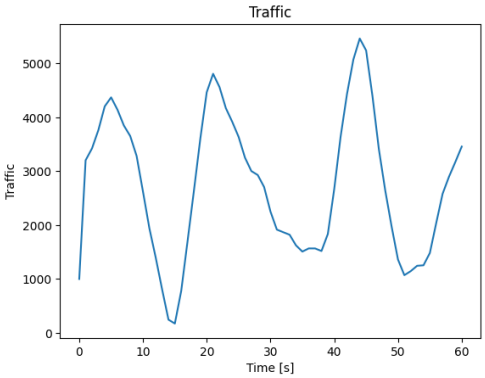
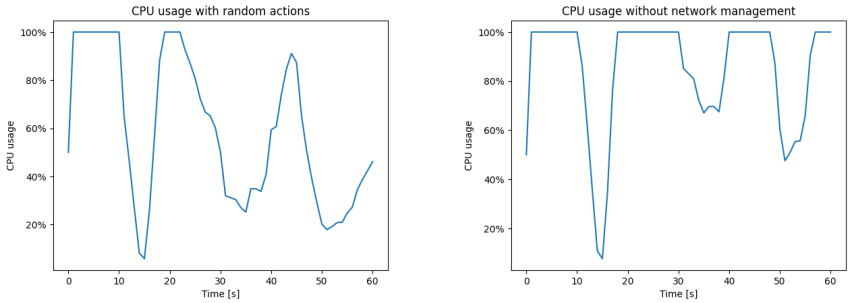


Figure B.8: Packet arrival rate of traffic pattern 3: a sinusoidal traffic variation.

Figure B.13.

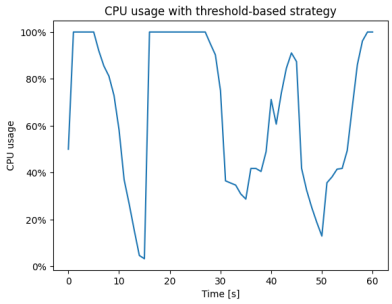
The random strategy agent took many scaling-out actions at the beginning and kept a large number of pods until the end. Some overload situations are only observed during peak times in the beginning.

Similar to the case in Traffic 3, the network managed by the no action agent suffered from an overloaded situation during all traffic peaks of the episode.

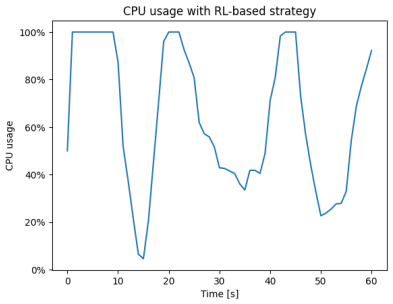


(a) Random actions

(b) No action



(c) Threshold-based strategy



(d) RL strategy

Figure B.9: CPU utilization rates performance of different agents for Traffic 3. (a) Random actions (b) No action (c) Threshold-based strategy (d) RL strategy.

The threshold-based agent could not finely adjust the number of pods to the traffic. Even though scaling actions were taken very often, the CPU usage amplitude was large and changed quite frequently.

The RL DQN agent also failed to follow the traffic pattern. Indeed, some continuous scaling-in and scaling-out actions can be observed. It did not stabilize the CPU utilization rate. CPU usage varied from 10% to almost 100%.

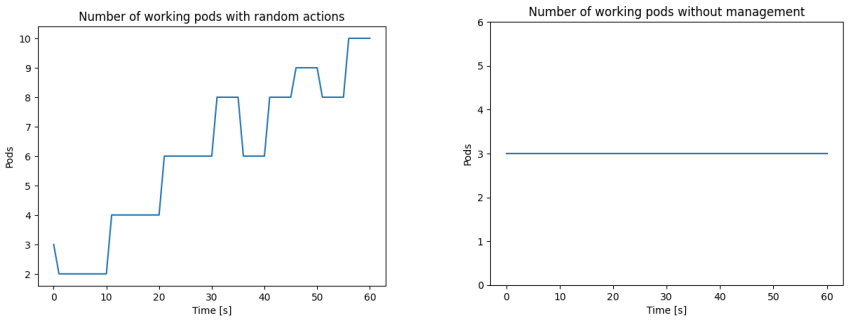
B.3.5 . Result analysis

The performance of the four agents is summarized in Table B.1.

The Threshold-based strategy agent had the lowest reject packet for Traffic 1 and 2. For sinusoidal traffic patterns, Traffic 3 and 4, the RL DQN agent obtained better results.

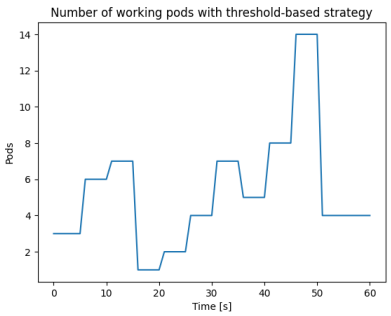
For the packet waiting time in the queue, it is the Threshold-based strategy agent that performed better for Traffic 1, 2, and 4. For Traffic 3, the RL DQN agent had the least packet waiting delay.

Pod resource allocation becomes an important indicator as it can reflect strategy energy efficiency. Since this aspect is not considered for the Random actions and No action agents, only the Threshold-based strategy and the RL DQN agents are compared. The RL DQN agent allocated the least pods to the network in Traffic 1 and 4. For Traffic 2 and 3, there is little difference in the

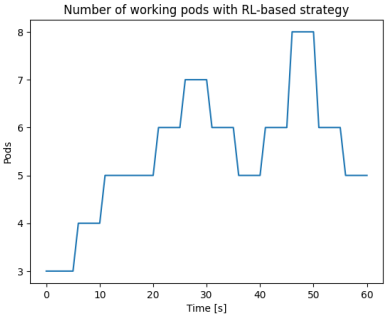


(a) Random actions

(b) No action



(c) Threshold-based strategy



(d) RL strategy

Figure B.10: Pod usage of different agents for Traffic 3. (a) Random actions (b) No action (c) Threshold-based strategy (d) RL strategy.

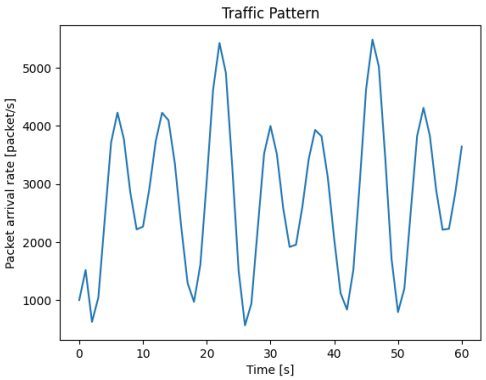
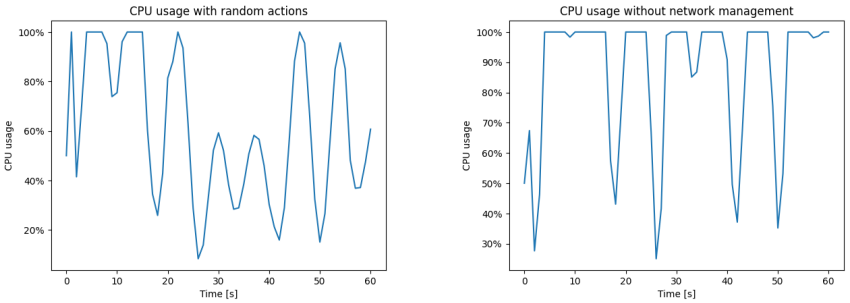


Figure B.11: Packet arrival rate of traffic pattern 4: a sinusoidal traffic variation.

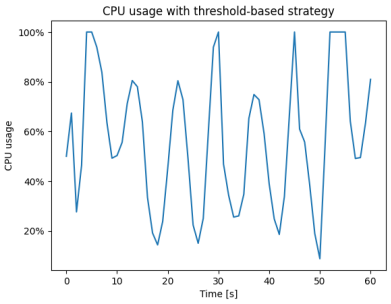
allocation of pod resources for the two agents.

In terms of resilience loss, again, it is the Threshold-based strategy agent that performed better for Traffic 1, 2, and 4. The RL DQN agent had the least resilience loss for Traffic 3. In general, the difference in resilience loss between the Threshold-based strategy agent and the RL DQN agent was not significant.

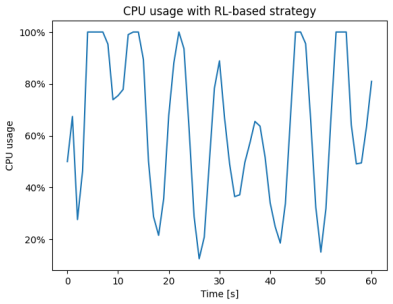


(a) Random actions

(b) No action



(c) Threshold-based strategy



(d) RL strategy

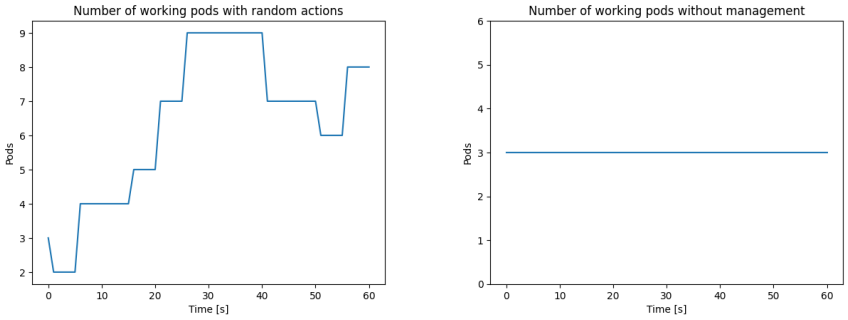
Figure B.12: CPU utilization rates performance of different agents for Traffic 4. (a) Random actions (b) No action (c) Threshold-based strategy (d) RL strategy.

B.4 . conclusion

This part of the work explores the RL method for the auto-scaling problem. Since the threshold-based strategy is not satisfying when the entropy of the traffic pattern is large, the RL agent seems to be one of the solutions to improve network performance and resilience during traffic variation.

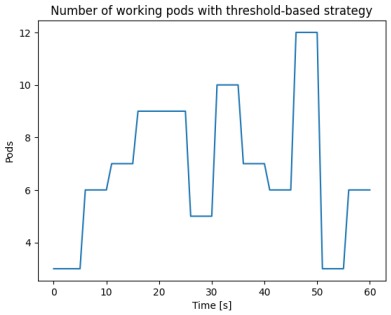
As an exploration project, only DQN is applied to solve the problem. The dimensions of state space and action space are limited. The result already shows that for some traffic patterns, RL DQN agent performs better than the threshold-based strategy and has a comparable performance for other traffic patterns. Many other RL algorithms may potentially train an agent with even better performance.

As further work, the RL agent can be trained and tested in a real telecommunication network environment. It can be integrated into the Kubernetes platform, which can provide the network state to the agent and manage the network according to the scaling orders from the agent.

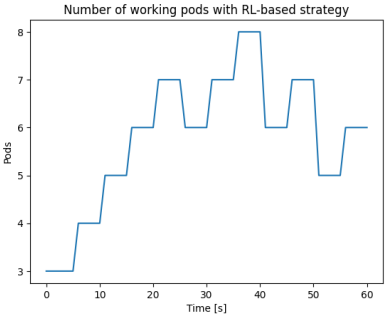


(a) Random actions

(b) No action



(c) Threshold-based strategy



(d) RL strategy

Figure B.13: Pod usage of different agents for Traffic 4. (a) Random actions (b) No action (c) Threshold-based strategy (d) RL strategy.

Table B.1: Agent performance comparison.

Performance	Traffic	Random	No Act.	Threshold	RL
Total rejected packet	1	30237	53750	**3750**	6245
	2	18730	6250	**3750**	4748
	3	16894	50538	31744	**14017**
	4	18580	51600	7457	**5696**
Total packet waiting delay [s]	1	4785.52	6568.51	**460.24**	1111.05
	2	4915.19	763.85	**459.12**	677.67
	3	2346.39	5667.71	4763.62	**2015.46**
	4	2616.83	5918.61	**1090.57**	1324.98
Pod resource allocation [pod · s]	1	215	180	440	345
	2	170	180	195	185
	3	385	180	325	330
	4	305	180	415	350
Resilience loss [s]	1	30.0212	35.6356	**2.4862**	7.6292
	2	23.2354	4.1437	**2.4862**	3.9133
	3	12.1758	29.3453	17.2358	**11.8155**
	4	15.3064	32.5931	**6.6404**	8.9052

Bibliography

[1] NGMN-Alliance. 5g white paper. Technical report, Next Generation Mobile Networks, February 2015.

[2] 5G-ACIA. Key 5g use cases and requirements—white paper. Technical report, 5G Alliance for Connected Industries and Automation, May 2020.

[3] 3GPP-TS22.104. Service requirements for cyber-physical control applications in vertical domains. Technical Specification (TS) 22.104, 3rd Generation Partnership Project (3GPP), September 2021. version 17.7.0.

[4] 3GPP-TS22.261. Service requirements for the 5G system. Technical Specification (TS) 22.261, 3rd Generation Partnership Project (3GPP), September 2022. version 17.11.0.

[5] 3GPP-TS22.280. Mission Critical Services Common Requirements (MCCoRe); Stage 1. Technical Specification (TS) 22.280, 3rd Generation Partnership Project (3GPP), September 2021. version 17.7.0.

[6] 3GPP-TS22.289. Mobile communication system for railways. Technical Specification (TS) 22.289, 3rd Generation Partnership Project (3GPP), December 2019. version 17.0.0.

[7] European Telecommunications Standards Institute ETSI. Technologies-5g. https://www.etsi.org/technologies/5G/, 2023. [Online; accessed 03 July 2023].

[8] David V Rosowsky. Defining resilience. *Sustainable and Resilient Infrastructure*, 5(3):125–130, 2020.

[9] Antoine Clément, François Marmier, Daouda Kamissoko, Didier Gourc, Liên Wioland, Virginie Govaere, and Julien Cegarra. Robustesse, résilience: une brève synthèse des définitions au travers d'une analyse structurée de la littérature. In *MOSIM'18-12ème Conférence internationale de Modélisation, Optimisation et SIMulation*, page 8, 2018.

[10] NGMN-Alliance. 5g end-to-end architecture framework. Technical report, Next Generation Mobile Networks, August 2019.

[11] ITU-R. Imt vision-framework and overall objectives of the future development of imt for2020 and beyond: Recommendation itu-r m.2083-0 [r]. https://www.itu.int/rec/R-REC-M.2083, 2015. [Online; accessed 10 August 2023].

[12] NGMN-Alliance. 5g whitepaper 2. Technical report, Next Generation Mobile Networks, 7 2020.

[13] Charalampos Sergiou, Marios Lestas, Pavlos Antoniou, Christos Liaskos, and Andreas Pitsillides. Complex systems: A communication networks perspective towards 6g. *IEEE Access*, 8:89007–89030, 2020.

[14] Seyedmohsen Hosseini, Kash Barker, and Jose E Ramirez-Marquez. A review of definitions and measures of system resilience. *Reliability Engineering & System Safety*, 145:47–61, 2016.

[15] Yi-Ping Fang, Nicola Pedroni, and Enrico Zio. Resilience-based component importance measures for critical infrastructure network systems. *IEEE Transactions on Reliability*, 65(2):502–512, 2016.

[16] P Stevenin, Micheline Guiserix, Gina Chiquillo, Jonathan Brown, Gwenaël Barbaud, and Hedi Ben Amor. La modélisation des systèmes complexes, un concept novateur pour la gestion des actifs des réseaux électriques. In *Congrès Lambda Mu 20 "Maîtriser les risques dans un monde en mouvement"*, 10 2016.

[17] Yi-Ping Fang and Enrico Zio. An adaptive robust framework for the optimization of the resilience of interdependent infrastructures under natural hazards. *European Journal of Operational Research*, 276(3):1119–1136, 2019.

[18] Wenjuan Sun, Paolo Bocchini, and Brian D Davison. Resilience metrics and measurement methods for transportation infrastructure: The state of the art. *Sustainable and Resilient Infrastructure*, 5(3):168–199, 2020.

[19] Fanlin Meng, Guangtao Fu, Raziyeh Farmani, Chris Sweetapple, and David Butler. Topological attributes of network resilience: A study in water distribution systems. *Water research*, 143:376–386, 2018.

[20] Massimo Condoluci and Toktam Mahmoodi. Softwarization and virtualization in 5g mobile networks: Benefits, trends and challenges. *Computer Networks*, 146:65–84, 2018.

[21] Ghada Arfaoui, Jose Manuel Sanchez Vilchez, and Jean-Philippe Wary. Security and resilience in 5g: Current challenges and future directions. In *2017 IEEE Trustcom/BigDataSE/ICESS*, pages 1010–1015. IEEE, 2017.

[22] Rui Li, Bertrand Decocq, Anne Barros, Yiping Fang, and Zhiguo Zeng. Complexity in 5G Network Applications and use cases. In *31st European Safety and Reliability Conference*, pages 3054–3061, Angers, France, September 2021. Research Publishing Services.

[23] Rui Li, Bertr Decocq, Anne Barros, Yiping Fang, and Zhiguo Zeng. Petri Net-Based Model for 5G and Beyond Networks Resilience Evaluation. In *2022 25th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pages 131–135, Paris, France, March 2022.

[24] Rui Li, Bertrand Decocq, Anne Barros, Yiping Fang, and Zhiguo Zeng. Modélisation d'un réseau 5G par des réseaux de Pétri pour estimer sa résilience. In *AlgoTel 2022 - 24èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, pages 1–4, Saint-Rémy-Lès-Chevreuse, France, May 2022.

[25] Rui Li, Bertrand Decocq, Yiping Fang, Zhiguo Zeng, and Anne Barros. A Petri Net-based model to study the impact of traffic changes on 5G network resilience. In *32nd European Safety and Reliability Conference (ESREL 2022)*, pages 3016–3023, Dublin, Ireland, August 2022. Research Publishing.

[26] Rui Li, Bertrand Decocq, Anne Barros, Yi-Ping Fang, and Zhiguo Zeng. Estimating 5g network service resilience against short timescale traffic variation. *IEEE Transactions on Network and Service Management*, 20(3):2230–2243, 2023.

[27] Rui Li, Bertrand Decocq, Anne Barros, Yiping Fang, and Zhiguo Zeng. Reliability challenges of 5g and beyond networks applications in high-speed trains. In *33rd European Safety and Reliability Conference (ESREL 2023)*, pages 1935–1942, Southampton, UK, September 2023.

[28] Rui Li, Bertrand Decocq, Yiping Fang, Zhiguo Zeng, and Anne Barros. High-mobility 5g communication service: availability and reliability analysis. In *2023 7th International Conference on System Reliability and Safety (ICSRS)*, November 2023.

[29] P. S. Khodashenas, J. Aznar, A. Legarrea, C. Ruiz, M. S. Siddiqui, E. Escalona, and S. Figuerola. 5g network challenges and realization insights. In *2016 18th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4, 2016.

[30] International Telecommunication Union. Setting the scene for 5g: Opportunities & challenges. https://www.itu.int/pub/D-PREF-BB.5G_01-2018, 2018. [Online; accessed 10 August 2023].

[31] Wind River. vran: The next step in network transformation. https://events.windriver.com/wrcd01/wrcm/2017/10/vRAN-The-Next-Step-in-Network-Transformation-White-Paper.pdf, 2017. [Online; accessed 03 July 2023].

[32] IEEE. 802.11 ad-2012–ieee standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements-part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 3: Enhancements for very high throughput in the 60 ghz band. *Amendment to IEEE Std*, 2012.

[33] Natale Patriciello, Sandra Lagen, Biljana Bojović, and Lorenza Giupponi. Nr-u and ieee 802.11 technologies coexistence in unlicensed mmwave spectrum: Models and evaluation. *IEEE access*, 8:71254–71271, 2020.

[34] 3GPP-TS 23.501. 5g;system architecture for the 5g system (5gs). Technical specification (ts), 3rd Generation Partnership Project (3GPP), July 2023. version 17.9.0.

[35] Docker Inc. What is a container? https://www.docker.com/resources/what-container/, 2023. [Online; accessed 10 October 2023].

[36] The European Union Agency for Cybersecurity (ENISA). Enisa threat landscape for 5g networks – updated threat assessment for the fifth generation of mobile telecommunications networks (5g). Technical report, ENISA, December 2020.

[37] ETSI. Network functions virtualisation (nfv); virtual network functions architecture. https://www.etsi.org/deliver/etsi_gs/NFV-SWA/001_099/001/01.01.01_60/gs_NFV-SWA001v010101p.pdf, December 2014. [Online; accessed 03 July 2023].

[38] ETSI. Network functions virtualisation (nfv);management and orchestration. https://www.etsi.org/deliver/etsi_gs/nfv-man/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf, December 2014. [Online; accessed 03 July 2023].

[39] David Breitgand, Vadim Eisenberg, Nir Naaman, Nir Rozenbaum, and Avi Weit. Toward true cloud native nfv mano. In *2021 12th International Conference on Network of the Future (NoF)*, pages 1–5, 2021.

[40] Maciej Gawel and Krzysztof Zielinski. Analysis and evaluation of kubernetes based nfv management and orchestration. In *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, pages 511–513, 2019.

[41] Leila Abdollahi Vayghan, Mohamed Aymen Saied, Maria Toeroe, and Ferhat Khendek. Deploying microservice based applications with kubernetes: Experiments and lessons learned. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 970–973, 2018.

[42] Amazon Web Services. Etsi nfvo compliant orchestration in the kubernetes/cloud native world. Whitepaper, Amazon, October 2022.

[43] Jangwon Lee and Younghan Kim. A design of mano system for cloud native infrastructure. In *2021 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 1336–1339, 2021.

[44] George Margetis, Barbara Valera-Muros, Konstantinos C. Apostolakis, Almudena Díaz Zayas, Laura Panizo, Pedro Tomás, Luis Cordeiro, Joao Henriques, and Constantine Stephanidis. Validation of nfv management and orchestration on kubernetes-based 5g testbed environment. In *2022 IEEE Globecom Workshops (GC Wkshps)*, pages 844–849, 2022.

[45] Abderaouf Khichane, Ilhem Fajjari, Nadjib Aitsaadi, and Mourad Gueroui. Cloud native 5g: an efficient orchestration of cloud native 5g system. In *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*, pages 1–9, 2022.

[46] National Research Council et al. Disaster resilience: A national imperative. Technical report, National Academies, Washington, D.C., 2012.

[47] European Parliament and The Council of the European Union. Directive (eu) 2022/2557 of the european parliament and of the council of 14 december 2022 on the resilience of critical entities and repealing council directive 2008/114/ec. Document, European Parliament, 2022.

[48] ResiliNets group. Resilinets wiki. https://resilinets.org/. [Online; accessed 03 July 2023].

[49] Hendrik Knoche and H De Meer. Quantitative qos-mapping: A unifying approach. In *Building QoS into Distributed Systems: IFIP TC6 WG6. 1 Fifth International Workshop on Quality of Service (IWQOS'97), 21–23 May 1997, New York, USA*, pages 345–356. Springer, 1997.

[50] Salah Eddine Elayoubi, Mikael Fallgren, Panagiotis Spapis, Gerd Zimmermann, David Martín-Sacristán, Changqing Yang, Sébastien Jeux, Patrick Agyapong, Luis Campoy, Yinan Qi, and Shubhranshu Singh. 5g service requirements and operational use cases: Analysis and metis ii vision. In *2016 European Conference on Networks and Communications (EuCNC)*, pages 158–162, 2016.

[51] NGMN Alliance. Perspectives on vertical industries and implications for 5g. *White Paper, Jun*, 2016.

[52] SGCC, China Telecom, and Huawei. Powered by sa: Smart grid 5g network slicing. https://www.gsma.com/

futurenetworks/wp-content/uploads/2020/03/2_Powered-by-SA_Smart-Grid-5G-Network-Slicing_China-Telecom_GSMA_v2.0.pdf, March 2020. [Online; accessed 03 July 2023].

[53] International Union of Railways. Future railway mobile communication system user requirements specification. https://uic.org/IMG/pdf/frmcs_user_requirements_specification-fu_7100-v5.1_0.pdf, February 2023. [Online; accessed 03 July 2023].

[54] Slawomir Kukliński and Lechosław Tomaszewski. Key performance indicators for 5g network slicing. In *2019 IEEE Conference on network softwarization (NetSoft)*, pages 464–471. IEEE, 2019.

[55] Apostolos Malatras, Georgia Bafoutsou, Edgars Taurins, and Marnix Dekker. Telecom Security Incidents 2021. Annual Report, European Union Agency for Cybersecurity (ENISA), July 2022.

[56] Christian Esposito, Antonios Gouglidis, David Hutchison, Andrei Gurtov, Bjarne Emil Helvik, Poul Einar Heegaard, Gianluca Rizzo, and Jacek Rak. On the disaster resiliency within the context of 5g networks: The recodis experience. In *2018 European Conference on Networks and Communications (EuCNC)*. Institute of Electrical and Electronics Engineers (IEEE), 2018.

[57] Jaya Preethi Mohan, Niroop Sugunaraj, and Prakash Ranganathan. Cyber security threats for 5g networks. In *2022 IEEE international conference on electro information technology (eIT)*, pages 446–454. IEEE, 2022.

[58] Jorge Navarro-Ortiz, Pablo Romero-Diaz, Sandra Sendra, Pablo Ameigeiras, Juan J. Ramos-Munoz, and Juan M. Lopez-Soler. A survey on 5g usage scenarios and traffic models. *IEEE Communications Surveys & Tutorials*, 22(2):905–929, 2020.

[59] Phudit Ampririt, Ermioni Qafzezi, Kevin Bylykbashi, Makoto Ikeda, Keita Matsuo, and Leonard Barolli. Application of fuzzy logic for slice qos in 5g networks: a comparison study of two fuzzy-based schemes for admission control. *International Journal of Mobile Computing and Multimedia Communications (IJMCMC)*, 12(2):18–35, 2021.

[60] Pingzhi Fan, Jing Zhao, and I Chih-Lin. 5g high mobility wireless communications: Challenges and solutions. *China Communications*, 13(2):1–13, 2016.

[61] Gosan Noh, Bing Hui, and Ilgyu Kim. High speed train communications in 5g: Design elements to mitigate the impact of very high mobility. *IEEE Wireless Communications*, 27(6):98–106, 2020.

[62] Taous Madi, Hyame Assem Alameddine, Makan Pourzandi, and Amine Boukhtouta. Nfv security survey in 5g networks: A three-dimensional threat taxonomy. *Computer Networks*, 197:108288, 2021.

[63] Montida Pattaranantakul, Ruan He, Qipeng Song, Zonghua Zhang, and Ahmed Meddahi. Nfv security survey: From use case driven threat analysis to state-of-the-art countermeasures. *IEEE Communications Surveys & Tutorials*, 20(4):3330–3368, 2018.

[64] Ijaz Ahmad, Tanesh Kumar, Madhusanka Liyanage, Jude Okwuibe, Mika Ylianttila, and Andrei Gurtov. Overview of 5g security challenges and solutions. *IEEE Communications Standards Magazine*, 2(1):36–43, 2018.

[65] Shane Fonyi. Overview of 5g security and vulnerabilities. *The Cyber Defense Review*, 5(1):117–134, 2020.

[66] Liudong Xing. Cascading failures in internet of things: Review and perspectives on reliability and resilience. *IEEE Internet of Things Journal*, 8(1):44–64, 2021.

[67] Carlos Colman-Meixner, Chris Develder, Massimo Tornatore, and Biswanath Mukherjee. A survey on resiliency techniques in cloud computing infrastructures and applications. *IEEE Communications Surveys & Tutorials*, 18(3):2244–2281, 2016.

[68] Michel Bruneau, Stephanie E Chang, Ronald T Eguchi, George C Lee, Thomas D O'Rourke, Andrei M Reinhorn, Masanobu Shinozuka, Kathleen Tierney, William A Wallace, and Detlof Von Winterfeldt. A framework to quantitatively assess and enhance the seismic resilience of communities. *Earthquake spectra*, 19(4):733–752, 2003.

[69] Mirza Golam Kibria, Kien Nguyen, Gabriel Porto Villardi, Ou Zhao, Kentaro Ishizu, and Fumihide Kojima. Big data analytics, machine learning, and artificial intelligence in next-generation wireless networks. *IEEE Access*, 6:32328–32338, 2018.

[70] Charalampos Sergiou, Marios Lestas, Pavlos Antoniou, Christos Liaskos, and Andreas Pitsillides. Complex systems: A communication networks perspective towards 6g. *IEEE Access*, 8:89007–89030, 2020.

[71] Teresa Gomes, János Tapolcai, Christian Esposito, David Hutchison, Fernando Kuipers, Jacek Rak, Amaro de Sousa, Athanasios Iossifides, Rui Travanca, João André, Luísa Jorge, Lúcia Martins, Patricia Ortiz Ugalde, Alija Pašić, Dimitrios Pezaros, Simon Jouet, Stefano Secci, and Massimo Tornatore. A survey of strategies for communication networks to protect against large-scale natural disasters. In *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pages 11–22, 2016.

[72] Egemen K. Çetinkaya, Dan Broyles, Amit Dandekar, Sripriya Srinivasan, and James P. G. Sterbenz. Modelling communication network challenges for Future Internet resilience, survivability, and disruption tolerance: a simulation-based approach. *Telecommunication Systems*, 52(2):751–766, February 2013.

[73] James Sterbenz, Egemen Cetinkaya, Mahmood Hameed, Abdul Jabbar, Shi Qian, and Justin Rohrer. Evaluation of network resilience, survivability, and disruption tolerance: Analysis, topology generation, simulation, and experimentation: Invited paper. *Springer Telecommunication Systems*, 12 2011.

[74] James P. G. Sterbenz, David Hutchison, Egemen K. Çetinkaya, Abdul Jabbar, Justin P. Rohrer, Marcus Schöller, and Paul Smith. Redundancy, diversity, and connectivity to achieve multilevel network resilience, survivability, and disruption tolerance invited paper. *Telecommunication Systems*, 56(1):17–31, 2014.

[75] David Tipper. Resilient network design: challenges and future directions. *Telecommunication Systems*, 56(1):5–16, 2014.

[76] Karina Gomez, Leonardo Goratti, Tinku Rasheed, and Laurent Reynaud. Enabling disaster-resilient 4g mobile communication networks. *IEEE Communications Magazine*, 52(12):66–73, 2014.

[77] Ying Wang, Jing Xu, and Lisi Jiang. Challenges of system-level simulations and performance evaluation for 5g wireless networks. *IEEE Access*, 2:1553–1561, 2014.

[78] Rahul Ghosh, Francesco Longo, Flavio Frattini, Stefano Russo, and Kishor S. Trivedi. Scalable analytics for iaas cloud availability. *IEEE Transactions on Cloud Computing*, 2(1):57–70, 2014.

[79] Mario Di Mauro, Giovanni Galatro, Maurizio Longo, Fabio Postiglione, and Marco Tambasco. Comparative performability assessment of sfcs: The case of containerized ip multimedia subsystem. *IEEE Transactions on Network and Service Management*, 18(1):258–272, 2021.

[80] nsnam. ns-3,a discrete-event network simulator for internet systems. https://www.nsnam.org/. [Online; accessed 07 October 2023].

[81] Katerina Koutlia, Biljana Bojovic, Zoraze Ali, and Sandra Lagén. Calibration of the 5g-lena system level simulator in 3gpp reference scenarios. *Simulation Modelling Practice and Theory*, 119:102580, 2022.

[82] Marco Mezzavilla, Sourjya Dutta, Menglei Zhang, Mustafa Riza Akdeniz, and Sundeep Rangan. 5g mmwave module for the ns-3 network simulator. In *Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '15, page 283–290, New York, NY, USA, 2015. Association for Computing Machinery.

[83] Ana Larrañaga, M Carmen Lucas-Estañ, Sandra Lagén, Zoraze Ali, Imanol Martinez, and Javier Gozalvez. An open-source implementation and validation of 5g nr configured grant for urllc in ns-3 5g lena: A scheduling case study in industry 4.0 scenarios. *Journal of Network and Computer Applications*, 215:103638, 2023.

[84] Katerina Koutlia, Sandra Lagén, and Biljana Bojovic. Enabling qos provisioning support for delay-critical traffic and multi-flow handling in ns-3 5g-lena. In *Proceedings of the 2023 Workshop on ns-3*, pages 45–51, 2023.

[85] Matteo Drago, Tommaso Zugno, Michele Polese, Marco Giordani, and Michele Zorzi. Millicar: An ns-3 module for mmwave nr v2x networks. In *Proceedings of the 2020 Workshop on Ns-3*, WNS3 '20, page 9–16, New York, NY, USA, 2020. Association for Computing Machinery.

[86] OpenSim. OMNeT++ technical articles. https://docs.omnetpp.org/. [Online; accessed 07 October 2023].

[87] Kristina Josifović, Stefan Boljević, Vukan Ninković, and Natan Turćinović. Simulating massive iot environmental monitoring scenario using omnet++. In *2019 27th Telecommunications Forum (TELFOR)*, pages 1–4. IEEE, 2019.

[88] Anupama Hegde and Andreas Festag. Artery-c: An omnet++ based discrete event simulation framework for cellular v2x. In *Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 47–51, 2020.

[89] Pablo Barbecho Bautista, Luis F Urquiza-Aguiar, Mónica Aguilar Igartua, Diego Javier Reinoso-Chisaguano, and Martha Cecilia Paredes Paredes. An evaluation of omnet++-based v2x communication frameworks: On the path towards 5g-v2x simulations. In *Proceedings of the 24th International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 75–78, 2021.

[90] Marija Gajić, Marcin Bosk, Susanna Schwarzmann, Stanislav Lange, and Thomas Zinner. Demonstrating qoe-aware 5g network slicing emulated with htb in omnet++. In *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–2. IEEE, 2022.

[91] Giovanni Nardini, Dario Sabella, Giovanni Stea, Purvi Thakkar, and Antonio Virdis. Simu5g–an omnet++ library for end-to-end performance evaluation of 5g networks. *IEEE Access*, 8:181176–181191, 2020.

[92] Antonio Virdis, Giovanni Nardini, Giovanni Stea, and Dario Sabella. End-to-end performance evaluation of mec deployments in 5g scenarios. *Journal of Sensor and Actuator Networks*, 9(4):57, 2020.

[93] Suryanarayanaraju Pusapati, Bassant Selim, Yimin Nie, Huang Lin, and Wei Peng. Simulation of nr-v2x in a 5g environment using omnet++. In *2022 IEEE Future Networks World Forum (FNWF)*, pages 634–638. IEEE, 2022.

[94] Jia Jun Tham. Simulation of 5g-based vehicle network using omnet++ and sumo. *EEE Student Reports (FYP/IA/PA/PI)*, 2023.

[95] TETCOS. Network simulator, netsim, emulator, 5g, military communication, vehicular networks. https://www.tetcos.com/index.html. [Online; accessed 07 October 2023].

[96] Arshad Iqbal and Wei Chen. An integrated distribution grid protection system using 5g urllc. In *2022 IEEE 5th International Electrical and Energy Conference (CIEEC)*, pages 3901–3908. IEEE, 2022.

[97] Richard Wiencek, Sagnika Ghosh, Maria C Laurent-Rice, Eden Black, Taheerah Mujahid, Sameer Kalyan, Monika Priya, Ayush Goyal, Cesar Luna, and Kanwalinderjit Kaur. Simulation of the 5g communication link between solar micro-inverters and scada system. In *2023 5th Global Power, Energy and Communication Conference (GPECOM)*, pages 311–316. IEEE, 2023.

[98] Sabira Khanam Shorna. Performance analysis of 5g ddos attack using machine learning. master's thesis, The University of Memphis, Jul 2021.

[99] Riverbed Technology. Riverbed modeler - discrete event simulator for network simulation. https://www.riverbed.com/products/riverbed-modeler/. [Online; accessed 07 October 2023].

[100] Tibor Petrov, Milan Dado, Karl Ernst Ambrosch, and Peter Holečko. Experimental topology for v2v communication based on internet of things. In *2016 ELEKTRO*, pages 72–76. IEEE, 2016.

[101] Zifan Zhou, Ying Yan, Sarah Ruepp, and Michael Berger. Analysis and implementation of packet preemption for time sensitive networks. In *2017 IEEE 18th international conference on high performance switching and routing (HPSR)*, pages 1–6. IEEE, 2017.

[102] Faruk Aktas, Celal Ceken, and Yunus Emre Erdemli. Iot-based healthcare framework for biomedical applications. *Journal of Medical and Biological Engineering*, 38:966–979, 2018.

[103] The MathWorks. 5g wireless technology development. https://www.mathworks.com/solutions/wireless-communications/5g.html. [Online; accessed 07 October 2023].

[104] The MathWorks. 5g toolbox. https://www.mathworks.com/products/5g.html. [Online; accessed 07 October 2023].

[105] Martin Klaus Müller, Fjolla Ademaj, Thomas Dittrich, Agnes Fastenbauer, Blanca Ramos Elbal, Armand Nabavi, Lukas Nagel, Stefan Schwarz, and Markus Rupp. Flexible multi-node simulation of cellular mobile communications: the vienna 5g system level simulator. *EURASIP Journal on Wireless Communications and Networking*, 2018(1):227, Sep 2018.

[106] Zeyu Huang, José Rodríguez-Piñeiro, Tomás Domínguez-Bolaño, Xuefeng Yin, Juyul Lee, and David Matolak. Performance of 5g terrestrial network deployments for serving uav communications. In *2020 14th European Conference on Antennas and Propagation (EuCAP)*, pages 1–5, 2020.

[107] Enrique Caballero, Aymen Fakhreddine, and Christian Bettstetter. Interference by drones to 5g ground users: A simulation study. In *Proceedings of the Ninth Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, DroNet '23, page 45–50, New York, NY, USA, 2023. Association for Computing Machinery.

[108] Mehdi Ashury, Jan Nausner, and Christoph F. Mecklenbräuker. 5g-positioning for traffic safety and intelligent intersections. In *2023 17th European Conference on Antennas and Propagation (EuCAP)*, pages 1–5, 2023.

[109] Florian Kaltenberger, Aloizio P Silva, Abhimanyu Gosain, Luhan Wang, and Tien-Thinh Nguyen. Openairinterface: Democratizing innovation in the 5g era. *Computer Networks*, 176:107284, 2020.

[110] Navid Nikaein, Mahesh K. Marina, Saravana Manickam, Alex Dawson, Raymond Knopp, and Christian Bonnet. Openairinterface: A flexible platform for 5g research. *SIGCOMM Comput. Commun. Rev.*, 44(5):33–38, oct 2014.

[111] Salvatore Costanzo, Ilhem Fajjari, Nadjib Aitsaadi, and Rami Langar. Dynamic network slicing for 5g iot and embb services: A new design with prototype and implementation results. In *2018 3rd Cloudification of the Internet of Things (CIoT)*, pages 1–7, 2018.

[112] Julio Manco, Guillermo Gallud Baños, Jérôme Härri, and Miguel Sepulcre. Prototyping v2x applications in large-scale scenarios using openairinterface. In *2020 IEEE Vehicular Networking Conference (VNC)*, pages 1–4, 2020.

[113] Rodolphe Bertolini and Mickael Maman. Evaluating handover performance for end-to-end lte networks with openairinterface. In *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, pages 1–5, 2021.

[114] Sukchan Lee. Open5gs. https://open5gs.org/. [Online; accessed 07 October 2023].

[115] George Amponis, Panagiotis Radoglou-Grammatikis, Thomas Lagkas, Savvas Ouzounidis, Maria Zevgara, Ioannis Moscholios, Sotirios Goudos, and Panagiotis Sarigiannidis. Towards securing next-generation networks: Attacking 5g core/ran testbed. In *2022 Panhellenic Conference on Electronics & Telecommunications (PACET)*, pages 1–4. IEEE, 2022.

[116] Porrama Rakchaithanakorn, Jitti Nitjaphanich, Robithoh Annur, Sushank Chaudhary, Amir Parnianifard, Lunchakorn Wuttisittikulkij, Pruk Sasithong, Pisit Vanichchanunt, Ittipon Yamyuan, Sukritta Paripurana, et al. Simulative investigation of real time video conferencing network by incorporating 5g technology. In *The 14th Regional Conference on Electrical and Electronics Engineering (RC-EEE 2021)*, page 85, 2022.

[117] Sergio Barrachina-Muñoz, Miquel Payaró, and Josep Mangues-Bafalluy. Cloud-native 5g experimental platform with over-the-air transmissions and end-to-end monitoring. In *2022 13th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, pages 692–697. IEEE, 2022.

[118] NYCU Communication Service/Software Laboratory. free5gc. https://free5gc.org. [Online; accessed 07 October 2023].

[119] Chia-Wei Liao, Fuchun Joseph Lin, and Yoichi Sato. Evaluating nfv-enabled network slicing for 5g core. In *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 401–404. IEEE, 2020.

[120] Yu-Herng Chai and Fuchun Joseph Lin. Evaluating dedicated slices of different configurations in 5g core. *Journal of Computer and Communications*, 9(7):55–72, 2021.

[121] Yi-Sung Chiu, Li-Hsing Yen, Tse-Han Wang, and Chien-Chao Tseng. A cloud native management and orchestration framework for 5g end-to-end network slicing. In *2022 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pages 69–76. IEEE, 2022.

[122] Yi Liu, Qiaoling Li, Qingping Cao, Zhilan Huang, Yangchun Li, and Yongbing Fan. Evaluation of free5gc forwarding performance on private and public clouds. In *2022 IEEE Cloud Summit*, pages 9–16. IEEE, 2022.

[123] Irin Dorathy and M. Chandrasekaran. Simulation tools for mobile ad hoc networks: a survey. *Journal of Applied Research and Technology*, 16(5), Jun 2018.

[124] Ahmad Musa and Irfan Awan. Functional and performance analysis of discrete event network simulation tools. *Simulation Modelling Practice and Theory*, 116:102470, 2022.

[125] Christos Bouras, Apostolos Gkamas, Georgios Diles, and Zacharopoulos Andreas. A comparative study of 4g and 5g network simulators. *International Journal on Advances in Networks and Services*, 13(1), 2020.

[126] German Dario Castellanos Tache. *Wireless network design for 5G networks and beyond*. PhD thesis, Ghent University, 2023.

[127] Panagiotis K. Gkonis, Panagiotis T. Trakadas, and Dimitra I. Kaklamani. A comprehensive study on simulation techniques for 5g networks: State of the art results, analysis, and future challenges. *Electronics*, 9(3), 2020.

[128] John B Bowles. A survey of reliability-prediction procedures for microelectronic devices. *IEEE Transactions on Reliability*, 41(1):2–12, 1992.

[129] Tuan Anh Nguyen, Dugki Min, Eunmi Choi, and Thang Duc Tran. Reliability and availability evaluation for cloud data center networks using hierarchical models. *IEEE Access*, 7:9273–9313, 2019.

[130] RF Forche. Analysis of reliability block diagrams with multiple blocks per component. In *Annual Proceedings on Reliability and Maintainability Symposium*, pages 145–148. IEEE, 1990.

[131] Soraya Sinche, Oswaldo Polo, Duarte Raposo, Marcelo Femandes, Fernando Boavida, André Rodrigues, Vasco Pereira, and Jorge Sá Silva. Assessing redundancy models for iot reliability. In *2018 IEEE 19th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pages 14–15. IEEE, 2018.

[132] Wan-Chi Chang and Pi-Chung Wang. Write-aware replica placement for cloud computing. *IEEE Journal on Selected Areas in Communications*, 37(3):656–667, 2019.

[133] Guto Leoni Santos, Theo Lynn, Judith Kelner, and Patricia Takako Endo. Availability-aware and energy-aware dynamic sfc placement using reinforcement learning. *The Journal of Supercomputing*, pages 1–30, 2021.

[134] Victor Netes. Ultra-reliable communications: Basic concepts, challenges and open issues. In *2023 Systems of Signals Generating and Processing in the Field of on Board Communications*, pages 1–6. IEEE, 2023.

[135] Wen-Shing Lee, Doris L Grosh, Frank A Tillman, and Chang H Lie. Fault tree analysis, methods, and applications: a review. *IEEE transactions on reliability*, 34(3):194–203, 1985.

[136] Qiong Li and Ruiying Li. Reliability evaluation for cloud computing system considering common cause failure. In *2016 35th Chinese Control Conference (CCC)*, pages 5267–5271. IEEE, 2016.

[137] Alexandru Butoi and Gheorghe Cosmin Silaghi. Fault-tree-based service availability model in cloud environments: A failure trace archive approach. In *Economics of Grids, Clouds, Systems, and Services: 13th International Conference, GECON 2016, Athens, Greece, September 20-22, 2016, Revised Selected Papers 13*, pages 74–86. Springer, 2017.

[138] Huahong Zhu, Jianzhao Li, Jianyuan Hu, and Wenyun Li. Failure-aware and automated disaster backup in the 5g core network. In *2022 International Communication Engineering and Cloud Computing Conference (CECCC)*, pages 48–53. IEEE, 2022.

[139] Jeonghoon Mo. *Markov Chain Modeling*, page 13–31. Synthesis Lectures on Learning, Networks, and Algorithms. Springer International Publishing, Cham, 2010.

[140] L. Xing and A. Shrestha. Qos reliability of hierarchical clustered wireless sensor networks. In *2006 IEEE International Performance Computing and Communications Conference*, pages 6 pp.–646, 2006.

[141] Hasan Farooq, Md. Salik Parwez, and Ali Imran. Continuous time markov chain based reliability analysis for future cellular networks. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2015.

[142] Tarik Taleb, Adlen Ksentini, and Bruno Sericola. On service resilience in cloud-native 5g mobile systems. *IEEE Journal on Selected Areas in Communications*, 34(3):483–496, 2016.

[143] Haoran Zhu, Jing Bai, Xiaolin Chang, Jelena Mišić, Vojislav Mišić, and Yang Yang. Stochastic model-based quantitative analysis of edge upf service dependability. In *Algorithms and Architectures for Parallel Processing: 20th International Conference, ICA3PP 2020, New York City, NY, USA, October 2–4, 2020, Proceedings, Part II 20*, pages 619–632. Springer, 2020.

[144] Mario Di Mauro, Maurizio Longo, and Fabio Postiglione. Availability evaluation of multi-tenant service function chaining infrastructures by multi-dimensional universal generating function. *IEEE Transactions on Services Computing*, 14(5):1320–1332, 2021.

[145] Jing Bai, Xiaolin Chang, Fumio Machida, Zhen Han, Yang Xu, and Kishor S Trivedi. Quantitative understanding serial-parallel hybrid sfc services: a dependability perspective. *Peer-to-Peer Networking and Applications*, 15(4):1923–1938, 2022.

[146] Karsten Wolf. Generating petri net state spaces. In Jetty Kleijn and Alex Yakovlev, editors, *Petri Nets and Other Models of Concurrency – ICATPN 2007*, pages 29–42, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[147] Dong Seong Kim, Jin B Hong, Tuan Anh Nguyen, Fumio Machida, Jong Sou Park, and Kishor S Trivedi. Availability modeling and analysis of a virtualized system using stochastic reward nets. In *2016 IEEE International Conference on Computer and Information Technology (CIT)*, pages 210–218. IEEE, 2016.

[148] Lanlan Rui, Xushan Chen, Zhipeng Gao, Wenjing Li, Xuesong Qiu, and Luoming Meng. Petri net-based reliability assessment and migration optimization strategy of sfc. *IEEE Transactions on Network and Service Management*, 18(1):167–181, 2020.

[149] Besmir Tola, Yuming Jiang, and Bjarne E Helvik. On the resilience of the nfv-mano: An availability model of a cloud-native architecture. In *2020 16th International Conference on the Design of Reliable Communication Networks DRCN 2020*, pages 1–7. IEEE, 2020.

[150] Haifeng Song and Eckehard Schnieder. Availability and performance analysis of train-to-train data communication system. *IEEE Transactions on Intelligent Transportation Systems*, 20(7):2786–2795, 2019.

[151] Juxing Zhu, Ning Huang, Junliang Wang, and Xiaopeng Qin. Availability model for data center networks with dynamic migration and multiple traffic flows. *IEEE Transactions on Network and Service Management*, 2023.

[152] Dong Seong Kim, Fumio Machida, and Kishor S Trivedi. Availability modeling and analysis of a virtualized system. In *2009 15th IEEE Pacific Rim International Symposium on Dependable Computing*, pages 365–371. IEEE, 2009.

[153] Mario Di Mauro, Giovanni Galatro, Maurizio Longo, Fabio Postiglione, M Tambasco, M Cepin, and R Bris. Availability evaluation of a virtualized ip multimedia subsystem for 5g network architectures. *Safety and Reliability-Theory and Applications*, pages 2203–2210, 2017.

[154] Stefano Sebastio, Rahul Ghosh, and Tridib Mukherjee. An availability analysis approach for deployment configurations of containers. *IEEE Transactions on Services Computing*, 14(1):16–29, 2018.

[155] Thilina Pathirana and Gianfranco Nencioni. Availability model of a 5g-mec system. *arXiv preprint arXiv:2304.09992*, 2023.

[156] Gabriel Otero Pérez, José Alberto Hernández, and David Larrabeiti López. Delay analysis of fronthaul traffic in 5g transport networks. In *2017 IEEE 17th International Conference on Ubiquitous Wireless Broadband (ICUWB)*, pages 1–5, 2017.

[157] S. Bjϕmstad, D. Chen, and R. Veisllari. Handling delay in 5g ethernet mobile fronthaul networks. In *2018 European Conference on Networks and Communications (EuCNC)*, pages 1–9, 2018.

[158] Luca Cominardi, Luis M. Contreras, Carlos J. Bcrnardos, and Ignacio Berberana. Understanding qos applicability in 5g transport networks. In *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–5, 2018.

[159] Line MP Larsen, Michael S Berger, and Henrik L Christiansen. Fronthaul for cloud-ran enabling network slicing in 5g mobile networks. *Wireless Communications and Mobile Computing*, 2018:1–8, 2018.

[160] Abin Mathew, Manikantan Srinivasan, and C. Siva Ram Murthy. Network calculus based delay analysis for mixed fronthaul and backhaul 5g networks. In *2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pages 205–214, 2020.

[161] Chathurika Ranaweera, Elaine Wong, Ampalavanapillai Nirmalathas, Chamil Jayasundara, and Christina Lim. 5g c-ran architecture: A comparison of multiple optical fronthaul networks. In *2017 International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6, 2017.

[162] Stela Kostadinova, Valentina Markova, Nikolay Kostov, Ivelina Balabanova, and Georgi Georgiev. Latency analysis for 5g optical transport network. In *2021 International Conference on Biomedical Innovations and Applications (BIA)*, volume 1, pages 9–12, 2022.

[163] Satyam Agarwal, Francesco Malandrino, Carla-Fabiana Chiasserini, and S. De. Joint vnf placement and cpu allocation in 5g. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 1943–1951, 2018.

[164] Satyam Agarwal, Francesco Malandrino, Carla Fabiana Chiasserini, and Swades De. Vnf placement and resource allocation for the support of

vertical services in 5g networks. *IEEE/ACM Transactions on Networking*, 27(1):433–446, 2019.

[165] Yu Bi, Carlos Colman-Meixner, Rui Wang, Fanchao Meng, Reza Nejabati, and Dimitra Simeonidou. Resource allocation for ultra-low latency virtual network services in hierarchical 5g network. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–7, 2019.

[166] Qiang Ye, Weihua Zhuang, Xu Li, and Jaya Rao. End-to-end delay modeling for embedded vnf chains in 5g core networks. *IEEE Internet of Things Journal*, 6(1):692–704, 2019.

[167] Prabhu Kaliyammal Thiruvasagam, Vijeth J. Kotagi, and C. Siva Ram Murthy. The more the merrier: Enhancing reliability of 5g communication services with guaranteed delay. *IEEE Networking Letters*, 1(2):52–55, 2019.

[168] Abdulaziz Abdulghaffar, Ashraf Mahmoud, Marwan Abu-Amara, and Tarek Sheltami. Modeling and evaluation of software defined networking based 5g core network architecture. *IEEE Access*, 9:10179–10198, 2021.

[169] Yue Liu, Xiaoyang Li, Yanhui Lin, Rui Kang, and Lianghua Xiao. A colored generalized stochastic petri net simulation model for service reliability evaluation of active-active cloud data center based on it infrastructure. In *2017 2nd International Conference on System Reliability and Safety (ICSRS)*, pages 51–56, 2017.

[170] Stefan Schneider, Arnab Sharma, Holger Karl, and Heike Wehrheim. Specifying and analyzing virtual network services using queuing petri nets. In *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 116–124, 2019.

[171] Daniel Carvalho, Laécio Rodrigues, Patricia Takako Endo, Sokol Kosta, and Francisco Airton Silva. Mobile edge computing performance evaluation using stochastic petri nets. In *2020 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6, 2020.

[172] Lanlan Rui, Xushan Chen, Xiaomei Wang, Zhipeng Gao, Xuesong Qiu, and Shangguang Wang. Multiservice reliability evaluation algorithm considering network congestion and regional failure based on petri net. *IEEE Transactions on Services Computing*, 15(2):684–697, 2022.

[173] H. Kerivin, D. Nace, and T.-T.-L. Pham. Design of capacitated survivable networks with a single facility. *IEEE/ACM Transactions on Networking*, 13(2):248–261, 2005.

[174] Vincenzo Eramo, Emanuele Miucci, Mostafa Ammar, and Francesco Giacinto Lavacca. An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures. *IEEE/ACM Transactions on Networking*, 25(4):2008–2025, 2017.

[175] Behrooz Farkiani, Bahador Bakhshi, and Seyyed Ali Mirhassani. A fast near-optimal approach for energy-aware sfc deployment. *IEEE Transactions on Network and Service Management*, 16(4):1360–1373, 2019.

[176] Csaba Rotter and Tien Van Do. A queueing model for threshold-based scaling of upf instances in 5g core. *IEEE Access*, 9:81443–81453, 2021.

[177] Yi Ren, Tuan Phung-Duc, Yi-Kuan Liu, Jyh-Cheng Chen, and Yi-Hao Lin. Asa: Adaptive vnf scaling algorithm for 5g mobile networks. In *2018 IEEE 7th International Conference on Cloud Networking (CloudNet)*, pages 1–4, 2018.

[178] Jorge Ortin, Pablo Serrano, Jaime Garcia-Reinoso, and Albert Banchs. Analysis of scaling policies for nfv providing 5g/6g reliability levels with fallible servers. *IEEE Transactions on Network and Service Management*, 19(2):1287–1305, 2022.

[179] Shirin Nikmanesh, Mohammad Akbari, and Roghayeh Joda. Proactive self-healing analysis-framework based on discrete-time markov decision process in 5g network and beyond. In *2018 9th International Symposium on Telecommunications (IST)*, pages 690–695, 2018.

[180] René David and Hassane Alla. Petri nets for modeling of dynamic systems: A survey. *Automatica*, 30(2):175–202, 1994.

[181] Peter Huber, Kurt Jensen, and Robert M Shapiro. Hierarchies in coloured petri nets. In *Advances in Petri Nets 1990 10*, pages 313–341. Springer, 1991.

[182] Hoon Choi, Vidyadhar G. Kulkarni, and Kishor S. Trivedi. Transient analysis of deterministic and stochastic petri nets. In Marco Ajmone Marsan, editor, *Application and Theory of Petri Nets 1993*, pages 166–185, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[183] András Horváth, Antonio Puliafito, Marco Scarpa, and Miklós Telek. Analysis and evaluation of non-markovian stochastic petri nets. In *Computer Performance Evaluation. Modelling Techniques and Tools: 11th International Conference, TOOLS 2000 Schaumburg, IL, USA, March 27–31, 2000 Proceedings 11*, pages 171–187. Springer, 2000.

[184] Pierre Dersin and René C Valenzuela. Application of non-markovian stochastic petri nets to the modeling of rail system maintenance and

availability. In *Proceedings of the 2012 Winter Simulation Conference (WSC)*, pages 1–12. IEEE, 2012.

[185] Bruno Pinna, Génia Babykina, Nicolae Brinzei, and Jean-François Pétin. Using coloured petri nets for integrated reliability and safety evaluations. *IFAC Proceedings Volumes*, 46(22):19–24, 2013.

[186] Lawrence E Holloway, Bruce H Krogh, and Alessandro Giua. A survey of petri net methods for controlled discrete event systems. *Discrete event dynamic systems*, 7:151–190, 1997.

[187] Behrouz Safarinejadian. Discrete event simulation and petri net modeling for reliability analysis. *Int. J. Soft Comput. Softw. Eng*, 2(5):2251–7545, 2012.

[188] Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[189] Marco Ajmone Marsan, Gianni Conte, and Gianfranco Balbo. A class of generalized stochastic petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems (TOCS)*, 2(2):93–122, 1984.

[190] Louchka Popova-Zeugmann and Louchka Popova-Zeugmann. *Time petri nets*. Springer, 2013.

[191] Kurt Jensen. *Coloured Petri nets: basic concepts, analysis methods and practical use*, volume 1. Springer Science & Business Media, 1996.

[192] Dongsheng Liu, Jianmin Wang, Stephen CF Chan, Jiaguang Sun, and Li Zhang. Modeling workflow processes with colored petri nets. *computers in industry*, 49(3):267–281, 2002.

[193] Falko Bause. Queueing petri nets-a formalism for the combined qualitative and quantitative analysis of systems. In *Proceedings of 5th international workshop on Petri nets and performance models*, pages 14–23. IEEE, 1993.

[194] The Kubernetes Authors. Horizontal pod autoscaling, kubernetes documentation. https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale, 2023. [Online; accessed 03 July 2023].

[195] Mihaela Matcovschi, Constantin Popescu, and Octavian Pastravanu. A new approach to hybrid system simulation: Development of a simulink library for petri net models. *Journal of Control Engineering and Applied Informatics*, 7(4):55–62, 2005.

[196] Franck Pommereau. Snakes: A flexible high-level petri nets library (tool paper). In *Application and Theory of Petri Nets and Concurrency: 36th International Conference, PETRI NETS 2015*, pages 254–265. Springer, 2015.

[197] Zhu En Chay, Bing Feng Goh, and Maurice HT Ling. Pnet: A python library for petri net modeling and simulation. *arXiv preprint arXiv:2302.12054*, 2023.

[198] Marco Paolieri, Marco Biagi, Laura Carnevali, and Enrico Vicario. The ORIS Tool: Quantitative Evaluation of Non-Markovian Systems. *IEEE Trans. Software Eng.*, 47(6):1211–1225, 2021.

[199] A. Buchmann, C. Dutz, S. Kounev, A. Buchmann, C. Dutz, and S. Kounev. Qpme - queueing petri net modeling environment. In *Third International Conference on the Quantitative Evaluation of Systems - (QEST'06)*, pages 115–116, 2006.

[200] Thomas Freytag and Martin Sänger. Woped-an educational tool for workflow nets. In *Business Process Management Demo Sessions 2014*, pages 31–35, 2014.

[201] Anne Vinter Ratzer, Lisa Wells, Henry Michael Lassen, Mads Laursen, Jacob Frank Qvortrup, Martin Stig Stissing, Michael Westergaard, Søren Christensen, and Kurt Jensen. Cpn tools for editing, simulating, and analysing coloured petri nets. In *International conference on application and theory of petri nets*, pages 450–462. Springer, 2003.

[202] Weng Jie Thong and MA Ameedeen. A survey of petri net tools. In *Advanced Computer and Communication Engineering Technology: Proceedings of the 1st International Conference on Communication and Computer Engineering*, pages 537–551. Springer, 2015.

[203] the Petri Nets World. Petri nets tools database quick overview. https://www2.informatik.uni-hamburg.de/TGI/PetriNets/tools/quick.html, 2021. [Online; accessed 03 July 2023].

[204] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, January 2017.

[205] Stefano Sebastio, Rahul Ghosh, Avantika Gupta, and Tridib Mukherjee. Contav: A tool to assess availability of container-based systems. In *2018 IEEE 11th Conference on Service-Oriented Computing and Applications (SOCA)*, pages 25–32. IEEE, 2018.

[206] Jing Jiang, Jie Lu, Guangquan Zhang, and Guodong Long. Optimal cloud resource auto-scaling for web applications. In *2013 13th IEEE/ACM international symposium on cluster, Cloud, and Grid Computing*, pages 58–65. IEEE, 2013.

[207] Laszlo Toka, Gergely Dobreff, Balazs Fodor, and Balazs Sonkoly. Adaptive ai-based auto-scaling for kubernetes. In *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, pages 599–608. IEEE, 2020.

[208] Gergő Pintér and Imre Felde. Analyzing the behavior and financial status of soccer fans from a mobile phone network perspective: Euro 2016, a case study. *Information*, 12(11):468, 2021.

[209] Delia Rico and Pedro Merino. A survey of end-to-end solutions for reliable low-latency communications in 5g networks. *IEEE Access*, 8:192808–192834, 2020.

[210] Steven M Pincus. Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences*, 88(6):2297–2301, 1991.

[211] 3GPP-TR 38.801. Study on new radio access technology: Radio access architecture and interfaces (release 14). Technical report (tr), 3rd Generation Partnership Project (3GPP), March 2017. version 14.0.0.

[212] Silivia Canale, Marco Tognaccini, Lourdes Maria de Pedro, Jaime Jesus Ruiz Alonso, Kostas Trichia, Despina Meridou, Andreas Georgakopoulos, Ioannis Maistros, George Loukas, Vincent Audebert, Tilemachos Doukoglou, Maria Kitra, Andreas Tzoulis, Evangelia Tzifa, Rodolphe Legouable, and Roberto Gavazzi. D1.1 Requirements Definition & Analysis from Participant Vertical Industries. https://doi.org/10.5281/zenodo.3530391, October 2018. [Online; accessed 07 October 2023].

[213] Mohand Ouamer Nait Belaid, Vincent Audebert, Boris deneuville, and Rami Langar. Defining an mv/lv protection, automation, and control system based on 5g network. *CIGRE Science and Engineering*, 27(B5):1–17, 01 2023.

[214] UIC. Frmcs and 5g for rail: challenges, achievements and opportunities., December 2020. Publication of UIC rail system department.

[215] Ruisi He, Bo Ai, Gongpu Wang, Ke Guan, Zhangdui Zhong, Andreas F Molisch, Cesar Briso-Rodriguez, and Claude P Oestges. High-speed railway communications: From gsm-r to lte-r. *Ieee vehIcular technology magazIne*, 11(3):49–58, 2016.

[216] UIC. Lte/sae – the future railway mobile radio system: Long-term vision on railway mobile radio technologies., November 2009. UIC Technical Report.

[217] Yong Niu, Yong Li, Depeng Jin, Li Su, and Athanasios V Vasilakos. A survey of millimeter wave communications (mmwave) for 5g: opportunities and challenges. *Wireless networks*, 21:2657–2676, 2015.

[218] Leonardo Bonati, Michele Polese, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. Open, programmable, and virtualized 5g networks: State-of-the-art and the road ahead. *Computer Networks*, 182:107516, 2020.

[219] Pingzhi Fan, Jing Zhao, and Chih-Lin I. 5g high mobility wireless communications: Challenges and solutions. *China Communications*, 13(2):1–13, 2016.

[220] Naser Al-Falahy and Omar Y. Alani. Technologies for 5g networks: Challenges and opportunities. *IT Professional*, 19(1):12–20, 2017.

[221] Hasan Farooq, Md. Salik Parwez, and Ali Imran. Continuous time markov chain based reliability analysis for future cellular networks. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2015.

[222] Long Qu, Maurice Khabbaz, and Chadi Assi. Reliability-aware service chaining in carrier-grade softwarized networks. *IEEE Journal on Selected Areas in Communications*, 36(3):558–573, 2018.

[223] Prabhu Kaliyammal Thiruvasagam, Vijeth J. Kotagi, and C Siva Ram Murthy. A reliability-aware, delay guaranteed, and resource efficient placement of service function chains in softwarized 5g networks. *IEEE Transactions on Cloud Computing*, 10(3):1515–1531, 2022.

[224] Hao Song, Xuming Fang, and Li Yan. Handover scheme for 5g c/u plane split heterogeneous network in high-speed railway. *IEEE Transactions on Vehicular Technology*, 63(9):4633–4646, 2014.

[225] Rasha El Banna, Hussein M. EL Attar, and Mohamed Aboul-Dahab. Handover scheme for 5g communications on high speed trains. In *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*, pages 143–149, 2020.

[226] Şafak Sönmez, Ibraheem Shayea, Sajjad Ahmad Khan, and Abdulraqeb Alhammadi. Handover management for next-generation wireless networks: A brief overview. *2020 IEEE Microwave Theory and Techniques in Wireless Communications (MTTW)*, 1:35–40, 2020.

[227] Jawad Tanveer, Amir Haider, Rashid Ali, and Ajung Kim. An overview of reinforcement learning algorithms for handover management in 5g ultra-dense small cell networks. *Applied Sciences*, 12(1), 2022.

[228] Ricky W Butler and George B Finelli. The infeasibility of quantifying the reliability of life-critical real-time software. *IEEE Transactions on Software Engineering*, 19(1):3–12, 1993.

[229] Zhenghua Xue, Xiaoshe Dong, Siyuan Ma, and Weiqing Dong. A survey on failure prediction of large-scale server clusters. In *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, volume 2, pages 733–738. IEEE, 2007.

[230] W Earl Smith, Kishor S Trivedi, Lorrie A Tomek, and Jerry Ackaret. Availability analysis of blade server systems. *IBM Systems Journal*, 47(4):621–640, 2008.

[231] William J Stewart. *Introduction to the numerical solution of Markov chains*. Princeton University Press, 1995.

[232] Martin L Shooman. *Reliability of computer systems and networks: fault tolerance, analysis, and design*. John Wiley & Sons, 2003.

[233] Algirdas Avizienis, J-C Laprie, Brian Randell, and Carl Landwehr. Basic concepts and taxonomy of dependable and secure computing. *IEEE transactions on dependable and secure computing*, 1(1):11–33, 2004.

[234] W Johnson Barry. *DESIGN AND ANALYSIS OF FAULT-TOLERANT DIGITAL SYSTEMS. CORR.* ADDISON-WESLEY., 1989.

[235] 3GPP-TS38.300. Nr; nr and ng-ran overall description; stage-2. Technical Specification (TS) 38.300, 3rd Generation Partnership Project (3GPP), January 2021. version 16.4.0.

[236] The R3 research group. Risk reliability resilience (r3) research group. http://r3.centralesupelec.fr/, 2023. [Online; accessed 10 August 2023].

[237] Thang Le Duc, Rafael García Leiva, Paolo Casari, and Per-Olov Östberg. Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey. *ACM Computing Surveys (CSUR)*, 52(5):1–39, 2019.

[238] Charles Ssengonzi, Okuthe P Kogeda, and Thomas O Olwal. A survey of deep reinforcement learning application in 5g and beyond network slicing and virtualization. *Array*, 14:100142, 2022.

[239] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[240] OpenAI. Openai gym. toolkit for developing and comparing reinforcement learning algorithms. https://gym.openai.com/, 2023. [Online; accessed 10 October 2023].

Appended papers

Paper I

Rui Li, Bertrand Decocq, Anne Barros, Yiping Fang, and Zhiguo Zeng. Complexity in 5G Network Applications and use cases. In *31st European Safety and Reliability Conference*, pages 3054–3061, Angers, France, September 2021.

Paper II

Rui Li, Bertr Decocq, Anne Barros, Yiping Fang, and Zhiguo Zeng. Petri Net-Based Model for 5G and Beyond Networks Resilience Evaluation. In *2022 25th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pages 131–135, Paris, France, March 2022.

Paper III

Rui Li, Bertrand Decocq, Anne Barros, Yiping Fang, and Zhiguo Zeng. Modélisation d'un réseau 5G par des réseaux de Pétri pour estimer sa résilience. In *AlgoTel 2022 - 24èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications*, pages 1–4, Saint-Rémy-Lès-Chevreuse, France, May 2022.

Paper IV

Rui Li, Bertrand Decocq, Yiping Fang, Zhiguo Zeng, and Anne Barros. A Petri Net-based model to study the impact of traffic changes on 5G network resilience. In *32nd European Safety and Reliability Conference (ESREL 2022)*, pages 3016–3023, Dublin, Ireland, August 2022.

Paper V

Rui Li, Bertrand Decocq, Anne Barros, Yi-Ping Fang, and Zhiguo Zeng. Estimating 5G network service resilience against short timescale traffic variation. *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pages 2230-2243, September 2023.

Paper VI

Rui Li, Bertrand Decocq, Anne Barros, Yiping Fang, and Zhiguo Zeng. Reliability challenges of 5g and beyond networks applications in high-speed trains. In *33rd European Safety and Reliability Conference (ESREL 2023)*, pages 1935–1942, Southampton, UK, September 2023.

Paper VII Rui Li, Bertrand Decocq, Yiping Fang, Zhiguo Zeng, and Anne Barros. High-mobility 5g communication service: availability and reliability analysis. In *7th International Conference on System Reliability and Safety (ICSRS 2023)*, accepted, November 2023.

COMPLEXITY IN 5G NETWORK APPLICATIONS AND USE CASES

Rui Li, Bertrand Decocq

Orange Innovation, Châtillon, France. E-mail: rui.li@orange.com, bertrand.decocq@orange.com

Anne Barros, Yiping Fang, Zhiguo Zeng

Laboratoire Génie Industriel, CentraleSupélec, Université Paris-Saclay, Gif-sur-Yvette, France.
E-mail: anne.barros@centralesupelec.fr, yiping.fang@centralesupelec.fr, zhiguo.zeng@centralesupelec.fr

The fifth generation (5G) of mobile telecommunication network is designed with an ambition to be a network faster, stronger, better and smarter than its predecessor. With the digital transformation, all industry sectors will develop new applications with new requirements regarding telecommunication networks that 5G should be able to meet. To meet the requirement of future 5G use cases and applications, it is crucial to study the complexity of such network system by distinguishing different parts, layers, components as well as their interdependencies. This paper describes the 5G networks from an End-to-End perspective (device, radio network, core network, data network) and from a multi-layer perspective (orchestration, virtualisation/containerization and infrastructure) to show how this system (or system of systems) is complex, especially when we address resilience challenges. Resilience requirements and challenges are further explained by proposing relevant scenarios and use cases. In this paper, we mainly intend to highlight 5G network complexity and open a discussion on methodologies to model such complex network for its resilience study with the hope that this paper could inspire the future study of researchers in the related field.

Keywords: 5G Network, resilience quantification, resilience metrics, network applications, complex system, vertical requirements.

1. Introduction

The telecommunication domain keeps evolving rapidly since its birth. From the first generation cellular network to the newest generation, every one of them brings convenience to daily life and work. As the key to the future technology ecosystem, the fifth generation (5G) of mobile telecommunication network is designed to be a faster, stronger, better and smarter telecommunication network than ever before.

By upgrading existing technologies and incorporating new technologies, 5G networks are without a doubt a promising solution for future telecommunication needs.

In the Radio Access Network (RAN), new radio technologies are applied to 5G networks. Terminals are eligible to use both 5G and 4G frequency bands and connect to both 4G and 5G antenna. The utilization of orthogonal frequency-division multiplexing (OFDM) allows multiple communication channels to coexist, and thus it is possible to treat high frequency and low frequency bands at the same time to obtain both higher bandwidth and wider coverage. Intelligent antennas using "massive MIMO" are implemented which can further improve network capacity (IEEE 802.11ad, 2012; Patriciello et al., 2020). Besides, the beamforming technology ensures to transmit signal in a specific direction where it is useful to users rather than sending in all directions, such that less interference is created and less energy is consumed.

The 5G core network (5GC) becomes a service based architecture. In this software designed architecture, each Network Function (NF) is delivering "services" to other NFs to access control plane functionalities, subscriber or network data repositories through an interface of a common framework (3GPP TS 23.501, 2021; Mademann, 2018). To deliver services more dynamically, 5G networks adopt the techniques such as Network Function Virtualisation (NFV) and Software Defined Networking (SDN).

By introducing the concept of slicing into the network, it is possible to create different virtual networks for different services. In 5G, such network will be a dedicated slice providing tailored network capabilities and network characteristics according to the requirement from the customers by respecting specific rules without disturbing the rest of the network outside the slice. Multiple users, if permitted, can connect to one same slice. One user equipment, if needed, may have access to multiple slices at one time. A network slice subnet represents a group of network functions that form part or complete constituents of a network slice. A network slice subnet may contain for example instances of Core Network functions only, or instances of Access Network functions only, or any combination thereof (3GPP TS 28.530, 2021).

Edge computing is a generic term encompassing a variety of different approaches to put computing and storage resources at the edge of the network close to the customer rather than in remote datacenters. Initially, this notion was introduced and used for mobile networks, hence the term Mobile Edge Computing (5G Smart, 2020). Later, the European Telecommunications Standards Institute (ETSI) defined the term Multi-access Edge Computing (MEC) as a

Proceedings of the 31th European Safety and Reliability Conference
Edited by Bruno Castanier, Marko Cepin, David Bigaud and Christophe Berenguer
 Published by Research Publishing, Singapore
ISBN: 981-973-0000-00-0 :: doi: 10.3850/981-973-0000-00-0 esrel2021-paper

generalization of Mobile Edge Computing to any network (ETSI, 2019). Some latency-sensitive network application functions can be deployed on MEC servers near the RAN or even at the macro base station. Therefore, some of the data will be stored and proceeded in distributed edge cloud services.

From an End-to-End perspective, these technologies break the boundary of different parts of network resources. From a multi-layer perspective, with NFV, a virtualization layer is added into the network architecture. Thus, 5G becomes a complex system and even a system of systems.

Combining the afore-cited technologies, 5G networks are going to greatly reshape the domain not only by its performance but also by offering a transition from a "horizontal" service delivery model toward a "vertical" service delivery model (Banchs et al., 2019). The former model provides identical services to all customers, while the latter provides tailored and personalized services for certain industry sectors. Such "vertical" delivery model introduces new scenarios and application use cases to 5G networks.

International Telecommunication Union (ITU-R, 2015) classifies 5G services into three categories: enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable and low-latency communications (URLLC). Virtual or Augmented Reality, one of eMBB use cases, which constantly transfers a huge amount of data could benefit from 5G high speed network connection. In the example of smart fabrication, a massive machine-type communications service, the network slicing may help the factory to check its production by connecting IoT equipment to a dedicated slice. Autonomous-driving, as a typical URLLC service, its massive data can be calculated in real time at the edge with the help of 5G networks.

With the digital and technological transformation, 5G can be applied to more and more scenarios, nevertheless, it also creates a resilience challenge not only for the service providers but also for all the supported industries or verticals. The more technologies are integrated into the system, the more potential risks. Furthermore, 5G becomes a complex system, one single failure, if not fully fixed, could propagate from one single point to a series elements of the system. Thus, it is crucial to study the complexity of such network system by comprehending each part and each layer of it. Only when we have a full understanding on how 5G networks are composed and how a Network Service is established, can we analyze the risk and resilience of the system.

The paper is structured as follows: in Section 2 we explain 5G complexity from an End-to-End perspective by decomposing each part of the End-to-End service; Section 3 is devoted to presenting a multilayer perspective of 5G network, where the Network Function Virtualisation is mainly discussed; in Section 4 we briefly describe the process of setting up a service using the technologies that we introduce; Section 5 takes on the resilience challenge by introducing adverse events in telecommunication network and recent works on performance evaluation in 5G system; Finally we conclude the paper by providing suggestions for further research in Section 6.

2. The complexity from the End-to-End perspective

Figure 1 shows the End-to-End architecture in 5G network. It includes in general terminals, Next generation Radio Access Network (Ng-RAN), Transport Network, 5G Core Network, Data network. In a 5G use cases such as autonomous vehicles, more than one Network Service may be needed. A Network Service may traverse all or only part of the aforementioned elements.

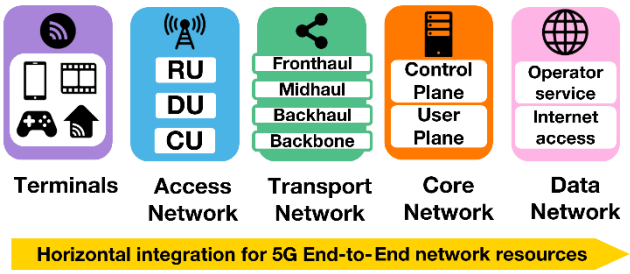


Fig. 1. 5G End-to-End Architecture.

2.1. *Terminals*

A Network Service normally starts from an end-user device. These devices are called terminals or user equipment (UE). Since 5G will be widely applied in telecommunication, variant terminals will be connected to the network. These terminals could be smart phones, vehicles, IoT terminals, etc. Among them, some terminals require simultaneous connections to multiple services. More specifically, a typical application would be the case where a smartphone plays an online football match for the user while in the background the device tries to get a push notification from mail service. In other cases, some devices may require the same service at the same time. Such scenario commonly happens in a factory. When an environment change is detected, all connected sensors will report this abnormal data or the extracted information to the central server at the same time. In the multiple services situation, the isolation between Network Services must be guaranteed to meet the requirement of each one of the services. In the massive devices access situation, the system must be resilient enough to cope with a potential congestion in the data processing and transport.

2.2. *Radio access network*

For a wireless terminal, to transfer data from the UE to the Radio Access Network (RAN), the data will be firstly received by the antenna embedded in Radio Unit (RU). Baseband Unit (BBU) connected to the RU will then transport a baseband frequency before sending the data to the edge or Core Network. Open RAN as a future generation of RAN chosen by 5G network, provides a standardized interface between RU and BBU as well as a standardized interface between BBUs to cooperate with multiple vendors. With virtualized RAN (vRAN), the BBU can be virtualized on multiple NFV platforms and be shared with operators (see Section 3.1 for more details on Network Function Virtualisation). BBU can be divided into multi parts. The first part, distributed unit (DU), takes charge of real-time BBU scheduling functions, while the second part,

centralized unit (CU), completes the non-real-time BBU functions. Some software parts of BBU will be placed together with RU. CU and DU can be deployed flexibly, namely co-located with RU, in edge cloud or regional datacenter. Virtualisation and standardized interface make the most of open interfaces by enabling sharing CU and DU with multiple vendors (Wind River, 2017).

2.3. *Transport network*

To ensure a highly reliable and good performance network, Transport Network plays a crucial role. Transport Network includes the fronthaul of remote units, the backhaul of base stations, optionally a midhaul between the distributed and centralized units, and the backbone between core datacenters. Different transmission technologies are used for each part of Transport Network, for example: dark fiber for fronthaul and midhaul with direct connections between the nodes (RU to DU and DU to CU respectively), WDM rings for backhaul and backbone networks. Network Slicing can be based in the first step on VLAN/VPN for each transport segment (called basic soft-slicing or logical isolation between slices), and later on new technologies like Segment Routing-Traffic Engineering (SR-TE) for enhanced soft slicing with specific performance or designed per type of slice, and in the third step on Flexible Ethernet (FlexE) or Time Sensitive Networking (TSN) for hard slicing where the slices are fully isolated with guaranteed services performance. For resilience purposes, the IP network, from the Edge to the Core Network, is doubled and relies on WDM rings, and is able to react in 50 milliseconds in case of failure.

2.4. *Core network*

In 5GC, one of the most important characteristics is the separation of the User Plane (UP) functions from the Control Plane (CP) functions (3GPP TS 23.501, 2021). UP functions mainly take care of traffics forwarding while the CP functions manage the authentication, network slice selections, etc. The principal advantage of such separation is being able to flexibly scale the CP functions independently on UP functions in case of traffic peak vice versa. Another benefit lies in the flexibility to separately deploy CP functions so that some functions can be deployed, according to the requirement of the use case, in a centralized datacenter or a distributed one close to the RAN. The flexibility in scaling and deployment completely makes 5G networks more complex than the last generation.

Core functions

Figure 2 depicts a 5G network architecture. The upper part of the architecture shows the 5GC Control Plane which uses Service-based interfaces. The 5GC Control Plane consists of the following Core Network Functions (NF).

- Authentication Server Function (AUSF).
- Access and Mobility Management Function (AMF).
- Data Network (DN), e.g. operator services, Internet access or 3rd party services.
- Network Exposure Function (NEF).
- Network Repository Function (NRF).
- Network Slice Specific Authentication and Authorization Function (NSSAAF).
- Network Slice Selection Function (NSSF).
- Policy Control Function (PCF).
- Session Management Function (SMF).
- Unified Data Management (UDM).
- Unified Data Repository Function (UDF).
- Application Function (AF).

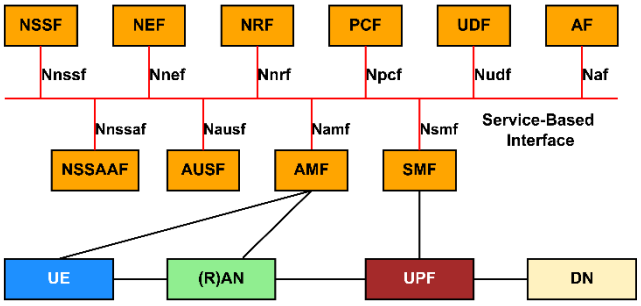


Fig. 2. 5G Service-Based Architecture.

Mobile Signaling is the engine of Mobile Networks Control Plane. As the Core Network adopts a service-based architecture, the signaling network elements are designed in form of Control Plane Network Functions. Each NF can thus expose a set of services called NF services to the service-based interface. Thus, these NFs can be both a consumer where they seek to consume the NF services provided by other NFs and an NF service producer where they provide their exposed services to NF service consumers. Each NF can provide multiple NF services for different NF consumers and can consume NF services from multiple service providers.

To avoid ambiguity in the paper, NF service refers to part of functionality of a Network Function that can be consumed by other NF, while Network Service defines a set of NFs connected together that facilitate a network operation.

2.5. *Edge computing*

Edge computing is an optional solution for 5G networks. The presence of Multi-access Edge Computing (MEC) reduces some Network Service latency as well as the network contention, resulting in a better service experience for end-users.

Apart from signaling services, a user equipment may also interact with other Network Functions such as third party application functions. By introducing edge computing, these functions are able to be hosted in a decentralized cloud. The main advantage of edge computing lies in the possibility to deploy such decentralized MEC cloud close to the UE's access point of attachment similar to the distributed deployment of some control plane NFs. It is indeed possible for a MEC to be deployed at the RAN edge, in a distributed datacenter or even in a centralized datacenter depending on the service requirement. Edge computing or MEC is not a new technology. To enable the interaction between MEC system and 5GC control plane, the design approach taken by 3GPP allows the mapping of MEC onto Application Functions (AF) (ETSI, 2018). MEC can thus

interact with the 5G system using Network Exposure Function (NEF) that provides information from external application to 3GPP network, or directly with the target 5G NFs if permitted (ETSI, 2020). Edge computing could be a suitable solution for URLLC type scenarios, e.g. collection and analysis of a large amount of information from massive IoT devices such as connected sensors. The application running on a MEC host deployed on the RAN edge could process the data locally and extract the useful information to the central server. Integrated with MEC system, 5G networks need to be resilient to guarantee the Core Network Function availability and to ensure the connection between MEC and 5G Core Network.

2.6. *Service function chaining*

To deliver an End-to-End service, various Network Functions are required. A service function chain (SFC) defines an ordered set of Network Functions and ordering constraints that must be applied to packets and/or frames and/or flows selected as a result of classification and/or policy to deliver such an End-to-End service. The mechanism of building such function chains and forwarding packets, frames or flows through them is called service function chaining (ITU-T, 2016). From an End-to-End perspective, a SFC defines how a Network Service is implemented. Since the Network Function Virtualisation is applied in 5G networks (see Section 3.1 for more details), the SFC becomes Virtualized Network Function (VNF) chain (see Figure 3 an example of two service function chains). To allocate SFC request on NFV Infrastructure is challenging. The VNF instances should be hosted at the server with enough resources. Some specific rules may define the isolation or co-location of VNF instances. VNFs may have specific behaviors: Some VNFs can be load balancers, thus parallel processing is allowed; Some VNFs may have multiple outputs (next VNF) depending on the attribute of input traversing traffic; Sometimes, the traffic arrived at a VNF cannot be processed immediately, it has to be queued. Taking into consideration all these constraints, the placement of SFC is a complex problem. A lot of works address this problem by proposing different approaches including resolving a shortest path problem (Martini et al., 2015), integer linear programming (ILP) (Baumgartner et al., 2015) or mixed-integer linear programming (MILP) (Dietrich et al., 2017).

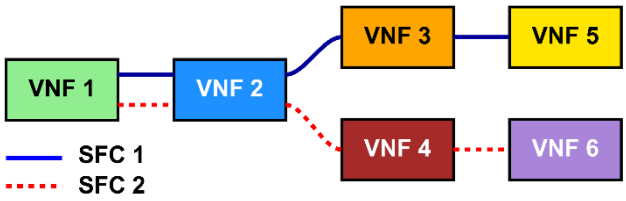


Fig. 3. Two VNF Chains in one network.

Figure 4 shows an example where two service function chains are deployed in the network. Both SFC 1 and 2 start from consuming VNF 1 and VNF 2. Since these two chains serve different services, they consume different VNFs separately thereafter. SFC 1 utilizes VNF 3, 5 and SFC 2 utilizes VNF 4, 6. Each VNF has several instances and they can be deployed on different servers. Multiple SFCs can share the same VNF instance or they consume different instances. In Figure 4, Server 2, 3 are used by both SFCs while Server 1 is used only by SFC 1 and Server 4 is needed only by SFC 2.

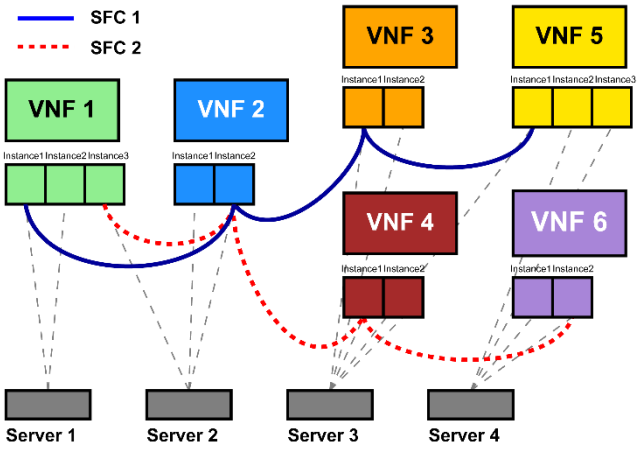


Fig. 4. Deployment of two Service Function Chains.

3. The complexity from the Multilayer perspective

The End-to-End perspective illustrates horizontally the complexity when delivering a Network Service. The multilayer perspective reveals vertically the complexity while orchestrating and managing a Network Service.

3.1. *Network function virtualisation*

With more and more new services joining the network, the resource allocation and service maintenance become a bottleneck for improving the service performance. In 5G networks, the Network Function Virtualisation (NFV) technology is proposed to solve the problem. This technology reforms the network architecture by separating software from the hardware with the help of virtualisation (Chiosi, 2012). Virtualizing Network Functions enables flexible distribution of hardware resources to improve the service performance and rapid launch of innovative services to generate new revenue sources. It is also an enabler for the formerly mentioned flexible deployment of several 5GC Network Functions, e.g. AMF, and co-locating them with the access network and thus eliminating long-distance data transport (Han et al., 2015). All core NFs and some access NFs are targeted to be virtualized in 5G networks in some scenarios (ENISA, 2020).

With virtualisation, the physical Network Functions become Virtualized Network Functions (VNFs). These VNFs can be deployed on virtual machines (VMs) or containers. The former is a traditional virtualisation environment while the latter is lighter-weight and more agile. The infrastructure resources including storage, compute and network are virtualized and useable for virtualized layer. Instead of allocating a fixed amount of physical resources, VNFs can allocate dynamically virtual resources according to the service request and traffic in real-time.

More specifically, NFV adopts a three-tier architecture (ETSI, 2014) as shown in Figure 5. At the top is the operation layer, with Business Support Systems (BSS) and Operations Support Systems (OSS) to support various End-to-End telecommunication services. Some processes covered by OSS/BSS include: network management, service delivery, fulfilment, assurance, and billing. Lower down is the Network Service and Network Function layer. Inside this layer, the Virtualized Network Functions are managed by Element Managements (EMs). EM's role includes security management, fault management for the exposed Network Function services provided by VNFs. At the bottom lies the NFV infrastructure (NFVI). Storage and compute are two main physical hardware resources which are normally pooled. Another physical resource is networking devices including routers and links. The virtualisation layer abstracts the hardware resources and decouples the VNF software from the underlying hardware, ensuring a hardware independent lifecycle for the VNFs. For the majority of current deployments, the virtualisation layer in an NFVI comprises a hypervisor to partition physical servers into VMs and a network controller, typically a Software-Defined Network (SDN) controller, to help partition the physical network that connects the physical servers into multiple virtual networks interconnecting groups of VMs. While the vast majority of NFV deployments is still based on hypervisor technologies, container-based virtualisation (a.k.a. Operating System (OS) virtualisation) is gaining momentum and might become the norm for 5G. Containers provide an isolation capability that allows multiple VNF instances to share the same host OS while virtual machines require a separate guest OS for each VNF instance.

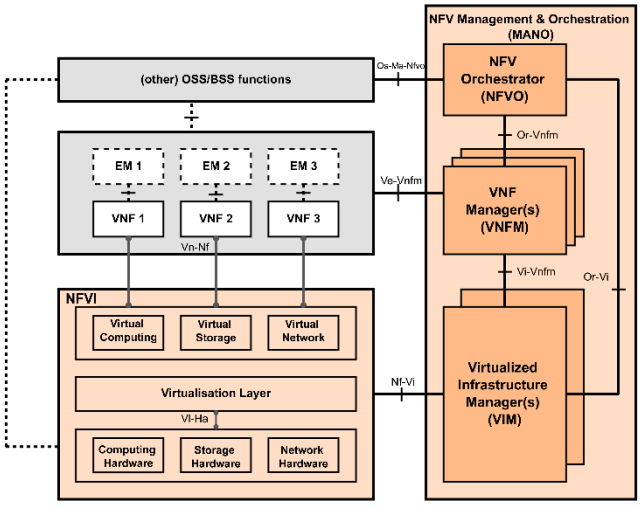


Fig. 5. Network Function Virtualisation architecture.

NFV Management and Network Orchestration (MANO) takes charge of the management of NFVI and orchestrates the allocation of resources needed by the Network Services and VNFs (ETSI, 2016). NFV MANO includes three functional blocks. NFV Orchestrator (NFVO) is responsible in general for the life cycle operations of a Network Service. NFVO functions can be classified into End-to-End resource orchestration, and Network Service orchestration. VNF Manager (VNFM) is in charge of the life cycle operations as well as performance, fault and configuration management of a VNF. Specifically, the managements include instantiation, heal, operation (changing the state), information modification, changing connectivity, scaling and termination. Each VNF manager serves one or multiple VNFs according to the network design. The third block, Virtualized Infrastructure Manager (VIM), involves all life cycle operations of a virtualized resource. Concretely, a VIM controls and manages the interaction of a VNF with physical and virtualized resources including compute, storage and network. Similar to VNFM, multiple VIMs can be deployed in the network.

3.2. *VNF deployment*

VNF can be VM based or container based depending on the choice of technologies. Traditional VMs virtualizes an underlying computer while the container is lighter, containing the code and dependencies needed, taking less time to image and to run an application.

A VNF is composed of one or multiple VNF components (VNFCs) (ETSI, 2014). A VNFC is a software entity in charge of different functionalities or data bases (in 5G network). Thus, a VNF is mapped to one or several VMs or containers in NFVI servers. It is worth noting that even though there could be multiple VNF components belonging to a same VNF instance, they are not necessarily deployed all on the same host.

3.3. *Example of operations in Network Service*

Scaling is a typical action to manage a Network Service. Scaling can be categorized into two classes: horizontal scaling and vertical scaling. The former includes scaling in and out, which refers to a process where one or more instances are removed or added. The latter action includes scaling up and down, which refers to a process of adding or releasing resources to or from an existing instance.

Network Service level scaling out and in are important operations during the management of Network Services (3GPP TR32.842, 2015). They may be triggered from OSS, by an operator manually or by some Network Manager-level functions (e.g. Load Balancing) automatically. Then NFVO will receive the request to scale out or in a Network Service instance. For a Network Service, scaling is necessary when one or several of the actual VNF instances in the network is/are overloaded or too redundant for the Network Service. The scaling out in the Network Service level can be done by scaling out or scaling up concerning VNF instances. The service level scaling in can be done by scaling in or scaling down the corresponding VNF instances. Precisely, Network Service can be scaled out (or in) by expanding (contracting) some of existing VNF instances or by instantiating new (terminating existing) VNF instances. When there is a resource change involved, NFVO will also send a request to VIM to allocate the changed resources.

The VNF level scaling can be triggered from NFVO, VNFM, EM, OSS, or manually by the operator. This scaling concerns the management of VNF components. The scaling

request will be received by VNFM and it performs the VNFC instantiation or termination procedure for horizontal scaling. In vertical scaling, VNFM requests an update of the resources for VNFC instances.

In case of threat or failure, an operation can be triggered from different levels of the network. When an overload situation happens on a server, this will cause malfunction on VNFCs relying on it. After failure or overload detection, multiple entities of the network may react competitively as depicted in Figure 6. In case of Kubernetes container, Kubernetes hypervisor may decide an instantiation of one or multiple related Pods (a group of containers) on available servers. VNFM may scale out or up the concerning VNFC. NFVO may also change the deployment flavor and apply a rule such as instantiating VNF or increasing the capacity allocated to the concerning VNF. The 5G network mechanism at the VNF level may decide to reduce the number of messages sent to the VNF hosted on the overloaded server or redirect the messages. All these decisions are helpful to some extent but not complementary. It is necessary to determine upstream the best entity which must make the decision according to the situation to avoid a bad decision making situation worse. If we find ourselves in a case of a signaling storm, deploying new instances of VNF or VNFC can impact the datacenter at large by saturating again the servers, whereas actions at the 5G network level would have made it possible to resolve the problem.

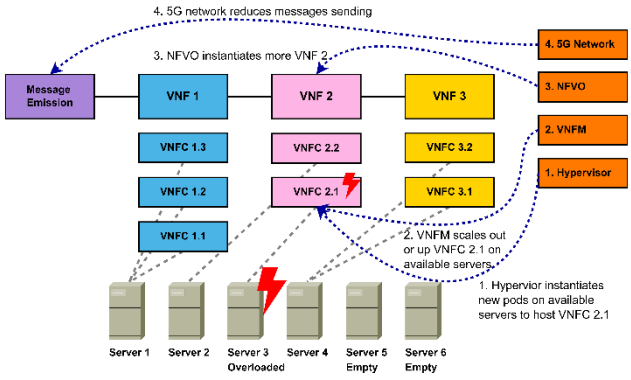


Fig. 6. Four competitive reactions of different network entities to face one overload situation.

4. Procedures of setting up a slice

5G networks enable a variety of Network Services. With the use of network slicing, the application requirements become more challenging and heterogeneous in order to better serve the vertical industry (ITU-T, 2018; Foukas et al., 2015). A use case from vertical industries may be translated into several Network Services in 5G networks. To set up these Network Services, several procedures such as service selection, resource allocation, are indispensable for Mobile Network Operator (MNO). A detailed example of procedures may be as follows:

- Expression of need and service selection: description of the service and associated requirements referring to an existing catalog for solution
- Communication service request with relevant inputs and performance requirements via Communication Service Management Function (CSMF): this step includes creating service order (Sending Customer Facing Service information to Service Order Management), updating resource order (sending Resource Facing Service information to Resource Order Management) and sending a request to Network Slice Management Function (NSMF) in order to create a Network Slice Instance (NSI) for this Service Profile
- Slice selection via NSMF: NSMF determines network Slice subnet requirements (slice profile for Subnet includes RAN, Core Network and Transport Network subnets) and the associated Network Slice Subnet Management Function (NSSMF); Secondly, NSMF sends a Network Slice Subnet Instance (NSSI) allocation request to NSSMF
- Subnet creation via NSSMF: a NSSMF takes charge of checking the feasibility of Network Slice Subnet Requirements, creation of a new NSSI as well as a New RAN Network Slice with RAN Network Functions (in RU, DU, CU), determination of Network Service Descriptor for Core Network NSSI components
- Service orchestration via NFV-MANO: NFVO derives a SFC, the location and behaviors of the VNFs from the requirements; NFVO interacts with the VNFM of each VNF and with the VIM of each datacenter; Each VNFM instantiates VNF by taking into account redundancy requirements, affinity or anti-affinity rules to select the best server to host the VNF components; VNFM sends the instantiation request to the VIM; VIM creates containers for VNF components, allocates resources for each container and ensures the connectivity between the servers inside the datacenter
- Transport Network Subnet creation via NSSMF: NSSMF derives requirements for Transport Network NSSI component
- Service orchestration via NFV-MANO in Transport Network: NFVO derives Network Slices Virtual Links Descriptor and interacts with WAN Infrastructure Manager to connect RAN and Core Network Point of Presence. The transport network will ensure the connectivity of all required functions from RAN or Core Networks.

5. Resilience challenges in 5G network

Resilience is a relatively new field in system engineering that drew great attention over the last decade. Resilience could be defined as the ability of a system to prepare and plan for, absorb, recover from, and more successfully adapt to adverse events. (National Academies Committee on Increasing National Resilience to Hazards and Disasters, 2012).

From ENISA Telecom Services Security Incidents 2019 Annual Analysis Report, system failures, human errors and natural phenomena are the three main causes of telecommunication incidents (ENISA, 2020). More than half of the telecom security incidents were caused by system failures. The most common system failures are hardware

failures and software bugs. This type of adverse events often happens on one single element, however other elements of the communication network may also be victims as a result of failure propagation due to interdependencies in telecommunication networks (Martins et al., 2017). Human errors normally result from an imperfect system design or a wrong configuration. For natural phenomena, the main characteristic is the broad effect in scope. For example, a blizzard would possibly result in multiple network nodes and links failures in the affected region. Other kinds of failures including malicious attacks are rare compared with the three main failures, thus are not the focus in our approach of resilience challenges in 5G network.

A resilient 5G network should be able to offer services with high Quality of Service (QoS) at all time regardless of the adverse events. QoS is the ability of a service to comply with quality requirements and service level as agreed (or targeted) with the end user. The QoS is often interpreted into important performance parameters of the telecommunication system, typically referred to Key Performance Indicators (KPIs) (Kukliński et al., 2019). As 5G is supposed to be a vertical service delivery model, the challenge of resilience in 5G systems resides in the fact that KPIs and service requirements vary from one case to another. These KPIs may include latency, availability, throughput, etc. In the use case of tele-action in power systems, the End-to-End service latency is the main KPI, which should be less than 50 milliseconds in order to guarantee the functioning of the grid network (ENEDIS, 2020). In another use case of autonomous vehicles in manufacturing environments, network availability should be higher than 99.90% and network End-to-End latency should be lower than 10 milliseconds (5G EVE, 2018).

To evaluate the performance and resilience of 5G Network Services, recent works focus on two types of situations, namely in the design process and system recovery process. With the NFV environment, Service Function Chain (SFC) becomes the carrier of Network Services. Then the problem is to evaluate and optimize the SFC deployment. In the SFC design process, it aims to prevent failure before it happens by analyzing for example necessary redundancy on each element of the chain and verify the KPIs on the chain as well as the redundant ones. In the recovery process, it is important to analyze the cost and time to provision the backup elements in the chain. Given the complexity of 5G, the problem needs to be solved by taking into consideration of both End-to-End and multi-layer perspectives. From the End-to-End perspective, a SFC should include the Radio Access Network, Core Network, Transport Network, and the Edge Computing and Data Network if applicable. While modeling the network, we should be capable to answer how is RAN deployed and where are RU and BBU deployed; what are the Control Plane Core Network Functions engaged in this SFC; what technologies are used on Fronthaul, Midhaul, Backhaul and Backbone; what are the elements co-located in the network. On the other hand, the multi-layer perspective emphasizes the complexity of SFC management. The operations such as network element scaling and failure recovery should be modeled. The competitive actions from different layers as presented in Section 3.3 will also be a challenge for resilience analysis.

The mainstream researches optimize the SFC deployment while subjecting to the resources, placement and performance constraints. These optimizations are mostly based on ILP or MILP models. The optimization goals vary from one to another, such as minimizing the bandwidth usage across the network (Qu et al., 2017), minimizing the total cost of deploying all network slice requests (Da Silva Coelho et al., 2020), minimizing the cost of protecting the SC against single failure (Carlinet et al., 2020), jointly minimizing the overall deployment cost and service delay (Leivadeas et al., 2019). These works simplify the Transport Network and Core Network and neglect the complex RAN. Only a few of them consider the system recovery process and these models often neglect the interdependency between each layer, the existence of MANO, the interaction between NFV-MANO and VNFs. For example, in the recovery process, the failure on MANO actually will block the VNF level or service level scaling.

Some recent works based on Petri Net have studies the recovery process (Rui et al., 2020), the decomposition of VNF (Di Mauro et al., 2017), Network Function behaviors (Schneider et al., 2019), NFV-MANO structure (Tola, et al., 2019) in the telecommunication network. The models based on Petri Net and its extensions seems to be a promising tool in Network Service evaluation since it can better describe the complex 5G network. Petri Nets' main attraction as a modeling formalism is how the basic aspects of concurrent systems are identified both conceptually and mathematically (Bonet, 2007). The marking of the state of a Petri Net model shows the state of the telecommunication system. The transition of a Petri Net model represents an action, e.g. scaling, failure and recovery. Its extensions such as Colored Petri Nets, Timed Petri Nets and Stochastic Petri Net enrich the capacity of the model and make it possible to measure the performance of the Network Service, e.g. availability and latency.

6. Conclusion

In this paper, we have introduced the complexity in 5G networks from both End-to-End and multi-layer perspectives. Some use cases are given to further explain the complexity in setting up a Network Service. By implementing new technologies, in particular NFV, 5G networks are becoming more flexible but also more complex. Therefore, more and more resilience challenges are awaiting to be taken before introducing 5G networks into new scenarios.

As we noted, modeling the network and its complexity is an important step for evaluating Network Service performance. We propose Petri-Net as a promising tool for the 5G network performance analysis. The work on modelling 5G networks to evaluate the resilience of an End-to-End service related to verticals, is in progress.

ACKNOWLEDGMENT

This work is funded by Orange in the framework of the Chair on Risk and Resilience of Complex Systems (CentraleSupélec, EDF, Orange, SNCF)

References

Specific Requirements. Part 11: Wireless LAN Medium Access Control, Standard IEEE 802.11ad, 2012.

Patriciello, N., S. Lagén., B. Bojović and L. Giupponi (2020). NR-U and IEEE 802.11 Technologies Coexistence in Unlicensed mmWave Spectrum: Models and Evaluation. *IEEE access 8*, 71254-71271.

3GPP (2021 (accessed April 2021)). *TS 23.501 V17.0.0 System architecture for the 5G System (5GS)*.

Mademann, F. (2018). The 5G system architecture. *Journal of ICT Standardization* 6 (3), 77-86.

3GPP (2021 (accessed April 2021)). *TS 28.530 V16.9.0 Management and orchestration; Concepts, use cases and requirements*

5G-SMART (2020 (accessed April 2021)). 5G Common Terminology. *5G Smart Manufacturing.* https://5gsmart.eu/wp-content/uploads/5G-SMART-common-terminology.pdf

ETSI (2019 (accessed April 2021)). *GS MEC 001 V2.1.1. Multi-access Edge Computing (MEC) Terminology*.

Banchs, A., D. M. Gutierrez-Estevez, M. Fuentes, M. Boldi, and S. Provvedi (2019). A 5G mobile network architecture to support vertical industries. *IEEE Communications Magazine 57*(12), 38-44.

ITU-R (2015). Framework and overall objectives of the future development of IMT for 2020 and beyond. *Radiocommunication Sector of ITU*.

Wind River (2017(accessed April 2021)). vRAN: The Next Step in Network Transformation.

ETSI (2018 (accessed April 2021)). MEC in 5G networks. *ETSI White Paper No. 28.*

ETSI (2020 (accessed April 2021)). *GR MEC 031 V2.1.1. Multi-access Edge Computing (MEC): MEC 5G Integration.*

ITU-T (2016). Deployment models of service function chaining. *ITU Y-series Recommendations–Supplement 41*.

Martini, B., F. Paganelli, P. Cappanera, S. Turchi, and P. Castoldi (2015). Latency-aware composition of virtual functions in 5G. *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, 1-6.

Baumgartner, A., V. S, Reddy, and T. Bauschert (2015). Mobile core network virtualization: A model for combined virtual core network function placement and topology optimization. *Proceedings of the 2015 1st IEEE conference on Network Softwarization (NetSoft)*, 1-9.

Dietrich, D., C. Papagianni, P. Papadimitriou, and J.S. Baras (2017). Network function placement on virtualized cellular cores. *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, 259-266.

Chiosi, M. (2012). Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges & Call for Action. *ETSI White Paper*.

Han, B., V. Gopalakrishnan, L. Ji, and S. Lee (2015). Network function virtualization: Challenges and opportunities for innovations. *IEEE Communications Magazine 53*(2), 90-97.

The European Union Agency for Cybersecurity (ENISA) (2020). *ENISA threat landscape for 5G networks – Updated threat assessment for the fifth generation of mobile telecommunications networks (5G)*

ETSI (2014 (accessed April 2021)). *GS NFV-MAN 001 V1.1.1 Network Function Virtualisation (NFV); Management and Orchestration.*

ETSI (2014 (accessed April 2021)). *GS NFV-SWA 001 V1.1.1 (2014-12) Network Functions Virtualisation (NFV); Virtual Network Functions Architecture*

3GPP (2015 (accessed April 2021)). *TR 32.842 V13.1.0. Telecommunication management; Study on network management of virtualized networks*

ITU-T (2018). Service function chaining in mobile networks. *Telecommunication standardization sector of ITU*.

Foukas, X., G. Patounas, A. Elmokashfi, and M.K. Marina (2017). Network slicing in 5G: Survey and challenges. *IEEE Communications Magazine 55*(5), 94-100.

Committee on Increasing National Resilience to Hazards and Disasters (2012), *Disaster Resilience: A National Imperative*, Natl. Acad. Press.

The European Union Agency for Cybersecurity (ENISA) (2020). *Telecom Services Security Incidents 2019 Annual Analysis Report*.

Martins, L., R. Girao-Silva, L. Jorge, A. Gomes, F. Musumeci, and J. Rak (2017). Interdependence between power grids and communication networks: A resilience perspective. *DRCN 2017-Design of Reliable Communication Networks; 13th International Conference*, 1-9.

Kukliński, S., and L. Tomaszewski, (2019). Key Performance Indicators for 5G network slicing. *2019 IEEE Conference on Network Softwarization (NetSoft)*, 464-471.

ENEDIS (2020). *Description et étude des protections de découplage pour le raccordement des Installations de Production raccordées au Réseau Public de Distribution*. https://www.enedis.fr/sites/default/files/Enedis-NOI-RES_13E.pdf

5G EVE (2018). *Deliverable D1.1 Requirements Definition & Analysis from Participant Vertical Industries 5G European Validation platform for Extensive trials.* https://www.5g-eve.eu/wp-content/uploads/2018/11/5g-eve-d1.1-requirement-definition-analysis-from-participant-verticals.pdf

Qu, L., C. Assi, K. Shaban, and M. J. Khabbaz (2017). A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks. *IEEE Transactions on Network and Service Management 14*(3), 554-568.

Da Silva Coelho, W., A. Benhamiche, N. Perrot, and S. Secci (2020). On the impact of novel function mappings, sharing policies, and split settings in network slice design. *2020 16th International Conference on Network and Service Management (CNSM)*, 1-9.

Carlinet, Y., N. Perrot, and A. Alves-Tzitas (2019). Minimum-Cost Virtual Network Function Resilience. *INOC 2019.*

Leivadeas, A., G. Kesidis, M. Ibnkahla, and I. Lambadaris (2019). Vnf placement optimization at the edge and cloud. *Future Internet 11*(3), 69.

Bonet, P., C. M. Lladó, R. Puijaner, and W. J. Knottenbelt (2007). PIPE v2. 5: A Petri net tool for performance modelling. *23rd Latin American Conference on Informatics CLEI*, 50–62.

Rui, L., X. Chen, Z. Gao, W. Li, X. Qiu, and L. Meng (2020). Petri Net-Based Reliability Assessment and Migration Optimization Strategy of SFC. *IEEE Transactions on Network and Service Management 18(1)*, 167-181.

Di Mauro, M., M. Longo, F. Postiglione, and M. Tambasco (2017). Availability modeling and evaluation of a network service deployed via NFV. *International Tyrrhenian Workshop on Digital Communication*, 31-44.

Schneider, S., A. Sharma, H. Karl, and H. Wehrheim (2019). Specifying and analyzing virtual network services using queuing petri nets. *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 116-124.

Tola, B., Y. Jiang, and B.E. Helvik (2020). On the Resilience of the NFV-MANO: An Availability Model of a Cloud-native Architecture. *2020 16th International Conference on the Design of Reliable Communication Networks DRCN 2020*, 1-7.

Petri Net-Based Model for 5G and Beyond Networks Resilience Evaluation

Rui Li*†, Bertrand Decocq*, Anne Barros†, Yiping Fang†,Zhiguo Zeng†
*Orange Innovation
Châtillon, France
{rui.li, bertrand.decocq}@orange.com
†Laboratoire Génie Industriel
CentraleSupélec, Université Paris-Saclay
Gif-sur-Yvette, France
{rui.li, anne.barros, yiping.fang, zhiguo.zeng}@centralesupelec.fr

***Abstract*—The promise of telecommunication networks to deliver more demanding and complex applications requires them to become more flexible and efficient. To achieve better performance, telecommunication networks adopt technologies such as NFV (Network Function Virtualization). However, this evolution also brings more potential risks to the telecommunication network. Reliability and resilience are becoming critical for service delivery in the networks. To answer to service requirements of high level availability and reliability, a model with a global view of infrastructure, virtual network elements, and network layer structure is required. Toward this end, this paper presents a Petri Net method to model 5G and beyond telecommunication networks. We introduce an extended Petri Net to model physical infrastructure, virtual infrastructure, network services, their behaviors, and dependencies. We present a simulation result on network availability estimation. This result shows the potential of the Petri Net-based model to be applied to a complex telecommunication system resilience assessment.**

***Index Terms*—Petri Net, 5G networks, B5G, resilience, availability, modeling, simulation**

I. Introduction

Telecommunication networks are becoming indispensable for modern production and living. Facing the diverse and high requirements from a broad vertical industry, 5G and beyond networks are expected to be both efficient and reliable for service delivery. Keeping such systems at good performance during their whole life cycle is essential for service providers and operators. Although 5G has been under development for the last ten years, its resilience has not been studied enough.

5G is more service-oriented than 4G by offering a transition from a "horizontal" service delivery model toward a "vertical" one [1]. To better meet the different requirements and high demand, Network Function Virtualization (NFV) should be introduced for both RAN (Radio Access Network) and CN (Core Network) to better adjust the network configuration to the requirements. Thus, the resilience of the telecommunication networks is no longer an issue only for infrastructure but also for virtual elements and service delivery [2].

This study builds a Petri Net-based model to describe 5G and beyond networks. This model could be applied to communication service availability [3] (the ability to allow correct operation of the application) analysis, communication service reliability (the measure of continuous correct service delivery) estimation, and the resilience [4] (the ability to provide and maintain an acceptable level of service in the face of various faults and challenges) evaluation of the system.

While still at an early stage, this paper introduces a case study regarding network virtualization characteristics for testing Network Function Virtualization (NFV) self-healing in the dysfunctional mode and analyzing the network availability.

The paper is structured as follows. First, we present related work concerning telecommunication resilience analysis and modeling in section II. Then in section III, we focus on the telecommunication network model and, in particular explaining how an extended Petri Net models the virtualization characteristics. A case study on self-healing and the results are given in section IV. Section V concludes the work with some remarks and outlines the future works.

II. Related work

Recently, some research regarding the 5G and beyond telecommunication network resilience has been carried out. These studies mainly focus on network resilience optimization, only a few on resilience assessment. In optimization, while solving linear programming problems is still the mainstream of the research as found in [5]–[7], other methods such as Shortest Paths [8], Divide and Conquer [9] are also applied. The complexity of solving such a problem grows with the number of constraints. However, numerous new constraints on resource allocation and network management will be required if the virtualization layer is considered. In assessment, methods such as Reliability block diagram [10], Markov chain [11], [12] are addressed. These methods fail to model a complex telecommunication network by considering infrastructure, virtualization, network layers, and their dependencies. A model that can capture all network elements and their relationship for resilience assessment and optimization is still missing.

As a critical technique for the 5G and beyond, Network Virtualization has been particularly studied in some works. An availability model and analysis of a virtualized system based

on Virtual Machines(VMs) are introduced in [13]. The authors in [14] present a performance modeling approach that goes into the microservice level to estimate the effect of resource configuration on the Quality of Service (QoS). Although VMs are already widely used in virtualization, containerization, as a novel and lightweight virtualization method, is believed to be a promising solution for 5G and beyond. However, rarely is container-based virtualization modeled. Because it makes the telecommunication network modeling even more complex [2].

Some recent works draw attention to Petri Net-based model to study telecommunication networks thanks to its convenience in modeling discrete event systems. In [15], a Petri Net model is proposed for Service Function Chain (SFC) reliability assessment. However, the model does not take into consideration the risks of infrastructure failures. A Queuing Petri Net model is applied in [16] to evaluate the QoS of a video streaming service. The dysfunctional state of the system is not yet considered in their study. In [17], the authors apply a Petri Net model to describe the probabilistic behaviors of network service. In [18], an extension of Petri Net is applied only to model the NFV MANO framework to analyze its availability. Finally, the authors in [19], introduce a Petri Net-based performance model for containerized applications deployed by Kubernetes. However, these works are limited and do not propose a comprehensive perspective by considering NFV characteristics, infrastructure behaviors, and QoS in functional and dysfunctional mode.

Inspired by the related work, this paper proposes an extended Petri Net approach to fill the gaps in related work. This extended Petri Net is given to model SFC, containerization-based NFV elements, and infrastructure layer of the network. It is also capable to estimate the performance and the resilience.

III. Petri Net-based network model

A. Timed Stochastic Colored Petri Net

1) Petri Net mathematical representation: Petri Net is also known as Place / Transition net. It is a widely used technique tracking systems' states, dynamics, and constraints. As well defined in [20], the Petri Net is a 5-tuple $\mathcal{N} = \langle P, T, F, W, M_0 \rangle$, where P is a finite set of places often representing the different states of a system. Places are graphically presented in circles. T is a finite set of transitions representing the state-changing process. Transitions are graphically presented in rectangles or squares. F is a finite set of arcs with $F \subseteq (P \times T) \cup (T \times P)$. W is a multi-set of arcs $(P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ assigning the weight to inputs and outputs of a transition. M is the marking of the Petri Net graph and $M_0 = P \rightarrow \{m_1, m_2, ..., m_{|P|}\}$, therefore, assigning the initial marking of the graph. Tokens of the graph describe the dynamic and concurrent activities of systems. The marking in Petri Net records the token number of each place.

2) Extensions of Petri Net: The classical Petri Net is not directly applicable to telecommunication systems. Some state-changing processes in such a system could be stochastic, time-dependent, and require additional information. Some extensions of Petri Net can help modeling a complex network better.

One of the most important extensions of Petri Net is Stochastic Petri Net [21]. It includes a new set $R = \{r_1, r_2, ..., r_{|T|}\}$, representing the firing rate of each transition. This extension could be applied to describe a failure process in the telecommunication network.

In order to describe a time-dependent process, for instance, the packet transmission, Timed Petri Nets [22] are introduced. A new set $D : T \rightarrow \mathbb{Q}_0^+$ associates each transition with a specific non-negative number to represent the time factor.

Colored Petri Net attaches a value to a token. It indeed distinguishes different kinds of tokens that a place holds. This extension adds the following items [23], [24]:

1) Σ is a finite set of non-empty types, called color sets.
2) C is a function $P \rightarrow \Sigma$ defining the type of tokens allowed in a place.
3) G: $T \rightarrow \mathbb{B}$ associate the transition with a precondition g (Boolean expression). The transition will be fired only when g returns true value.
4) E is an arc expression function defined from F into expressions such that $\forall a \in F : C(E(a)) = C(p)$.
5) I is an initialization function mapping place $p \in P$ with an expression such that $I(p)$ is associated to $C(p)$.

Combining the extensions as aforementioned, we use a Timed Stochastic Colored Petri Net (TSCPN) to describe the 5G system. Such a TSCPN is a multi-tuple: $TSCPN = \langle \Sigma, P, T, F, W, m_0, C, G, E, I, R, D \rangle$

TSCPN is then applied to describe the different parts of the telecommunication system.

B. Composition of the network system

We divide the 5G and beyond networks into three layers.

The first layer is the service layer. In this layer, the network service is delivered by steering packets between a set of functions called the service function chain. We consider a network service where all its network functions engaged are virtualized. The service delivery is presented by a series of Virtual Network Functions(VNFs) connected via virtual links.

The second layer is the NFV elements layer. A virtual link in this layer is based on a physical transport network, and a VNF is a virtual functional building block hosted on a physical server. Unlike physical elements, virtual elements may have an unfixed size and an unfixed number of replicas.

The third layer is the infrastructure layer. Physical machines and physical links belong to this layer.

C. Basic hypothesis

1) Service function chain: We consider an End-to-End service with an ordered SFC. In this system, user equipment sends service request packets to the SFC in the network. We assume that every packet conveys a same size of data, and its SFC always follows the same order of VNFs.

An SFC is a series of VNFs connected by links. The transport network is considered as a perfectly reliable system. We only consider a fixed time delay spent on the transmission link between a user equipment and VNF, and between different VNFs.

2) VNF and Virtualization: A VNF is, in fact, an application that consists of several microservices. An example of 3 VNFs and their microservices are shown in Fig. 1. Each microservice is considered as a sub-function of VNF. We assume that if a request packet needs multiple microservices, they should be pursued in a given order. A packet cannot consult two different microservices at the same time.

Fig. 1. VNFs and their microservices.

There are many ways of virtualization. In this paper, we choose to model the deployment of these microservices in containers. Kubernetes is used as the system for automating deployment and managing containerized applications.

Pods are the smallest deployable units in Kubernetes. A pod is one or a cluster of containers with shared storage and network resources. We assume that only one container is deployed on a pod. For each container, it corresponds to a microservice the VNF supplier predefines. Pods are running on Kubernetes nodes. All these nodes are physical machines.

3) Infrastructure and resources: The infrastructures used to deliver an end-to-end function are physical links and physical servers. Each physical machine has a certain amount of CPU, storage, and network resources. Pods can only be hosted on the server with enough resources.

4) Orchestration and management: Kubernetes is an enabler for the orchestration and management of containerized applications. To evaluate the system resilience under failure, we consider self-healing operation. Kubernetes can regularly detect the healthiness of the pods or nodes. In case of failure, they will be terminated, and new ones will be created. Other operations such as auto-scaling, in which Kubernetes detects particular indicators and change the deployment manners accordingly, will be studied in future work.

D. Telecommunication network modeling

1) 5G Service Function Chain: This is the top layer Petri Net which represents the process of an SFC containing m VNFs as presented in Fig. 2. This pipeline style Petri Net consists of a set P of $2(m+1)$ places and a set T of $2m+1$ transitions. It signifies the progress of packet processing.

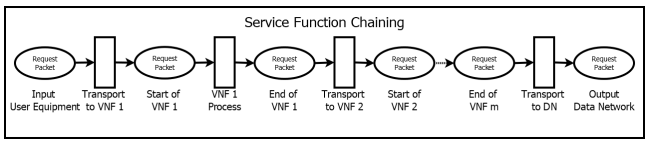


Fig. 2. Petri Net of SFC example.

At this level, a token refers to a packet that conveys a request message that needs to traverse the SFC. The color function C allows only 'packet' type tokens to stay at the places, $C = \{\text{'Packet'}\}$. These 'packet' tokens can also convey some values, including packet serial number, latency requirement, packet starting time, etc.

$P_{\text{sfc}} = \{p_{\text{UE}}, p_{\text{Start of VNF1}}, p_{\text{End of VNF1}}, \ldots, p_{\text{End of VNFm}}, p_{\text{DN}}\}$ represents different steps of the processing procedure. 'Packet' tokens at the place p_{DN} signify that the packets are successfully delivered. 'Packet' token information such as latency can be further investigated to verify if service is delivered correctly.

The transition set $T_{\text{sfc}} = \{T_{\text{tran}} \cup T_{\text{treat}}\}$ contains the transport and the packet treatment in SFC. The transitions $T_{\text{tran}} = \{t_{\text{Transport to VNF 1}}, t_{\text{Transport to VNF 2}}, \ldots, t_{\text{Transport to DN}}\}$ stand for the packets being transmitted from the previous place to the next one. A duration function D is attached to these transitions and returns a time delay for each packet transmission according to the distance and the transmission technology. The treatment transitions $T_{\text{treat}} = \{t_{\text{VNF 1 process}}, t_{\text{VNF 2 process}}, \ldots, t_{\text{VNF m process}}\}$ are expanded into sets of sub-networks explained in the following sections.

2) Virtual Network Functions: We expand the VNF i process transition $t_{\text{VNF i process}}$ as depicted in Fig. 3. This transition takes a token from the place $p_{\text{Start of VNF i}}$, and after processing, it returns the token to the place $p_{\text{End of VNF i}}$. Inside a VNF i, there are n microservices embedded in containers. The token color set is $\Sigma = \{\text{'Packet', 'Packet list'}\}$. The latter refers to a list of packets to be treated. When a packet gets into this i-th VNF, an immediate transition $t_{\text{service selection}}$ routes them to the microservice k it looks for with the help of guard function G. Then this packet is inserted to the 'Packet list' token at the place $p_{\text{MS queue k}}$. Microservice k transition $t_{\text{MS k-transition}}$ is enabled as long as there is at least one microservice pod with enough capacity left to treat this packet. After the treatment, the packet arrives at $p_{\text{End of MS in VNF i}}$. Two assertion transitions associated with guard functions will check if other microservices should be consulted before leaving current VNF.

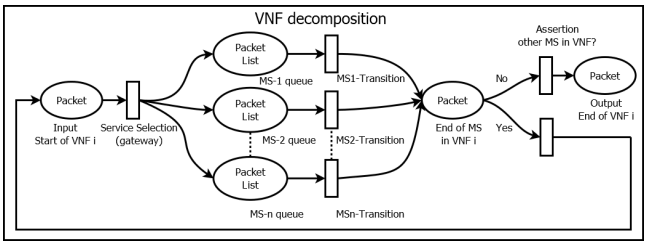


Fig. 3. Petri Net of VNF decomposition.

3) Microservices: Microservices level Petri Net explains the transition $t_{\text{MS k-transition}}$ in detail. The microservices applications are in the form of containers and they are embedded into pods. A new token color type, 'Pod', is added. These tokens are stored at $p_{\text{available pods}}$ and $p_{\text{failed pods}}$.

The microservice is modeled with two transitions, as shown in Fig. 4. The transition $t_{\text{MS bounding}}$ couples the first packet of the packet list token with an available pod. A Boolean guard function B associated with this transition checks if the pod is eligible to provide service for the packet. To complete the

task, the packet will borrow a certain computation resources from the pod. These resources are seen as homogeneous by assumption. A function D assigns the processing time of the microservice to this transition. An output packet token will be sent to $p_{\text{bounded packets}}$ after this duration. The transition $t_{\text{MS process}}$ verifies the completion of the service. If a pod fails during the packet treatment, the packet needs to redo the same microservice. Otherwise, the borrowed resource will be released, and the packet will try the next microservice.

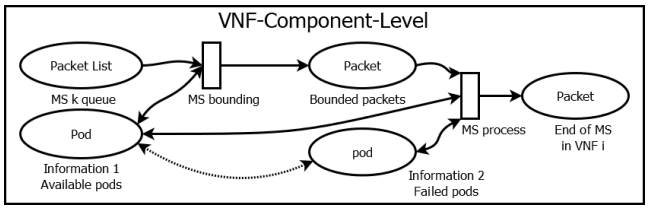


Fig. 4. Petri Net of Mirco-service treatment.

4) Failure and Self-healing: Numerous failures could happen in a telecommunication system. In this study, we mainly consider pod failure, one of the most common failures that affect the a vitrulized system's performance.

Due to space, only pod software failure is explained, as shown in Fig. 5. A stochastic transition $t_{\text{Stochastic pod failing process}}$ connects the state change of a pod token. We assume that all pods in our system are identical, and thus, they have the same mean time to failure (MTTF). We also assume that a pod failure is in accordance with the exponential distribution $X \sim \text{Exp}(\lambda)$, and with a constant rate of $\lambda = \text{MTTF}^{-1}$.

Kubernetes launches a liveness probe once in a while to detect the healthiness of pods. This time interval is called periodsecond of the probe. If a pod is unhealthy, Kubernetes starts the self-healing by terminating the pod and creating new one. The transition $t_{\text{Pods termination}}$ consumes the failed token after a graceful termination time. Transition $t_{\text{Pods creation}}$ will create a pod containing the same microservice on an available node with enough resources. We introduce place $p_{\text{Available node}}$ containing a new token color, 'Node', to represent the nodes.

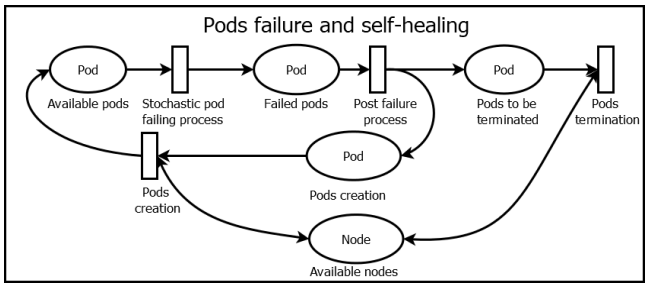


Fig. 5. Pod Self-healing process

IV. Networks availability estimation

The first step of our work is to apply the model to estimate the system resilience by looking at the virtualization and infrastructure layer without mapping them to telecommunication services. By doing so, the availability of the network to provide services to the packets is estimated. Two major failures, the physical failure on nodes and the software failure on pods, are identified as the main risks to the system. When a failure occurs, the Kubernetes Master will do self-healing to ensure availability. We consider a system with one VNF that consists of two microservices. We assume that these microservices container pods are deployed on the same Data Center. For load balancing reasons, each microservice initially has three identical pod replicas. Other parameters are given in Table. I.

TABLE I
VNF parameters

Parameter	Value
Pod failure rate	MTTF = 1258 hours [25]
Pod termination time	30 seconds (fixed value)
Node failure rate	MTTF = 8760 hours
Node repair rate	MTTR = 0.5 hours
Average time for pod instantiating	5 seconds
Average time for node creation	1 second
Node capacity	3 pods per node
Data Center Capacity	4 servers
Self-healing probe periodsecond	0 (immediate), 2, 5 and 10 seconds

A microservice is considered available at time t, if the token quantity of such microservice at $p_{\text{available pods}}$ is greater than the desired replica quantity. The uptime of a microservice is the duration of time that a microservice is available. Then the average availability of a microservice i can be calculated as:

$$A_i = \frac{\text{microservice } i \text{ uptime}}{\text{total simulation time}}$$

In the first situation, we assume the self-healing detection is immediate, i.e., a failure on pod or node can be detected with no delay. We simulate the microservice behavior over 50 years. The average value of microservices' uptime over 20000 simulations is taken as the final result. We assume that the two microservices are from the same VNF supplier and are managed by the same Kubernetes Master. The results in Fig. 6 show that if the desired replica quantity is three, then the availability of a single microservice is 99.9996712% (5 nines). If the desired replica quantity is one (one pod is enough, but three initial pods bring high redundancy), then the availability of this microservice can achieve up to 9 nines. The overall availability for the VNF (at least three available replicas for both microservice 1 and 2) is 99.9993523%[1].

In the second situation, the effect of self-healing probe frequency t_p on system availability is studied. The result is shown in Fig. 7. We compare the overall VNF availability for t_p varying from 0 to 10 seconds. The longer the probe periodsecond, the lower the overall availability. The availability drops from 99.9993523% (5 nines) to 99.9987198%[2] (4 nines) by changing immediate detection to 10 seconds. Thus, the telecommunication network can consume less energy while satisfying the availability requirement by wisely optimizing the periodsecond if allowed, according to this result.

[1]95% confidence interval [99.9993520304%, 99.9993526071%]
[2]95% confidence interval [99.9987192945%, 99.9987202301%]

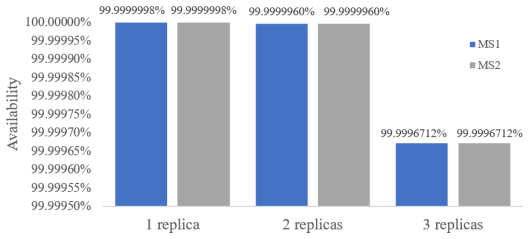


Fig. 6. Microservice availability with immediate detection.

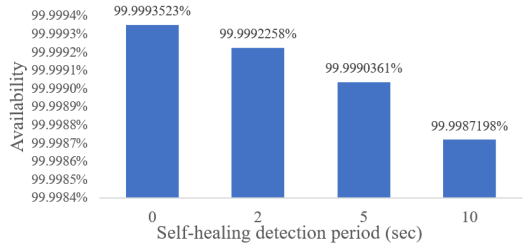


Fig. 7. Overall availability under immediate, 2, 5, and 10 seconds detection.

After 20000 simulation iterations, the results converge well. It took from half to two hours (depending on detection interval) to run these 20000 simulations in CPN tools on a personal computer equipped with Windows 10, 2.10 GHz CPU, and 8GB memory. Indeed, the computation time is proportional to the number of pods and inversely proportional to the periodsecond.

V. Conclusion

This paper presents a Petri Net-based model to analyze the performance and resilience of 5G and beyond networks. This model divides a telecommunication system into multiple layers and proves its ability to describe new features of 5G and beyond. The results of the Monte-Carlo simulation on VNF self-healing show the prospects of this model on telecommunication network availability analysis.

The results remain optimistic since other risks such as network failure or maintenance are not fully considered. In addition, more precise parameters need to be collected from our experts and suppliers. For the next step, the auto-scaling case study will be carried out to complete the service-level reliability and resilience analysis, and to see how networks adapt to different packet traffic. We also intend to expand the case study from one single VNF to an SFC and apply the model to simulate a real use case from the verticals.

References

[1] A. Banchs, D. M. Gutierrez-Estevez, M. Fuentes, M. Boldi and S. Provvedi, “A 5G mobile network architecture to support vertical industries,” in IEEE Communications Magazine, vol. 57, no. 12, pp. 38–44, December 2019.

[2] R. Li, B. Decocq, A. Barros, Y. Fang, and Z. Zeng, “Complexity in 5G Network Applications and use cases,” In 31st European Safety and Reliability Conference, pp. 3054–3061, 2021.

[3] B. Sayrac, “5G Common Terminology,” 5G Smart Manufacturing, 2020, [Online] Available: https://5gsmart.eu/wp-content/uploads/5G-SMART-common-terminology.pdf.

[4] ResiliNets Wiki, [Online] Available: https://resilinets.org.

[5] T. Hößler, L. Scheuvens, N. Franchi, M. Simsek and G. P. Fettweis, “Applying reliability theory for future wireless communication networks,” 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pp. 1-7, 2017.

[6] K. Sayad, B. Lemoine, A. Barros, Y. Fang, and Z. Zeng, “Dynamic orchestration of communication resources deployment for resilient coordination in critical infrastructures network,” In 31st European Safety and Reliability Conference, pp. 2055–2062, 2021.

[7] D. Harutyunyan, N. Shahriar, R. Boutaba and R. Riggio, “Latency-Aware Service Function Chain Placement in 5G Mobile Networks,” 2019 IEEE Conference on Network Softwarization, pp. 133–141, 2019.

[8] L. Qu, C. Assi, K. Shaban, and M.J. Khabbaz, “A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks,” IEEE Transactions on Network and Service Management, vol. 14, no. 3, pp. 554–568, 2017.

[9] Y. Carlinet, N. Perrot and A. Alves-Tzitas, “Minimum-cost virtual network function resilience,” in INOC, pp. 37–42, 2019.

[10] M. Di Mauro, G. Galatro, M. Longo, F. Postiglione and M. Tambasco, “Comparative performability assessment of SFCs: the case of containerized IP multimedia subsystem,” in IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 258–272, March 2021.

[11] M. Mosahebfard, J. Vardakas and C. Verikoukis, “Modelling the admission ratio in NFV-based converged optical-wireless 5G networks,” in IEEE Transactions on Vehicular Technology, vol. 70, no. 11, pp. 12024-12038, 2021.

[12] H. Farooq, M. S. Parwez and A. Imran, “Continuous time Markov chain based reliability analysis for future cellular networks”, Proc. IEEE Glob. Commun. Conf. (GLOBECOM), pp. 1–6, 2015.

[13] D.S. Kim, F. Machida, and K.S. Trivedi, “Availability modeling and analysis of a virtualized system,” in 2009 15th IEEE Pacific Rim International Symposium on Dependable Computing, pp. 365–371, 2009.

[14] M.G. Khan, J. Taheri, M.A. Khoshkhologhi, A. Kassler, C. Cartwright, M. Darula, et al., “A performance modelling approach for SLA-aware resource recommendation in cloud native network functions,” in : 6th IEEE Conference on Network Softwarization, pp. 292–300, 2020.

[15] L. Rui, X. Chen, Z. Gao, W. Li, X. Qiu and L. Meng, “Petri Net-based reliability assessment and migration optimization strategy of SFC,” in IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 167–181, March 2021.

[16] S. Schneider, A. Sharma, H. Karl and H. Wehrheim, “Specifying and analyzing virtual network services using queuing Petri Nets,” 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 116–124, 2019.

[17] M. Di Mauro, M. Longo, F.Postiglione and M. Tambasco, “Availability modeling and evaluation of a network service deployed via NFV,” in International Tyrrhenian Workshop on Digital Communication, Springer, Cham, pp. 31–44, 2017.

[18] B. Tola, Y. Jiang and B. E. Helvik, “On the resilience of the NFV-MANO: an availability model of a cloud-native architecture,” 2020 16th International Conference on the Design of Reliable Communication Networks (DRCN) , pp. 1–7, 2020.

[19] V. Medel, O. Rana, J. A. Bañares, and U. Arronategui, “Modelling performance and resource management in kubernetes,” in Proceedings of the 9th International Conference on Utility and Cloud Computing, Association for Computing Machinery, New York, pp. 257–26, 2016.

[20] T. Murata, “Petri Nets: Properties, analysis and applications,” Proceedings of the IEEE, vol. 77, no. 4, pp. 541-580, April 1989.

[21] M. Marsan, G. Conte, G. Balbo, “A class of generalized stochastic Petri Nets for the performance evaluation of multiprocessor systems,” ACM Trans. Comput. Syst, vol, 2. pp. 93-122, 1984.

[22] L. POPOVA-ZEUGMANN, “Time petri nets,” Time and Petri Nets, Springer, Berlin, Heidelberg, pp. 31–137, 2013.

[23] K. Jensen, “9,” Volume 1, Basic Concepts. Springer, Berlin, 1992.

[24] D. Liu, J. Wang, S. Chan, J. Sun, L. Zhang, “Modeling workflow processes with colored Petri Nets,” Computers in Industry, vol. 49. pp. 267–281, 2002.

[25] S. Sebastio, R. Ghosh, A. Gupta and T. Mukherjee, “ContAv: a Tool to assess availability of container-based systems,” 2018 IEEE 11th Conference on Service-Oriented Computing and Applications (SOCA), pp. 25–32, 2018.

Modélisation d'un réseau 5G par des réseaux de Pétri pour estimer sa résilience

Rui Li[1,2] et Bertrand Decocq[1] et Anne Barros[2] et Yiping Fang[2] et Zhiguo Zeng[2]

[1]*Orange Innovation, Châtillon, France*
[2]*Laboratoire Génie Industriel, CentraleSupélec, Université Paris-Saclay, Gif-sur-Yvette, France*

Avec l'évolution des technologies des télécommunications, les réseaux 5G sont censés, par rapport aux générations précédentes, répondre d'une part à des besoins d'échange de données plus importants des utilisateurs finaux, d'autre part à des nouveaux besoins liés au développement de services exigeants en termes de latence et de fiabilité ou au large déploiement d'objets connectés. Cependant, cette évolution de réseau entraîne davantage de risques potentiels pour le réseau 5G et elle entraîne également une complexité croissante de la gestion du réseau. De ce fait, la fiabilité et la résilience sont essentielles pour les opérateurs afin d'assurer le lancement des services. Pour évaluer la résilience, il est nécessaire de comprendre la structure de la 5G et de construire un modèle basé sur celle-ci. Cet article aborde ce problème en proposant un modèle basé sur un réseau de Pétri, un outil bien connu pour la modélisation des systèmes. Nous introduisons un réseau de Pétri étendu pour modéliser l'infrastructure physique, l'infrastructure virtuelle, les services de réseau, leurs comportements et leurs dépendances. Ce modèle pourrait être appliqué à l'analyse de la disponibilité des services de communication, à l'estimation de la fiabilité des services de communication et à la résilience. Nous présentons dans un premier temps le résultat sur l'estimation de la disponibilité du réseau vis-à-vis de la défaillance des éléments du réseau. Ce résultat montre le potentiel du modèle à être appliqué à l'évaluation de la résilience d'un système de télécommunication.

Mots-clefs : Petri Net, 5G networks, resilience, availability, modeling, simulation

1 Introduction

Telecommunication networks are becoming indispensable for modern production and living. Facing the diverse and high requirements from a broad vertical industry, 5G and beyond networks are expected to be both efficient and reliable. Keeping such systems at good performance during their whole life cycle is essential for service providers and operators. To meet the different requirements and high demand, Network Function Virtualization (NFV) should be introduced for both RAN (Radio Access Network) and CN (Core Network) to better adjust the network configuration to the requirements. Thus, the resilience of the telecommunication networks is no longer an issue only for physical infrastructure but also for virtual elements and service delivery [ASVW17]. This study builds a Petri Net-based model to describe 5G networks. This model could be applied to communication service availability analysis, service reliability estimation, and network resilience evaluation [Say20], and it will be helpful for operators to deploy vertical 5G services.

2 Petri Net-based network model

2.1 Timed Stochastic Colored Petri Net

Petri Net is a widely used technique for tracking systems' states, dynamics, and constraints. It can also be used to describe a complex system like 5G. As defined in [Mur89], Petri Net is a 5-tuple $\mathcal{N} = \langle P, T, F, W, M_0 \rangle$. P is a finite set of places representing different states of a system. Places are graphically presented in circles. T is a finite set of transitions representing the state-changing process. Transitions are presented in rectangles. F is a finite set of arcs with $F \subseteq (P \times T) \cup (T \times P)$. W is a multi-set of arcs $F \rightarrow \mathbb{N}$

assigning the weight to inputs and outputs of a transition. $M_0 = P \rightarrow \{m_1, m_2, ..., m_{|P|}\}$ assigns the initial marking of the graph. Tokens of the graph describe the dynamic and concurrent activities of systems.

The classical Petri Net is not directly applicable to telecommunication systems. We need to introduce some extensions. A Stochastic Petri Net includes a new set $R = \{r_1, r_2, ..., r_{|T|}\}$, representing the firing rate of each transition. This extension could be applied to describe a failure process. In order to describe a time-dependent process, for instance, the packet transmission, Timed Petri Nets are introduced. A new set $D : T \rightarrow \mathbb{Q}_0^+$ associates each transition with a specific non-negative number to represent the time factor. Colored Petri Net [Jen92] attaches a value to tokens to distinguish them. This extension adds the following items: Σ, a finite color set; C, a function defining the type of tokens allowed in a place; G: a function associating a transition with a Boolean expression; E, an arc expression ; I, an initialization function.

Combining the extensions above, we use a Timed Stochastic Colored Petri Net (TSCPN) to describe the 5G system. Such a TSCPN is a multi-tuple: $TSCPN = \langle \Sigma, P, T, F, W, m_0, C, G, E, I, R, D \rangle$.

2.2 Composition of the telecommunication network system

We divide a 5G network into three layers (network service, virtualization, and physical infrastructure).

Network service: We consider an End-to-End service with an ordered Service Function Chain (SFC). In this system, the user equipment sends service request packets to the SFC. We assume every packet conveys the same data size, and its SFC always follows the same order of VNFs. An SFC defines a series of VNFs connected by links. The transport network is considered a perfectly reliable system (with redundant paths).

VNF and virtualization: A VNF is an application consisting of several microservices. Each microservice is considered a sub-function. In this paper, we adopt container-based virtualization. We assume that Kubernetes is used to automate deployment and manage containerized applications in 5G NFV. Pods are the smallest deployable units. One pod is composed of one container, and it corresponds to one replica of the microservice. Pods are running on Kubernetes nodes in the form of physical machines.

Infrastructure and resources: The infrastructures used to deliver an end-to-end function are physical links and physical nodes. Each node (physical machine) has a certain amount of CPU, storage, and network resources. Pods can only be hosted on a node with enough resources.

In this virtualized network, Kubernetes is an enabler for the orchestration and management to keep its normal operation. To evaluate the system resilience under failure, we consider self-healing operation [Chu20]. Kubernetes can regularly detect the healthiness of the pods or nodes. In case of failure, they will be terminated, and new ones will be created. Other operations such as auto-scaling, in which Kubernetes detects particular indicators and changes the deployment manners accordingly, will be studied in future work.

2.3 Telecommunication network modeling

The top layer Petri Net, as presented in Fig. 1 depicts the packet processing in an SFC with m VNFs.

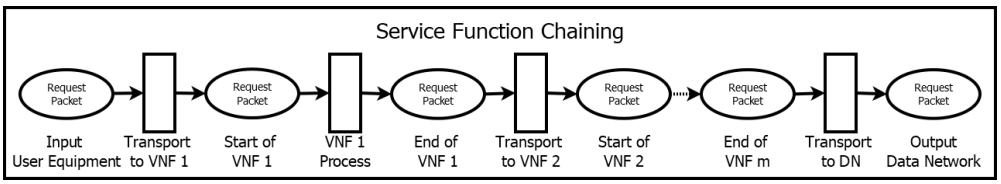


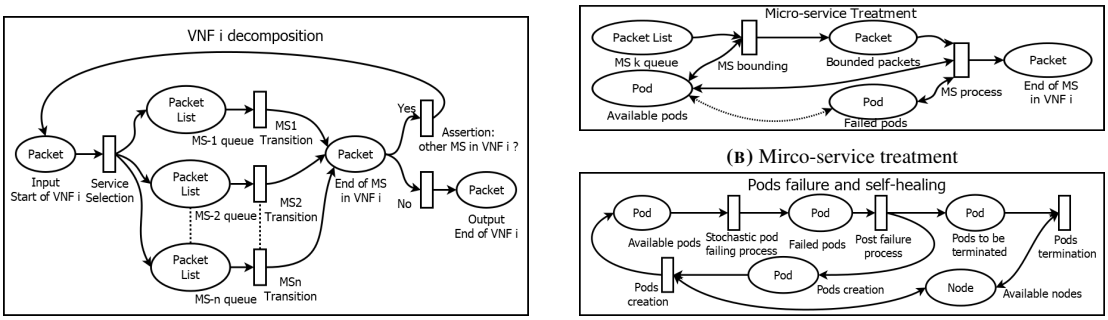
FIGURE 1: Petri Net of SFC example.

The VNF i process transition $t_{\text{VNF i process}}$ is expanded in Fig. 2a. Inside a VNF i, there are n microservices embedded in containers. When a packet gets into this i-th VNF, an immediate transition $t_{\text{service selection}}$ routes them to the microservice k it looks for. Then this packet is inserted into a packet list at the place $p_{\text{MS queue k}}$ waiting to be processed. Microservice k transition $t_{\text{MS k-transition}}$ is enabled as long as there is at least one microservice pod with enough capacity. After finishing the task, the packet will be processed at next microservice in the current VNF or in the next VNF.

Microservices level Petri Net explains the transition $t_{\text{MS k-transition}}$ in detail, as shown in Fig. 2b. The transition $t_{\text{MS bounding}}$ couples the first packet of the packet list with an available pod. To complete the task,

the packet borrows certain computation resources. An output packet token will be sent to $p_{\text{bounded packets}}$ after a certain processing duration. Finally, the transition $t_{\text{MS process}}$ verifies the completion of the service.

Failures can also be described by a Petri Net. In this paper, we explain a pod failure process. This is one of the most common failures that affect the virtualized system's performance. As shown in Fig. 2c, a stochastic transition $t_{\text{Stochastic pod failing process}}$ connects the state change of a pod token. Kubernetes launches a liveness probe once in a while to detect the healthiness of pods. This time interval is called periodsecond of the probe. If a pod is unhealthy, Kubernetes starts self-healing. The transition $t_{\text{Pods termination}}$ consumes the failed token after a graceful termination time. Transition $t_{\text{Pods creation}}$ will create a pod containing the same microservice on an available node with enough resources.



(A) VNF decomposition

(B) Mirco-service treatment

(C) Pod Self-healing process

FIGURE 2: Different levels of Petri Net-based model

3 Network availability estimation

The first step of our work is to apply the model to estimate the system resilience by looking at the virtualization and infrastructure layer without mapping them to telecommunication services. By doing so, the availability of the network to provide services is estimated. The failures of nodes and pods are identified as the main risks to the system. When a failure occurs, Kubernetes will launch self-healing. We consider a system with one VNF that consists of two microservices. We assume that these microservices container pods are deployed on the same Data Center with four physical nodes (servers). For load-balancing reasons, each microservice initially has three identical pod replicas. Other parameters are given in Table 1.

TABLE 1: VNF parameters

Parameter	Value
Pod failure rate	MTTF = 1258 hours
Pod termination time	30 seconds (fixed value)
Average time for pod instantiating	5 seconds
Node(server) failure rate	MTTF = 8760 hours
Node(server) repair rate	MTTR = 0.5 hours
Average time for node creation	1 second
Node(server) capacity	3 pods per node seconds

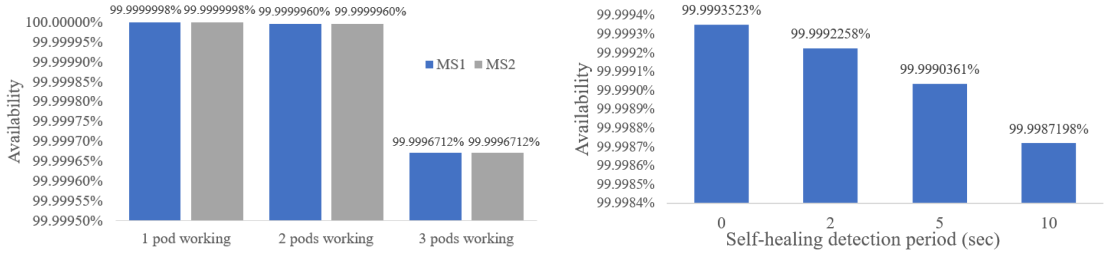
A microservice is considered available at time t if the number of pods is greater than the desired replica quantity. The uptime of a microservice is the duration of time that a microservice is available. Then the average availability of a microservice i can be calculated as:

$$A_i = \frac{\text{microservice } i \text{ uptime}}{\text{total simulation time}}$$

In the first situation, we assume the self-healing detection is immediate, i.e., a failure on a pod or node can be detected without delay. We assume that the two microservices are from the same VNF supplier and

are managed by the same Kubernetes Master. We simulate the microservice behavior over 50 years to get a high probability of failure in each iteration. It takes about two hours to run 20000 simulations (to get a higher confidence level) with CPN Tools. We take the average value of microservices' uptime as the final result. The results in Fig. 3a show that if the desired replica quantity is three pods, then the availability of a single microservice is 99.9996712% (5 nines). If the desired replica quantity is one pod (one pod is enough, but three initial pods only for redundancy reason), then the availability of this microservice achieves 9 nines (some URLLc services may require up to 8 nines according to 3GPP Release 16). The overall availability for the VNF (at least three available replicas for both microservice 1 and 2) is 99.9993523%.

In the second situation, the effect of self-healing probe frequency t_p on system availability is studied. The result is shown in Fig. 3b. We compare the overall VNF availability for t_p varying from 0 to 10 seconds. The longer the probe periodsecond, the lower the overall availability. The availability drops from 99.9993523% (5 nines) to 99.9987198% (4 nines) by changing immediate detection to 10 seconds. Thus, according to this result, the telecommunication network can consume less energy while satisfying the availability requirement by wisely optimizing the periodsecond if allowed.



(A) Microservice availability with immediate detection.

(B) Overall availability under 0, 2, 5, and 10 seconds detection.

FIGURE 3: Simulation results

4 Conclusion

This paper presents a Petri Net-based model to analyze the performance and resilience of 5G networks. This model divides a 5G system into multiple layers and proves its ability to describe new features of 5G. The simulation results on self-healing show the prospects of this model on telecommunication network availability analysis. The results remain optimistic since other risks, such as network failure or maintenance are not fully considered. In addition, more precise parameters need to be collected from experts. For the next step, the auto-scaling use case will be carried out to complete the service-level reliability and resilience analysis and see how networks adapt to different traffic changes. We also intend to expand the use case from one single VNF to an SFC and apply the model to simulate a real use case from the verticals.

References

[ASVW17] G. Arfaoui, J. M. Sanchez Vilchez, and J. Wary. Security and resilience in 5g: Current challenges and future directions. In *2017 IEEE Trustcom/BigDataSE/ICESS*, pages 1010–1015, 2017.

[Chu20] O. Chunikhin. Reliable, self-healing kubernetes explained. available at `https://kublr.com/blog/reliable-self-healing-kubernetes-explained/`, April 2020.

[Jen92] K. Jensen. Formal definition of coloured petri nets. In *Coloured Petri Nets: basic concepts,analysis methods and practical use*, volume 1, pages 65–87. Springer, Berlin, 1992.

[Mur89] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[Say20] B. Sayrac. 5g common terminology. available at `https://5gsmart.eu/wp-content/uploads/5G-SMART-common-terminology.pdf`, June 2020.

A Petri Net-based model to study the impact of traffic changes on 5G network resilience

Rui Li, Bertrand Decocq

Orange Innovation, Orange Labs, France. E-mail: {rui.li;decocq.bertrand}@orange.com

Yiping Fang, Zhiguo Zeng, Anne Barros

Chair on Risk and Resilience of Complex Systems, Laboratoire Génie Industriel, Centralesupélec, Université Paris-Saclay, France. E-mail: {yiping.fang;zhiguo.zeng;anne.barros}@centralesupelec.fr

The advent of 5G has enabled a wide variety of devices to access the network. With the digitization of industry, more and more vertical services, such as smart cities, remote health, and autonomous driving, rely on 5G networks for communication. These verticals bring new challenges to the telecommunication network resilience. Among them, sudden traffic change seems to be a critical challenge that impacts the resilience performance of 5G networks. This paper presents a network model for future 5G infrastructures based on Petri nets by taking into consideration the particularities of network virtualization and softwarization. This work also seeks to analyze the effectiveness of microservice-level autoscaling and network isolation by using discrete event simulation. The results suggest that both autoscaling and network isolation could increase network resilience when network traffic changes abruptly.

Keywords: 5G, Resilience, Quality of Service, telecommunication network, Kubernetes, Petri Net, discrete event simulation, complex system.

Acronyms

CN	Core Network
CU	Centralized Unit
DU	Distributed Unit
MS	Micro Service
NFV	Network Function Virtualization
RAN	Radio Access Network
SDN	Software-Defined Networking
SFC	Service Function Chaining
UPF	User Plane Function
VNF	Virtual Network Function

1. Introduction

Telecommunication systems and infrastructures keep continuously evolving in order to meet the growing needs of private and business users. By adopting technologies such as NFV (Network Function Virtualization), and SDN (Software-Defined Networking), 5G can meet the various requirements from end-users. However, 5G networks are under numerous challenges as well. As defined by Sterbenz et al. (2010), network resilience is the ability of the network to provide and maintain an acceptable level of service in the face of various faults and challenges to normal operation. It has drawn a lot of attention in the field of 5G networks. One of the most common challenges that a 5G network may encounter would be traffic variation. Traffic variation may be due to user equipment's behaviors, malicious attacks, or other issues. In the event of sudden traffic increases, large amounts of packets sent to the network will congest the network functions and saturate the telecommunication system.

With more and more objects connecting to the internet, dealing with the variant traffic to better adjust the network to the load becomes a critical issue for 5G. The good news is that by adopting NFV and SDN, the 5G system can change its scalability. Scaling can help 5G networks tackle this traffic variation issue. When traffic increases, some overloaded parts will be scaled out by creating more instances to share the load to avoid congestion. When the traffic decreases, the unnecessary instances are scaled in to free up unnecessary resources for other usages.

Some research has started the study of the scaling impact on 5G performance. Alawe et al. (2018), Rahman et al. (2018) and Subramanya and Riggio (2021) modeled the scaling problem

as a time series forecasting problem that predicts the future number of VNF instances in response to dynamic traffic changes. Rotter and Van Do (2021) proposed a queuing model for a scenario where a threshold-based algorithm controls the number of UPF (User Plane Function) instances depending on users' traffic. However, none of these works analyzes the impacts on resilience. The second drawback of these works is that the impact of the traffic variation over a short time interval is neglected. 5G is believed to deliver services to almost every vertical industry, including the services sensitive to latency or services that require very high reliability (3GPP (2020)). Even a second-level performance loss will significantly violate the service level agreement. Another limitation is the lack of consideration of risk propagation. The congestion brought by traffic change can easily propagate from one part of the network to another if not well isolated, degrading 5G network resilience.

The objectives of this work are to study the traffic variation impact on 5G network resilience performance in a short time interval and analyze the effectiveness of network isolation on congestion propagation. However, 5G is a complex system, especially in terms of management and orchestration (Nencioni et al. (2018)). To estimate the resilience of 5G networks, we need to build a comprehensive model that comprises the processing of a network service packet, the life cycle of network elements, etc.

In this work, we developed a Petri Net-based model for 5G networks whose network functions are managed by a Kubernetes management and orchestration system. By carrying out discrete event simulation, we show that our model is capable of evaluating the network service performance and resilience under different traffic patterns. The main contributions of this work are the following:

- A Petri Net based-model considering the virtualization of 5G and the transmission of user plane packets from an end-to-end point of view
- The modeling of microservice-level autoscaling mechanism
- The investigation of network congestion and its propagation using the discrete event simulation
- The estimation of network service latency and the acceptance rate under traffic change

The paper has been organized in the following way. We briefly introduce the virtualized telecommunication networks in section 2. In section 3, we present the Petri Net-based model. Two use cases on autoscaling and the simulation results are given in section 4. Finally, section 5 concludes the work with some remarks and outlines the future works.

2. The virtualized telecommunication system

To deliver an End-to-End service, 5G networks need to steer the traffic through a set of VNFs (Virtual Network Functions) distributed in RAN (Radio Access Network) and CN (Core Network), called SFC (Service Function Chaining). In the previous generations, these functions were implemented in the form of physical boxes. With NFV and SDN, these functions are virtualized, and further softwarized. By doing so, 5G networks become more flexible and can choose where and when to implement these functions. Virtual machines and containers are the most classical ways to implement virtualization. The former contains its own OS, while the latter packs only an application and necessary files.

2.1. *NFV architecture*

As shown in Figure 1, to deliver an End-to-End service, SFCs direct the traffic to traversal through a set of network functions. With NFV, these functions become VNFs and are connected by virtual links. Containerization, which is more lightweight and flexible, is selected as the virtualization solution in this work. Then, each of these VNFs is in the form of a set of containers. These containers are thus the components of a VNF and are equivalent to microservices. We assume that these containers are instantiated on physical servers. Therefore, to deploy a container, we need to select a physical machine and allocate a certain amount of resources, such as CPUs and memories.

The main benefit of adopting NFV is that it improves the scalability of 5G networks and facilitates the management of SFCs and network

functions by changing the quantities of containers at any time and place according to the service requirements, traffic conditions, or the decision of operators. In this paper, Kubernetes is selected to manage and orchestrate the containerized VNFs.

2.2. *Kubernetes: container deployment and management platform*

Kubernetes is in charge of deploying containers and managing the life cycle of containers, such as load balancing, self-healing, etc. Inside a Kubernetes cluster, there are a set of nodes that correspond to a set of worker machines. Kubernetes will deploy pods (groups of one or more containers) on nodes. We assume in our paper that a pod is equivalent to one container and one microservice application, which is also a component of a VNF. A node is equivalent to a physical machine or a server.

Since we focus on the resilience performance of network service under traffic variation, our main interest in Kubernetes is the autoscaling mechanism. There are many ways for Kubernetes to apply autoscaling. The most commonly used method is the horizontal pod autoscaling, which automatically updates the number of pods to match the traffic demand. To be more concrete, Kubernetes intermittently observes the metrics of a microservice such as CPU utilization and memory utilization and judges if scaling should be applied or not. When the traffic load increases, Kubernetes will try to scale out by deploying more Pods. If the load decreases, Kubernetes will scale in some pods to make sure the resource utilization is at the expected level.

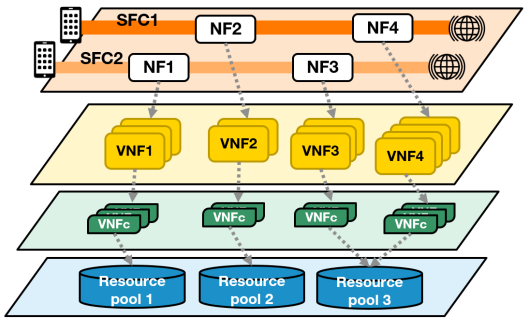


Fig. 1. 5G End-to-End service delivery model with SFC, VNF, VNF component and Resource layers.

We assume that in this paper, Kubernetes only observes the underlying resource, the CPU utilization. For each pod, it allocates several units of CPUs from the node. To process a packet, the pod will use one unit of CPU. We also assume that the autoscaling is applied at the microservice level, i.e., changing the number of pods (replicas). The metric we collect will be the average utilization rate of the pod of the same microservice. A scaling-out action will be carried out only when there are enough physical resources on a physical machine to create a new pod.

2.3. *Performance indicators of End-to-End network services*

This paper mainly focuses on two performance indicators, latency and acceptance rate. These two indicators can be directly applied to estimate the network performance under an undesirable event. We extend the definition of resilience by introducing the "resilience triangle" (Tierney and Bruneau (2007)) and the use service acceptance rate as a system performance indicator to measure network resilience.

Latency is one of the critical indicators for network service. Latency describes the time that takes to transfer a given piece of information from a source to a destination, from the moment it is transmitted by the source to the moment it is successfully received at the destination (3GPP (2021)). In our 5G system model, network service latency is composed of transmission time in RAN, processing time at each VNF(a set of microservices), and the waiting time in the queue of each microservice. Other types of latency, such as time spent on a switch, are not considered. We calculate the average latency of the packets in a time interval of 0.1 seconds. Service latency is very important for vertical industries such as remote health and autonomous driving. The 5G network resilience requires the system to meet low latency requirements despite the presence of risks.

We propose an acceptance rate indicator to estimate the 5G networks' resilience. Based on our assumptions, when a microservice queue is full,

the arriving packets may not join the queue and then be rejected. The packet losses in Transport Network and the radio transmission are not taken into consideration. In some cases, a network service can be very sensitive to packet loss since it impacts the quality of receiving data. The acceptance rate is the number of packets arrived at its SFC destination divided by the total sent packets over a time interval of 0.1 seconds. In a normal operation mode, the packet acceptance rate should be 100%. However, these indicators will not stay at a stable interval under some incidents. For example, in the case of traffic variation, congestion may occur at some microservices. As a result, packets may need to queue up for an available microservice pod and even be rejected if the queue is full. Then the latency will increase, and the acceptance rate may decrease.

In this work, both latency and acceptance rate are used to estimate the network service performance. By further presenting acceptance rate performance as a "resilience triangle", the resilience of the network can be described as the ability to adapt, maintain, and recover.

3. Petri Net-based model for 5G system description

Quite a number of different approaches have been applied to model 5G networks. In this paper, we focus on the dynamic behaviors of a 5G system. We intend to track how packets are processed in the system, creating numerous states for the system. Some approaches, such as Markov Chain and fault tree, are not practical to describe the dynamics or capture the dependencies of such a complex system. Petri Net is a widely used technique for tracking systems' states, dynamics, and constraints. We use Petri Net to model the 5G system. Readers are invited to refer to Li et al. (2022), our previous work, for more details on the Petri Net model.

3.1. *Petri Net for a packet processing*

A request packet is processed in 5G networks by a series of VNF, which can be further extended into a series of microservices. Figure 2 shows a Petri Net of one of these microservices. As explained in Table 1, a packet first arrives at the microservice at place p_1. Then the packet is inserted by t_1 to a waiting list if there is enough capacity in the queue p_2. Otherwise, this packet is rejected to the place p_3. According to the load on the microservices replicas (in the form of pods), the packet will be sent by t_2 to the less used pod p_4. The queue follows the rule of first-come, first-served. Then the packet passes a timed transition t_3 and finally successfully finishes the task in microservice p_5.

Table 1. Place and transition explanations of a microservice process Petri Net.

Element	Explanation
p_1	Packet(s) arriving at microservice
p_2	Packet waiting list for microservice
p_3	Packet rejected due to queue capacity
p_4	Pod replicas of microservice
p_5	Treated Packet(s)
t_1	Packet(s) inserting to waiting list
t_2	Pod selection based on workload
t_3	Packet processing

3.2. *Petri Net for microservice-level autoscaling*

Scalability is one of the most important features of a 5G system. Dynamically changing the system scale according to the load will vastly improve the performance, particularly the resilience performance. In this paper, we focus on microservice-level autoscaling. Kubernetes, as an automatic deployment and management platform for microser-

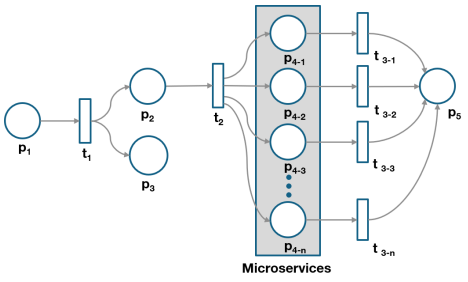


Fig. 2. Petri Net of a microservice process.

vice pods, takes charge of changing the number of pod replicas according to a certain algorithm as presented in Algorithm 1. In this paper, the autoscaling mechanism is to observe the average pod CPU usage metrics intermittently. This time interval is called the sync period. The CPU usage depends on the number of packets that a pod processes. When the resource utilization rate is above the upper threshold, Kubernetes will send a scaling-out decision to increase the number of microservice replicas (i.e., pods) and vice versa.

Algorithm 1 Autoscaling algorithm

Input:
CPU metric values: $I = [I_1, I_2, ..., I_n]$,
desired CPU metric value V,
upper bound:B_U, lower bound:B_L
Output: new replica number: N

desired number of pod replicas: $N \leftarrow n$
sum of indicator values: $s \leftarrow 0$
for $i = 1$ to n **do**
    $s \leftarrow s + I_i$
end for
average of indicator values: $a \leftarrow \frac{s}{n}$
desired replica number: $d \leftarrow$ ceil $(\frac{a}{V})$
if $a > B_U$ or $a < B_L$ **then**
    $N \leftarrow d$ ▷ new replica number
end if
return N

The Petri Net representation is depicted in Figure 3 and explained in Table 2. Kubernetes at place p_1 collects the metrics, which runs intermittently (15 seconds by default). The algorithm will tell Kubernetes to take different decisions depending on the metric value. If the value is higher than the threshold, the transition t_1 will activate and send an increasing pod number order at place p_2. Free resources at place p_4 and the order from p_2 work together to activate t_4 to create new replicas of the microservice at p_5. If the value is lower than the threshold, transition t_2 will activate and send a decreasing replica number order at place p_3. Then the running replicas at p_5 and p_3 activate the transition t_5 to terminate replicas and recycle the resources allocated by them to p_4. If the metric value is inside the threshold, only transition t_2 will activate, and autoscaling will not be triggered.

Table 2. Petri Net of Kubernetes autoscaling.

Element	Explanation
p_1	Kubernetes autoscaling probe
p_2	Increase pod number
p_3	Decrease pod number
p_4	Free resources
p_5	Running Pod replicas
t_1	Scaling-out decision
t_2	Scaling-in decision
t_3	No scaling decision
t_4	Create new replica(s)
t_5	Terminate replica(s)

4. Simulation and results

In order to test the performance and resilience of 5G networks, we modeled the system in a Python program based on the Petri Net representation. Then we run discrete event simulation using the SimPy framework based on Python. The microservice processes in the Petri Net model are coded as resource allocation events in the program. We apply the program to two use cases to evaluate network resilience under different situations. In the first use case, we consider a 5G system with one type of user equipment. We inject a traffic variation by increasing the number of packets sent by the user equipment to test the network resilience. In the second use case, the 5G system consists of four local RAN and one centralized CN. We inject the same traffic variation only to the

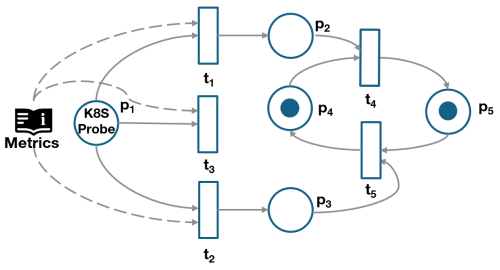


Fig. 3. Petri Net of autoscaling process.

user equipment in one zone and test the network resilience performance in the whole system.

For the first use case, the End-to-End service follows an SFC composed of 3 VNF, as depicted in Figure 4 (in the user plane, they could be DU(Distributed Unit), CU(Centralized Unit)) in RAN, and UPF in CN). The transmission time and packet processing time follow an exponential distribution. However, the packet arrival rate follows a Poisson distribution with a variant parameter. Other parameters are given in Table 3.

Table 3. 5G network parameters.

Number of VNF MSs (microservices)	DU:1 MS, CU:2 MS, UPF:1 MS
MS processing time	8 ms for DU and CU MSs, 10 ms for UPF MS
MS resource (in CPU units) allocation	6 for DU and CU MSs, 12 for UPF MS
Packet processing resource	1 CPU unit for each MS
Initial container/pod replicas	3 pods for each MS
Node capacity (in CPU units)	18 in RAN and 36 in CN
Number of nodes	4 in RAN and 8 in CN
Desired CPU utilization rate	50%
Autoscaling threshold	$\pm$ 30%
Simulation run	1000 iterations

The network has been initially well scaled to meet the traffic of 1000 requests per second. We inject a traffic variation into the system. The request arrival rate linearly increases which lasts 25 seconds from 1200 to 4200 requests per second, beginning at 10s and ending at 35 seconds. Then the traffic goes back to its normal state.

In the normal state, the packet average delay is around 0.035 s, including 34 ms processing delay, 1 ms transmission delay, and negligible waiting delay. However, the load on pods grows with the traffic, and they are soon congested. When a packet demands a microservice, there are no more available pods to serve it. The waiting delay increases, and when the waiting list is complete, the coming packets will be rejected. We compare the packet latency and acceptance rate results under different autoscaling strategies.

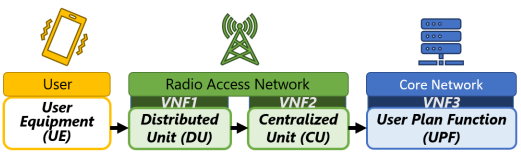


Fig. 4. Service function chain of use case 1.

The service delay result is given Figure 5. If there is no autoscaling, the waiting delay increases up to 80 ms, and the overall delay will not decrease unless the traffic comes back to normal. When we adopt a 15 seconds autoscaling sync period, we find that few pods are scaled out at time 15 s, and more pods are scaled out at 30 s. These two scaling operations are not enough to immediately handle the congestion. In the 10 seconds sync period situation, the scaling-out decisions are taken at 20 and 30 s. The network service delay is shorter than the 15 seconds sync period case after 30 s. Finally, in the 5 seconds sync period autoscaling case, scaling decisions are taken more frequently, and the congestion time and service delay are significantly reduced.

The service acceptance rate result is given in Figure 6. Without autoscaling, the acceptance rate may reduce up to almost 50%. With 5-second autoscaling, both duration and packet rejection is largely reduced. The resilience is improved by shortening the time to adapt to the reverse event and better maintaining the performance. While for 10-second or 15-second autoscaling, the disturbance interval is not significantly reduced, the maximum packet acceptance degradation is about 40%. The acceptance rate is improved only after 30 s. However, the system is not fully recovered. It keeps suffering from the disturbance since the autoscaling at 30 s is insufficient to cope with the continuously growing traffic.

By comparing the acceptance rate performance, the 5-second performs the best in terms of service latency, system suffering time, performance degradation, and restoration time. However, frequently adjusting the scale of 5G network may not be a wise choice. When doing scaling-in, it

takes some time to terminate pods gracefully. The pod resources will not be released immediately. During this time, some of the resources become unavailable, and the system may not be able to scale out when the traffic immediately increases due to a lack of resources. Therefore, some more complicated algorithms can be further applied to set up scaling rules to better adjust the system to the traffic load.

In the second example, we move closer to reality. The DU and CU are located in the local RAN, and UPF is located in a centralized CN as depicted in Figure 7. We consider a 5G system composed of 4 local RAN, which treats the local end users' packets, and a centralized UPF, which treat all end users' packets. Unlike the previous example where we only consider one VNF instance, in this use case, different DU and CU instances are assigned to the user equipment in different zone according to geographical locations. Therefore, these user equipment's packets are isolated in DU and CU but not for UPF. A traffic variation in one zone will firstly congest DU and CU and probably UPF, which is initially set up with more redundancy than local VNFs. Then the packets from other zones will also be delayed due to the congestion happening in shared UPF. In this example, the autoscaling sync period is set to 10 seconds. The abnormal traffic in zone 1 is the same pattern as in the first use case. The latency and acceptance rate for packets starting from different zones are presented in Figure 8 and Figure 9. The parameters are similar to use case 1.

The traffic change from zone 1 congests not only the local VNFs DU and CU but also UPF. Since the UPF is initially scaled for four radio network zones, the traffic congestion on UPF is

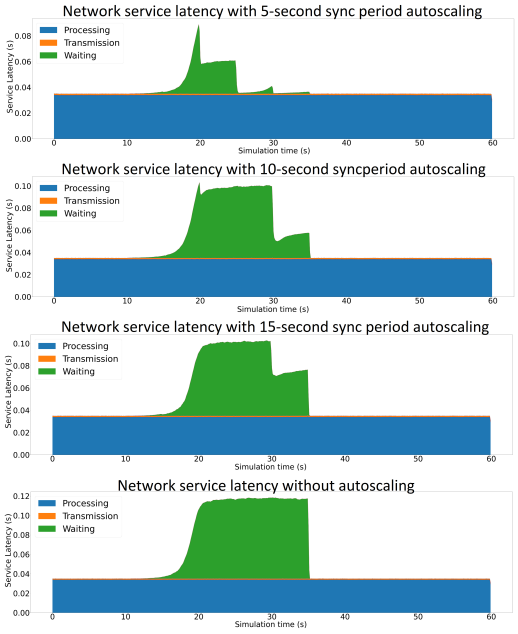


Fig. 5. Network service latency with and without autoscaling. Blue for processing delay, yellow for transmission delay and green for waiting delay.

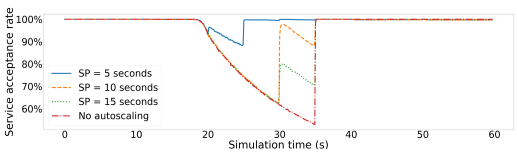


Fig. 6. Network service acceptance rate with 15, 10, 5 seconds sync period autoscaling and no autoscaling.

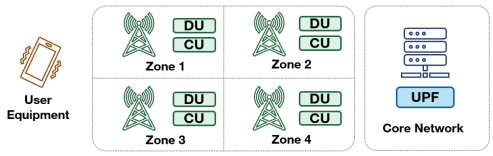


Fig. 7. Network installation of use case 2.

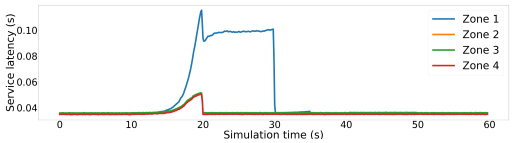


Fig. 8. Network service latency in different zones.

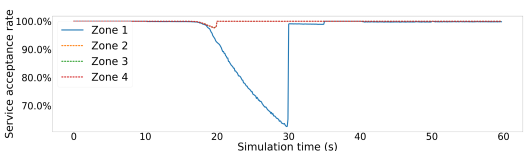


Fig. 9. Network service acceptance rate in different zones.

less severe than in use case 1. The packet waiting delay in zone 1 increases to 70 ms. The packet waiting delay in other zones is about 15 ms, only caused by UPF congestion. At 20 s, a scaling-out decision is taken by Kubernetes. It creates the maximum number of pod replicas that our scaling algorithm allows. For the UPF microservice, scaling out once is enough, the congestion is released, and the packets in zone 2-4 are no longer queuing for UPF. However, in zone 1, local DU and CU microservices are still congested. The result shows that a traffic change can cause congestion on VNF and the congestion can propagate from RAN to CN. By adopting a local RAN isolation, only zone 1 is largely impacted by the traffic change. The network services of other zones are less impacted, with only less than 3% packet loss proving that the effeteness of network isolation on improving network service resilience. Although in this use case, we are limited in physical or geographical network isolation, this result can still be meaningful since it may be extended to a virtual isolation case for 5G QoS or network slicing.

5. Conclusion

In this paper, we proposed a model based on Petri Net for 5G system, considering the dynamics of the virtualized telecommunication network. This model is applied to simulate the performance of 5G network services for two use cases. The results from two use cases show that microservice-level autoscaling can increase the resilience of 5G network in case of traffic variation and how congestion caused by traffic change can propagate from the local RAN to CN. These results are useful for network operators to implement orchestration and management systems and design network isolation according to service requirements.

This work gives an approach to estimating network performance and resilience. Although many parameters, such as processing time and pod capacity, are based on assumptions, the qualitative results on the efficiency of autoscaling and propagation of congestion are still valid. For better simulation results, they can be further refined.

The continuation of this work will focus on a 5G network comprising different network services and analyze if 5G networks are able to satisfy the resilience requirements of different verticals.

References

3GPP (2020, July). TR 38.913 V16.0.0 Study on Scenarios and Requirements for Next Generation Access Technologies.

3GPP (2021, December). TS 22.261 V16.16.0 Service requirements for the 5G system.

Alawe, I., A. Ksentini, Y. Hadjadj-Aoul, and P. Bertin (2018). Improving traffic forecasting for 5g core network scalability:a machine learning approach. *IEEE Network 32*(6), 42–49.

Li, R., B. Decocq, A. Barros, Y. Fang, and Z. Zeng (2022). Petri net-based model for 5g and beyond networks resilience evaluation. In *2022 25th Conference on Innovation in Clouds, Internet and Networks (ICIN)*, pp. 131–135.

Nencioni, G., R. G. Garroppo, A. J. Gonzalez, B. E. Helvik, and G. Procissi (2018, August). Orchestration and control in software-defined 5g networks: Research challenges. *Wireless communications and mobile computing 2018*.

Rahman, S., T. Ahmed, M. Huynh, M. Tornatore, and B. Mukherjee (2018). Auto-scaling vnfs using machine learning to improve qos and reduce cost. In *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6.

Rotter, C. and T. Van Do (2021). A queueing model for threshold-based scaling of upf instances in 5g core. *IEEE Access 9*, 81443–81453.

Sterbenz, J. P., D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith (2010). Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Computer Networks 54*(8), 1245–1265. Resilient and Survivable networks.

Subramanya, T. and R. Riggio (2021). Centralized and federated learning for predictive vnf autoscaling in multi-domain 5g networks and beyond. *IEEE Transactions on Network and Service Management 18*(1), 63–78.

Tierney, K. and M. Bruneau (2007). Conceptualizing and measuring resilience: A key to disaster loss reduction. *TR news* (250), 14–17.

Estimating 5G network service resilience against short timescale traffic variation

Rui Li, Bertrand Decocq, Anne Barros, Yi-Ping Fang, *Member, IEEE*, Zhiguo Zeng

***Abstract*—5G networks are designed to create a new ecosystem for vertical industries such as health care, energy, and public transport. These novel applications, on the other hand, bring new challenges to network resilience. Among them, traffic variation is one of the most vital threats to the 5G network. With tens of thousands of devices connected to the network, network service resilience is threatened by the heavy traffic change induced by the end users or malicious attacks. While long timescale traffic variation can be easily predicted based on historical data, short timescale abnormal traffic is hard to forecast yet can significantly violate the service requirements. The impact of short timescale traffic variation can be mitigated by 5G management and control systems. However, the complexity and dynamics of the virtualized 5G system make it hard to estimate its resilience. This paper provides a 5G network model that captures the data traffic changes and network dynamic management mechanism. The model is able to evaluate the performance of different network services with different requirements under traffic variation events. We analyze the effectiveness of auto-scaling and compare different isolation strategies for traffic congestion. The simulation results on service resilience estimation can become strong supporting information for 5G network deployment and configuration.**

***Index Terms*—5G, network resilience, auto-scaling, virtual networks, traffic variation, communication networks, Kubernetes, network service, Petri Net, discrete event simulation.**

I. Introduction

ONE of the most ambitious goals of 5G is to empower vertical markets and to realize a sustainable ecosystem. The Next Generation Mobile Networks (NGMN) Alliance [1] has identified many vertical industries that can benefit from 5G, such as transport, smart grid, health, and wellness. Each covers many different use cases. The smart grid applications, for example, may contain use cases of equipment monitoring, fault localization, network isolation, etc. 5G visions to support a large variety of these vertical applications with varying characteristics and requirements. Depending on different scenarios, the requirements on peak data rate, bandwidth, latency, and reliability can be completely different [2].

Building such a vertical ecosystem requires a more flexible network. In order to deliver services more dynamically, 5G networks take benefit from a set of technologies, such as Network Function Virtualization (NFV) and Software Defined Networking (SDN) [3]. The principle idea is to construct a virtualized network and deploy it flexibly according to specific requirements. NFV proposes to extract network functions from dedicated equipment and makes them work in a virtualized environment. It introduces a virtualization architecture based on the physical infrastructure on which several virtual machines or containers run. At the same time, SDN separates the control plane and the data plane by centralizing the intelligence of the hardware infrastructure at the level of a controller to support the NFV infrastructure and architecture configuration. Based on NFV and SDN, network slicing proposes a customized network for 5G verticals to support diverse requirements.

The above-mentioned technologies create a virtualized network to support the 5G ecosystem. However, a key issue before putting such a network into service is to verify if the diverse requirements can be satisfied, including its resilience in the presence of adverse events. With thousands of user devices and services connected, testing on a real network is not practical. We thus propose to simulate the network performance based on a 5G network model. In this paper, we mainly focus on vertical service's latency and acceptance rate requirements, and consider resilience to adverse events. This work chooses incidents caused by traffic variation as the adverse event for the analysis since traffic change happens more often, especially with the expansion of new connected objects, becoming a challenging issue to ensure service performance.

The traffic variation, one of the main threats to 5G network, brings many uncertainties to the configuration and makes it hard to prepare the system with an appropriate scale. 5G network is initially well configured for a desired functioning state of the services. 5G system can be dynamically configured using 5G NFV Management and Orchestration (NFV-MANO) when the environment changes. It tries to re-scale itself to save energy when there are few service requests. When the service requests grow, it increases its capacity. A long-time mobile traffic forecast can precisely anticipate the traffic change during a week or a day, as found in [4], [5]. However, in a short period, adverse event as DDoS attacks, flash mobs, and some impromptu events could induce abnormal traffic that is hard to predict. A real example of network behavior during a football match is reviewed in [6]. During an adverse event, a 5-minute disruption would be tolerable for a smartphone user. Yet it could be catastrophic for a reliable-sensitive use case and leads to severe consequences. For example, real-time applications, such as remote surgery, factory automation and intelligent transportation, require reliable and precise information and feedback [7]. When the connection is disrupted, some pieces of important information may not be completely delivered. Then this service loses its reliability and becomes unavailable and eventually causes serious railroad accidents. Although short-term performance loss becomes critical in network resilience, few works have focused on a short timescale traffic variation. On the one hand, traffic changes rapidly in fine timescales of seconds, which is hard to predict. On the other hand, the resilience performance may depend largely on the traffic pattern a 5G network encounters and the management methods it applies. In this article, we simulate the Kubernetes platform-Based NFV-MANO (as it provides operators with a lighter, more portable container 5G network) and its built-in control algorithm, propose different traffic change scenarios, and estimate the short-term resilience loss under traffic variation.

The main contributions of this work are the following:

- The 5G telecommunication network is modeled by a hierarchical Petri Net for short timescale resilience analysis.

- The model takes into consideration the dynamic behaviors of both packet processing and micro-service management.
- The resilience loss of different network services under traffic variation is estimated with a proposed service reliability-based resilience metric.
- The effectiveness of service isolation strategies during an adverse event is examined.

The paper has been organized in the following way. Related works are discussed in Section II. We present the virtualized 5G network in Section III. In Section IV, the Petri Net-based 5G network model is explained. Service performance and resilience metrics are introduces in Section V. The model is applied to two case studies in Section VI. Finally, Section VII concludes the paper and outlines future work directions.

II. Related works

For a communication network, resilience often refers to the ability to provide and maintain an acceptable level of service during failures and incidents, as pointed out in [8]–[10]. Focusing on 5G resilience, Esposito et al. [11] introduced the threats in Information and Communications Technology (ICT), such as extreme weather, power outage, software failure, and attacks that lead to the escalation of disasters in 5G networks. They highlight the importance of ensuring adequate levels of resiliency for future network paradigms. Dutta and Hammad [12] classify 5G threats based on different consequences, such as loss of availability and confidentiality. They also focus on identifying associated system vulnerabilities and corresponding mitigation techniques. Hutchison and Sterbenz [8] depict how a resilient network can be constructed by considering components that interact with each other. To build a resilient telecommunication network, operators need to evaluate the network resilience performance in case of various unfavorable events. Mauthe et al. [13] make an explicit mention of cost effectiveness in the resilience definition and highlight the need for resilience to be quantifiable. They also point out the importance of analyzing the risks associated with challenges in a given context. In [10], resilience-related metrics are classified into topological and functional metrics. Topological metrics, such as centrality, and connectivity, are the metrics directly related to the network topology and independent of how data is transmitted, as the works in [14], [15], whereas others focus on the functional metrics, such as latency, are metrics that are closely related to data flows and can evaluate the impact of an incident on applications and users, and they are strongly related to QoS metrics.

Some works estimate 5G resilience by looking at how an incident may impact resilience metrics dynamically. Awad et al. [16] build a framework to improve software-defined radio access networks' resilience to sudden changes in network parameters where the system functional metrics, including network latency, are evaluated during the incident. Liu et al. [17] estimate an mMTC network service's performance response function evolution during a typhoon disaster using an assessment framework consisting of five mathematical models. Nakayama et al. [18] estimate the service performance of data transmission during the communication failure scenarios to test a resilience management architecture for communication on portable assisted living applications. [19] proposes a resilient VNF allocation model for increasing the number of accepted requests in a dynamic request scenario and develops a reinforcement learning-based approach. Although dynamic request situation is considered, there is no temporal resilience analysis. [20] formulates a resilient VNF placement model that minimizes the computation resource cost and guarantees recovery against single node failure within the recovery time objective defined for each service.

Indeed, only limited works have drawn attention to the evaluation of network service resilience from the perspective of how network service suffers and adapts to the incident. They neglect the network components and relations between them, which could be necessary for system resilience analysis. Instead of estimating the performance evolution during adverse events, most works assume there is a more "static" or "average" service performance loss in case of incidents or failures, and it can be helpful for system conception and design from a preventive perspective.

5G network performance assessments have been carried out by various studies. The considered performance indicators may include the quality of service, network availability, installation, and operational cost. Depending on the goal and the context, the applied approaches can differ from one to another.

Di Mauro et al. [21] model the probabilistic behavior of a containerized IP Multimedia Subsystem using Stochastic Reward Networks and Reliability Block Diagram. This model gives a joint analysis of availability and performance by considering both failure and repair events.

With a focus on the base station, Farooq et al. [22] use the Continuous Time Markov Chain to analyze the reliability behavior of a base station for the future by taking into account the arrival of faults and recovery effects. In [23], the authors develop a semi-Markov model to quantitatively estimate both transient and steady-state availability of a Multi-access Edge Computing service function chain. Although dynamic behaviors can be investigated using this model, service requirements such as latency and packet loss are not considered.

In [24], a queuing-based model is introduced to the network orchestrator to optimize the system resource allocation regarding the vertical's requirements. In this work, service delay is chosen as the main performance indicator. In [25], an analytical queueing model is also established to accurately evaluate the E2E packet delay for multiple traffic.

Li et al. [26] propose a game-theoretical approach to solve an SFC embedding problem. In this approach, SFC is seen as a player and minimizes the overall latency subject to capacity constraints. Singh et al. [27] give a more general insight by surveying the game theory applied to analyzing and modeling the 5G system. They give special attention to the coalition games applications on resource management, interference management, and miscellaneous.

Linear programming (LP) has been widely used to formalize a telecommunication network problem. Instead of estimating a transient service performance, this approach seeks an optimized solution subjected to certain constraints. Objective functions formulate the aim of optimization, such as minimizing cost, minimizing resource allocation, or maximizing performance. Decision variables are the configurable parameters in the 5G network system to be estimated to obtain the optimal solution. The other 5G system structure or limitations and the service requirements are presented as constraints. In [28], a cost minimization problem is proposed using integer linear programming to obtain a cost-efficient solution to VNF redundancy allocation. In [29], to efficiently find the minimum end-to-end service latency, Dong et al. [30] minimize the total cost of service function chain deployment while ensuring that the Quality of Service (QoS) requirements are

satisfied. Wu et al. [31] formulate an integer linear programming problem to decide where to place virtual network functions (VNFs) while guaranteeing service reliability. In [32], two integer linear programming problems are formulated to minimize the network service deployment cost while meeting latency requirements and identify the optimal locations concerning reliability.

In the above work, the network performance, either latency or reliability, is generally treated in a static or stationary way. The latency is normally calculated without considering congestion. The reliability is seen from the system level (hardware and software reliability) without considering how many service requests can be successfully delivered during a short period in adverse conditions. Indeed, various network metrics are dynamic, and the scale and parameters of the 5G network change according to the environment. The aspect of the dynamic transient behavior of 5G networks is missing in these approaches.

In order to take into account dynamic behaviors, Petri Net-based model has recently been introduced to network service performance evaluation. Schneider et al. [33] use Queuing Petri Nets to formally and unambiguously specify the behaviors of network functions. They succeed in expressing queuing, synchronization, processing delays, and changing traffic volume and characteristics at each VNF. This approach allows to estimate and compare the QoS of different configurations. Rui et al. [34] proposed a Petri Net-based algorithm that can choose the service chain based on service reliability in a service pool. Petri Network is used to describe the failure and propose the migration strategy. This work analyzes reliability from both transient and steady state perspectives. However, the service performance aspect, such as service latency and packet loss, is missing. The traffic flow is also not modeled. In [35], a hierarchical colored generalized stochastic Petri Net-based framework is proposed to evaluate a cloud data center service reliability. The dynamics of service delivery are taken into consideration. This study focuses on the reliability of the system.

Despite the efforts made in these frameworks, not all dynamic behaviors that affect the performance of short-time labeling services are well captured. In particular, the dynamic management and configuration of the network, to which the service performance and resilience are sensitive, are not addressed. In this paper, we intend to build a Petri Net-based model that describes the dynamic behavior of the network, namely, the auto-scaling mechanism, and captures the packet-level network performance to help produce a short-term resilience evaluation during an adverse event.

In our previous work [36], we introduced a Petri Net-based model for network availability estimation. This model captures single failures and common cause failures, and describes how self-healing takes action in a failure event but we does not consider the traffic and any service using the network. In [37], we have refined the model to calculate service data packet latency and rejection rates. In this paper, we present the model comprehensively, adding the Protocol Data Unit (PDU) session connectivity and provide resilience analysis from network service perspective.

III. Virtualized 5G system

In this section, we introduce the scope of the proposed model: NFV, PDU sessions, and network slicing. Then in the second part, we present the importance of capturing network dynamics for resilience analysis during adverse events.

A. Functional description of virtualized network

To provide innovative, customized vertical services on demand and guarantee service performance and resilience of a 5G system, network slicing based on SDN, NFV, and a cloud-native 5G core is a promising solution [38], [39]. With network slicing instances [40], the 5G physical network is sliced into multiple isolated logical networks of varying sizes and structures dedicated to different services that provide the necessary flexibility and scalability to vertical networks [41]. Protocol Data Unit (PDU) builds connectivity for end-to-end services. This connectivity enables the data packet exchange between a single end user and the internet. Thus, as pointed out by Ferrús [42], the realization of network slicing relies on the principle that each PDU session is associated with a particular network slice. End users for different network services will use different network slices and establish different PDU sessions. Once the session is established, the end user can start exchanging packets with the network by steering between a set of network functions belonging to its slice. Then above the physical infrastructure, we create several virtual networks. The whole network resources are therefore allocated to different slices according to the service requirements.

During an anomaly, network slicing isolates the service from outside adverse events. However, network slicing requires more resource allocation than a shared network to maintain network service performance during an incident. When an incident occurs in a shared network, by applying a priority mechanism, priority is given to guaranteeing critical services while sacrificing some less critical services to avoid violating service level agreements.

To provide efficient control for such a complex system facing various adverse events, NFV Management and Orchestration (NFV-MANO) [43] is used to anticipate the incident or adjust network rapidly to avoid requirement violation and, eventually, economic loss. NFV-MANO manages and orchestrates VNFs and other software components and ensures the correct operation of the NFV infrastructure and VNFs [44]. The exact mechanism to implement the NFV-MANO could depend on the service requirement, or the choice of operator, but at the moment, it is hard to have a mechanism that can economically avoid the degradation of service performance under all scenarios.

B. Challenges in system resilience

In order to perform a resilience assessment, we need to understand how the complex virtualized network is composed and look at the specific scenario in which it is applied.

Though at the conception phase, the networks are designed with a certain degree of redundancy margin and some NFV-MANO mechanisms. If the initial margin is not enough, the VNF-MANO takes over and changes the configuration to avoid overload. Therefore, we are faced with a dynamical system where the traffic can be dependent on time, and the network configuration may also change with traffic demand and service of quality demand. Without capturing the dynamics of the system, a short-term degradation of service quality caused by adverse events will be neglected, making it difficult to analyze service resiliency and to configure the network.

IV. A Petri Net-based model for dynamical 5G network

To better model the constraints and dynamics of 5G, we propose a hierarchical Petri Net model to represent the 5G

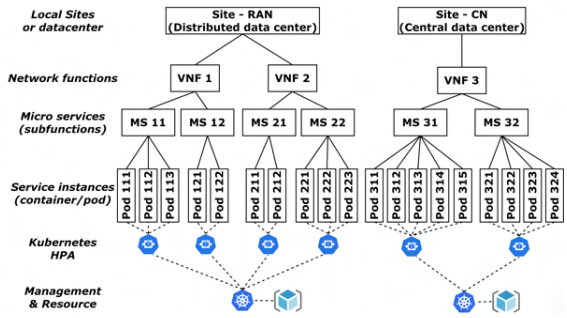


Fig. 1. 5G container-based NFV hierarchy topology example with one local site and one centralized data center site.

network. We focus on how a network service traverses the 5G network, and how the 5G system dynamically reacts to meet the service requirement.

In this work, the proposed generic approach could be applied to different network designs. Even though two cases are proposed in Section VI, this approach is not limited to the parameter settings in the cases. We can easily vary the design parameters, such as network locations, the number of containerized micro-services. However, the choice of multiple locations and the number of considered micro-services will increase the complexity of the model. When changing the network design and parameters, the relationship between these sub-Petri Nets should be carefully and explicitly expressed when the network setup changes. Otherwise, the model could fail to capture precisely how the network service process works.

A 5G network system topology is considered hierarchical, as presented in Fig. 1. It comprises of several physical sites, including locally distributed sites and central data centers. In each site, network functions are virtually implemented. We assume that VNFs are containerized. Each VNF consists of container-based micro-services (equivalent to sub-functions). These micro-services have multiple replicas in parallel to share the load. These basic units are managed by a micro-service level controller, which is connected to Kubernetes, taking charge of the utilization of the resource pool of the site. By using hierarchical Petri Net, the 5G system is decomposed into sub-Petri Nets, which are given in the following sections. Since the exact 5G system structure may vary from operators and service providers, we briefly introduce a generic system model based on our assumptions.

Based on the preceding works [36], [37], we build a Hierarchical Timed Stochastic Colored Petri Net. The highest level is the network functions Petri Net, which represents packet generation, processing, and transmission in the 5G network. The sub-networks are used to represent how the packet is generated, processed and transmitted. From the management aspect, a sub-network on micro-service management shows how the network dynamics react to the environment.

The net model uses places and transitions to represent how the network system and service dynamically change with time. Message packets and telecommunication network components are represented in tokens that can change the states. Places P represent the state of the process of packets, such as transmission and processing, or the state of the network components, such as working mode and failure mode. Transitions T enable these packet and component tokens to change their states.

A. Service delivery

5G network is composed of Radio Access Network (RAN), Transport Network (TN), and Core Network (CN). In this study, a virtualized RAN (vRAN) is directly located in the local cell. The functions in RAN are all virtualized using the physical resources in the distributed local site, just as Site - RAN in Fig. 1. TN is assumed to be 100% reliable and with enough capacity to transfer all packets. The CN is installed in the operator's data center, just as Site - CN in Fig. 1. We consider a vertical industry network service in which only the up-link data is transferred and it happens only in the User Plane (UP). The request packets start from end users. End users randomly appear in cells. Each end user will use either vertical service 1 or vertical service 2. Before sending packets to the internet, we assume that the end user has already established a PDU session, which builds connectivity between the end user and the network. Once the PDU session is launched, the end user starts sending packets to the network until the session terminates. These packets follow a service function chain containing three VNFs by assumption, Distributed Unit (DU, providing support for the lower layers of the protocol stack), Centralized Unit (CU, providing support for the higher layers of the protocol stack) in vRAN, and User Plane Function (UPF, connecting the data from the RAN to the Data Network) in CN. The packets are locally processed at the distributed RAN sites for DU and CU, and then at Core Network for UPF.

Fig. 2 shows an exemplified service delivery level Petri Net, including local site layer, network function layer. Local RAN sites 1-4 and Core Network correspond respectively to Site - RAN and Site - CN in Fig. 1. The VNF processes in Fig. 2 correspond to the Network functions layer in Fig. 1. As explained in Table I, p_1 is the starting place, representing the end users from the cells. Then they start PDU sessions by a sub-Petri Net represented in transition t_1. The established PDU sessions in place p_2 keep generating packets with t_2 during the lifetime of the session. These packets in p_3 will then start the vRAN process in the local site where it starts. In a Local RAN (site 1, for example), the packet becomes input in place p_{41}, the ingress gateway, and processed in the VNF process sub-Petri Net t_{41}. After being processed by the VNF, it arrives as p_{51}. As VNFs are processed in order, transition t_{51} sends the packet back to p_{41} to pursue the next VNF, CU, if the packet finishes all processes in DU. If a packet is processed in both DU and CU, it will be transmitted to Core Network p_{40}, where it will pursue processes with UPF. Finally, after being processed in t_{40}, the packet arrives at p_{50} and then transition t_6 transmits the packet to Data Network p_6.

TABLE I
DESCRIPTIONS OF TRANSITIONS IN SERVICE DELIVERY

Transition	Type	Input token	Output token
t_1: PDU generation	Sub-Petri Net	User	PDU session
t_2: Packet generation	Sub-Petri Net	PDU session	New packet
t_{3x}: Radio transmission	Immediate	New packet	Packet
t_{4x}: VNF process	Sub-Petri Net	Packet	Packet
t_{5x}: VNF Route	Immediate Timed(to CN)	Packet	Packet
t_6: Packet reception	Immediate	Packet	Packet

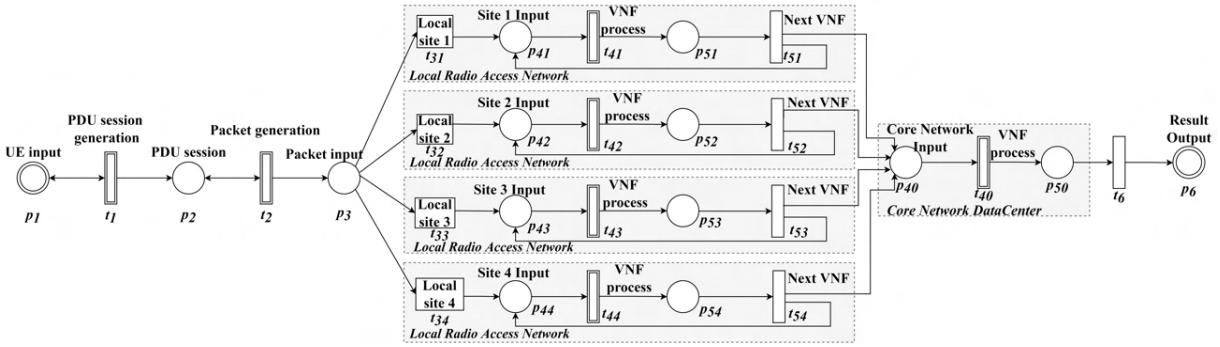


Fig. 2. Service delivery level Petri Net. Example with four radio cells and one core network data center.

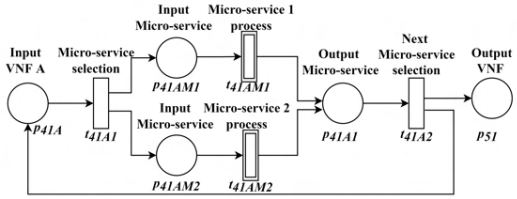


Fig. 3. VNF processing level Petri Net. Example of VNF A.

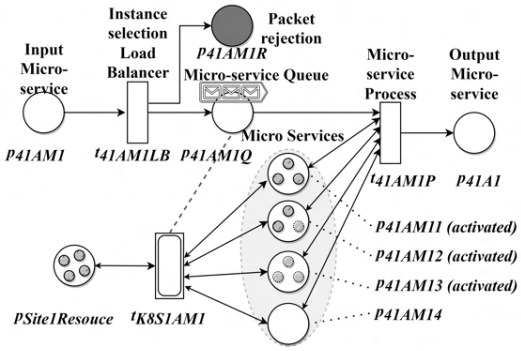


Fig. 4. Packet processing level Petri Net. Example of micro-service of the first VNF in t_{41}, VNF A.

B. VNF and Micro-services

As a site has a set of VNFs, a VNF is composed of a set of sub-functions known as micro-services. The sub-Petri Net transitions t_{4x} (for example, t_{40}, t_{41}, t_{42}, t_{43} and t_{44}) in Fig. 2 lead the service packet to the corresponding VNF needed according to its service function chain and its PDU session. One of the VNF process, VNF A process is shown in Fig. 3. In this level, after one micro-service is processed, the packet will pursue the next one in the same VNF or another VNF, according on the processing sequence.

C. Micro-service/container processing

We model the micro-service process by a queueing model. A detailed example of the micro-service in VNF A of site 1 is given in Fig. 4. When a packet arrives at the micro-service p_{41AM1}, it will pass through a resource-based load balancer t_{41AM1Q} to different micro-service instances. By adopting NFV in 5G, these instances are either VM-based or container-based. In this 5G model, we assume that all network functions are container-based and are managed by the Kubernetes platform. The minimum manageable unit in Kubernetes is a pod, which is one or a set of relevant containers. We assume that in this model, each pod is exactly one container. Based on the resource limit of the site, we also assume a maximum of n (4, for example) pods that can be instantiated to share the traffic load. A pod is equivalent to a container, requiring specific resources (CPU in our case) to instantiate. The place $P_{Site1Resource}$ provides a shared resources pool to all micro-services on the site. When instantiating a pod instance, CPU resource tokens will move to the corresponding pod place. When deleting a pod instance, its resource tokens will move back to the site resource pool. To process a packet that arrives at the load balancer, t_{41AM1P} takes one resource from the pod with the most CPU resources. This timed transition will bring the packet to p_{41A1} and return the resource after a processing time. When there are no available resources in any of these pods, this packet will have temporally waited until there is a new resource. If the queue is full of packets, the system may reject a newly arrived packet. A detailed explanation of transitions and places is listed in Table II and III.

TABLE II
EXPLANATION OF TRANSITIONS IN PACKET PROCESSING

Transition	Type	Conditions
t_{41AM1Q} Join the queue	Immediate	Packet joins p_{41M1Q} if not congested Packet is rejected if p_{41M1Q} is full
t_{41AM1P} MS process	Timed	Process packet if resource is available Packet waits if no available resource
$t_{K8S1AM1}$ MS controller	Periodic Immediate	Intermittent activation Subject to MS resource utilization

TABLE III
DESCRIPTIONS OF PLACES IN PACKET PROCESSING

Place	Token color	Explanation
p_{41M1}	Packet	Packet to be processed in MS
p_{41M1R}	Packet	Packet rejected due to capacity limit
p_{41M1Q}	Packet list	MS packet waiting list
p_{41A1}	Packet	Packet processed by MS
$p_{\text{Site1Resource}}$	Resource unit	Resource pool of the site
p_{41AM1x}	Resource unit	MS pod with a certain capacity

D. Micro-service management

We demonstrate micro-service management using a site containing four micro-services as shown in Fig. 5. This Petri Net is divided into several subparts, four in the case of Fig. 5 and one shared resources place in the center. Each subpart

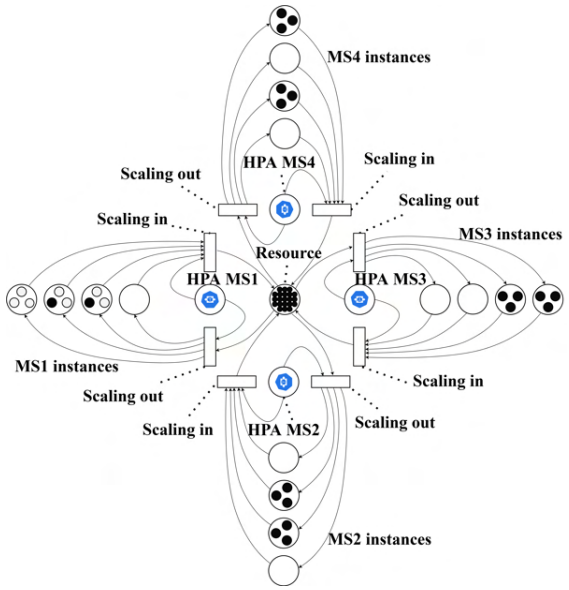


Fig. 5. Micro-service management level Petri Net. Example of a site with four micro-services.

can perform scaling out and scaling in functions proposed by Kubernetes Horizontal Pod Autoscaler (HPA). Kubernetes is assumed to be a fully reliable platform. While Kubernetes takes charge of service orchestration and management, our model only incorporates the function of HPA as a control algorithm for managing the number of micro-service pod instances. The built-in algorithm of the HPA controller runs auto-scaling intermittently (the default interval is 15 seconds). By applying auto-scaling, Kubernetes updates resource allocation, with the aim of automatically scaling the workload to match demand. The controllable objects of the HPA controller are the pod instances of the micro-service in a VNF. A target resource utilization rate is defined for each micro-service, then the controller fetches the CPU utilization metrics and takes the mean utilization value. If this value is outside a specified range, the HPA controller calculates the desired pod replica number needed to obtain the target utilization rate. If the desired number exceeds the current one, it launches a scaling-out action to create supplementary replicas. On the contrary, if the desired number is smaller than the current one, it removes the unnecessary pods. In general, the goal is to dynamically change and adapt the scale of the network so that in a light traffic period, the system uses fewer pods to save energy and resource allocation, and in a heavy traffic period or during an incident, the system creates more pods to avoid being overloaded and guarantee the network service resilience.

V. Performance and Resilience Metrics

In this study, we focus on estimating the resilience of the services that the network operator can offer to vertical industries. In order to address the resilience under traffic change, we propose several resilience-related metrics for evaluation. End-to-end delay and packet loss, two objective functional metrics, are first discussed. They are often used to determine terms of service level agreements and could be very sensitive to congestion caused by traffic variation. In order to analyze and compare the resilience under different traffic variation scenarios, a service reliability-based resilience triangle is introduced. This proposed resilience metric is different from other state-of-the-art metrics as it considers both of the two aforementioned objective functional metrics. Finally, resource allocation cost is considered an additional performance metric from the economic aspect.

A. End-to-end latency

End-to-end latency or end-to-end delay is the time it takes to transfer a given piece of information from a source to a destination [45]. This latency refers to the time to transfer a packet from the end user to Data Network for uplink. For the downlink, it is the opposite direction.

Most vertical services have strict requirements for end-to-end service. From a 3GPP Technical Specifications, in the auto function, for the service of cooperative collision avoidance between users, the maximum end-to-end latency is 10 ms [46]. For urban area railway Very Critical Data Communication, end-to-end latency requirement is also 10 ms for reasons of train safety [47].

When we investigate the latency evolution for a couple of seconds, it seems impractical to examine the end-to-end latency, packet by packet. During congestion, the difference in delay between two consecutive packets can be significant because the waiting time for each packet is random due to the stochastic packet arrival rate. Instead, we prefer to look at the average delay during a short time slot. Equation (1) illustrates a way to calculate the delay of one time slot $]t,t+\Delta T]$ where it uses the average latency of all N delivered packets out of M transmitted packets during this time interval. d_i is the end-to-end delay of the i-th packet. x_i is a binary variable, and it takes value 1 when the i-th packet has arrived at its destination and takes value 0 when the target does not receive it.

$$\text{Delay}(t) = \frac{\sum_{i=1}^{M} d_i \cdot x_i}{N}, \text{ where } N = \sum_{i=1}^{M} x_i \tag{1}$$

B. Packet Loss Rate

Packet Loss Rate is the share of packets the target could not receive, including packets dropped, packets lost in transmission, and packets received in wrong formats [48]. Under the scope of this work, we only consider the packet drop due to the heavy traffic load in the VNF process. More concretely, we consider that for each VNF or each of its components, there is a waiting queue with a limited capacity. When the traffic increases and exceeds the capacity, the packets that cannot join the queue will be dropped. Those lost packets can be fatal for vertical usages, such as the automatic control system, where continuous signals are indispensable. Equation (2) shows how packet loss in the time slot $]t,t+\Delta T]$ is calculated.

$$\text{PL}(t) = (1 - \frac{N}{M}) \cdot 100\%. \tag{2}$$

C. Service Reliability

Reliability in the context of network layer packet transmissions is the percentage value of the packets successfully delivered to a given system entity within the time constraint required by the targeted service out of all the packets transmitted [45]. It is a combined perspective of end-to-end latency and packet loss rate. Service reliability in one time slot, is the percentage of the requests that are not rejected, and whose delay is below the

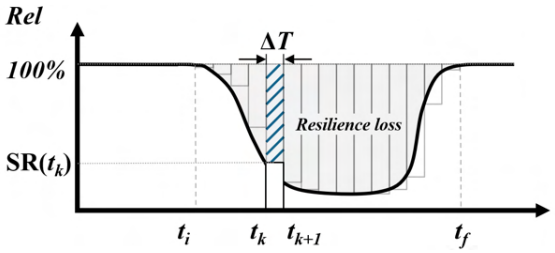


Fig. 6. The resilience triangle. The incident takes place at t_i. The system recovers at t_f. The gray part represents resilience loss of the k-th time slot.

latency requirement. Equations (3) and (4) give the calculation of service reliability SR.

$$\mathrm{SR}(t) = \left(\frac{\sum_{i=1}^{M} x_i \cdot y_i}{M}\right) \cdot 100\%. \tag{3}$$

$$y_i = \begin{cases} 0, & \text{if } x_i = 0 \text{ or } d_i > \text{latency requirement} \\ 1, & \text{otherwise} \end{cases} \tag{4}$$

D. Resilience metric

American National Academy of Science [49] defines resilience in a general way as the ability to prepare and plan for, absorb, recover from, or more successfully adapt to actual or potential adverse events. In this article, we give special attention to the ability of 5G to continue providing services that meet the requirements under an adverse event.

As proposed by Bruneau et al. [50], the resilience triangle can be used to quantify the resilience concept. As the reliability takes both the acceptance and service latency into consideration, we adopt service reliability as functional performance function. The resilience loss can be quantified by calculating the area of the degradation in the service reliability over time. Since the service reliability is discretized based on a time slot $]t_k, t_k + \Delta T]$ in the proposed simulation model as shown in Fig. 6, the estimated resilience loss of the network service under a certain incident is given as:

$$\mathrm{R} = \int_{t_i}^{t_f} [1 - Rel(t)]dt = \sum_{t=t_1}^{t_K} [100\% - \mathrm{SR}(t)]\Delta T \tag{5}$$

In Equation (5), t_i is the time when the incident starts, and t_f is the time when the service is completely recovered. If we discretize the impacted duration into K time slots of length ΔT (the same slots as we calculate the performance metrics), the continuous integral of resilience loss equals the sum of $[100\% - \mathrm{SR}(t_k)]\Delta T$.

E. Resource cost

In addition to the service performance, network resource allocation is also a critical concern. Over-allocating CPU resources to network services improves resilience performance in the presence of adverse events. Nevertheless, the over-booked resources will not only charge an extra fee but also consume more energy. As shown in Table IV, it takes 20 CPU units of resources to run a pod of DU or CU micro-service and 40 for a pod of UPF micro-service. When Kubernetes takes charge of auto-scaling, it can adjust the number of pod instances according to the traffic congestion situation and thus resulting in changing the resource allocation. To quantify resource cost, the resource usage metric

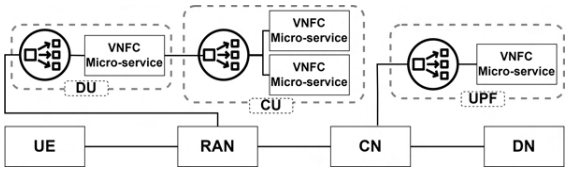


Fig. 7. Service function chain including 3 VNFs.

is introduced. We define in Equation (6), resource cost RC as the sum of the resource cost of each pod j in the 5G system, measured in CPU unit · second. For each pod, its resource utilization is the product of CPU resources that have been allocated to the pod and the pod lifetime $(t_{ej} - t_{0j})$. An ideal 5G system should have highly resilient performance while using fewer resources.

$$\mathrm{RC} = \sum_{j \in P} \mathrm{RC}_j = \sum_{j \in P} cpu_j (t_{ej} - t_{0j}) \tag{6}$$

VI. Case Studies

This section presents two case studies demonstrating how the proposed model can be applied to estimate network resilience performance.

The 5G network we consider is fully virtualized. This network hosts two network services. Service 1 is a latency-sensitive type application, with small size packet. A slight congestion can cause a severe latency requirement violation. Service 2 is an IoT-type application. Its latency requirement is relatively less strict. Both of these two services are considered uplink user-plane applications.

In the local RAN, Distributed Unit and Centralized Unit are used to provide connection to the Core Network. In the virtualized CN, UPF routes and forwards the packets to the internet. The service function chains are the same for these two services, as presented in Fig. 7.

We consider simplified network settings as given in Table IV. All parameters, including components of VNF, and their capacities eventually depend on the actual services suppliers provide. The service packet in the 5G network generated by the user will be processed locally by the micro-services (in order) in the RAN, then transmitted to CN, processed again, and finally delivered to the internet. We adopt a higher RAN functional split [51]. Then CU gathers more functions than DU, so it comprises more microservices. Since UPF is in the aggregated CN, each UPF pod allocates more CPU units to treat more packets in parallel. The processing time and transmission time are given in Table V. The packet processing time is proportional to the packet size, as we assume that one packet can be treated by one CPU unit only. With more resources allocated to VNFs in CN, UPF is capable of treating twice the packet than the VNFs in RAN, but all micro-services process packets at the same rate. The variant part of packet delay is the service delay in the micro-service queue. When a pod micro-service is overloaded (congested), the arrival packets will queue up and wait for available resources. When the queue reaches the maximum length, the arriving packet will be rejected. The parameters of processing time and transmission time, in reality, may be associated with uncertainty as well. Since the major interest of this study is to estimate the network service resilience to congestion effects due to traffic variation, and the uncertainty of processing time is assumed to stay unchanged during adverse events, these parameters are considered fixed values.

TABLE IV
SERVICE FUNCTION CHAIN COMPOSITION

	Number of instances	Capacity
VNFs in RAN		
DU	1 MS	infinite number of pods
MS	initially 1 pod	20 CPU units per pod
CU	2 MS	infinite number of pods
MS	initially 1 pod	20 CPU units per pod
VNF in CN		
UPF	1 MS	infinite number of pods
MS	initially 2 pod	40 CPU units per pod

TABLE V
NETWORK PROCESSES PARAMETERS

	Value	Remarks
Processing time		
Distributed Unit MS	short packet: 2 ms long packet: 4 ms	fixed time
Central Unit MSs	short packet: 2 ms long packet: 4 ms	fixed time
UPF MS	short packet: 2 ms long packet: 4 ms	fixed time
Transmission time		
Radio+transport	1.25 ms	fixed time
Service queue		
MS queue length	50 requests	first come first serve priority if applicable reject if time out
Maximal waiting time	1000 ms	

To achieve an accurate result, the model is programmed in Python with SimPy platform to run discrete event simulation. We take all iterations' average service latency, service reliability, and service resilience values generated by Monte Carlo Simulation. We limit the time duration to 60 seconds in order to estimate the timely dynamic response of the 5G network. The simulations are run 2000 times to get a confident result.

A. Resilience improvement by using Auto-scaling

To test the effectiveness of auto-scaling, we consider a network consisting of one RAN and one CN. No network slicing or priority is considered in this case. As introduced in Section.IV, auto-scaling is designed to be an approach to dynamically changing the cloud service scale to adjust to the load. The auto-scaling setup is given in Table VI. To create a new pod, it takes time to instantiate, run, and build the connection with other pods. This time is assumed to be an exponentially distributed random variable. The pod termination time and auto-scaling interval can be set by grace-period and sync-period flags in Kubernetes. The auto-scaling goal, threshold and stabilization window can be configured in Kubernetes. Kubernetes can configure HPA scaling behaviors by changing these parameters and create thus different scaling strategies. We compare different strategies: no auto-scaling (No AS), threshold-based basic Kubernetes built-in auto-scaling (Basic AS), and threshold-based basic auto-scaling combined with stabilization window (Win.AS) under four different traffic variations: a short traffic change, a long-term traffic variation, and two fluctuating traffic changes. The traffic arrival follows an exponential distribution, and service 1 always has twice the traffic arrival rate as service 2, as shown in Fig. 8. The irregularity of these traffic patterns increases one by one.

In No AS strategy, no auto-scaling is performed. 5G system will maintain the same scale during the traffic variation. In Basic

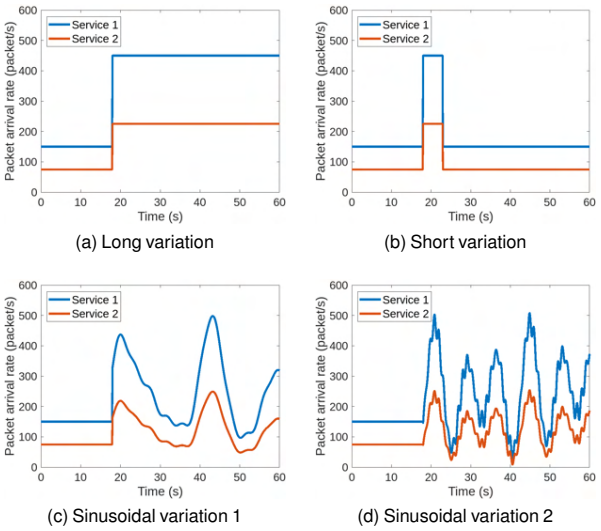


Fig. 8. Four traffic patterns with different arrival rate variations after t = 18s. (a) Long-term constant variation pattern, approximate entropy: 0.0108. (b) Short-term constant variation pattern, approximate entropy: 0.0207. (c) Sinusoidal (superposition) variation pattern 1, approximate entropy: 0.1019. (d) Sinusoidal (superposition) variation pattern 2, approximate entropy: 0.3676.

TABLE VI
NETWORK MANAGEMENT PARAMETERS

	Value	Remarks
Pod creation time	50 ms	exponential distribution
Pod termination time	15 s	fixed value
Auto-scaling interval	5 s	fixed value
Auto-scaling goal	50%	CPU utilization rate
Auto-scaling thresholds	30%&70%	down and up thresholds
Stabilization window	15 s	if applicable

AS strategy, the Kubernetes HPA sends a prob to detect the CPU utilization rate of each micro-service every 5 seconds. If the utilization rate of a micro-service is outside the threshold interval, a new scale of the micro-service will be calculated as follows:

$$\text{New scale} = \left\lceil \frac{\text{Current utilization}}{\text{Desired utilization}} \right\rceil \cdot \text{Current scale}. \quad (7)$$

If the new scale is greater than the current scale, a scaling-out decision is made to create more micro-service instances. Otherwise, a scaling-in decision is made to remove some existing instances. In Win. SA strategy, the HPA does not directly trigger a scaling action every 5 seconds. Instead, the decision is based on the resource utilization information during the stabilization window. In case 1, the window is 15 seconds. Therefore, a scaling-out decision is adopted if there are three successive scaling-out proposals during the last 15 seconds and it scales out to the smallest proposed scale. A scaling-in decision is triggered only after three successive scaling-in proposals and chooses the biggest estimated scale.

The simulation results of the three strategies under these four different traffic patterns are presented in Figs. 13, 14, and 15. In the simulation, the network suffers from abnormal traffic from both services' end users, starting from 18 seconds. Some packets will be rejected during the overloaded situation due to the micro-service queue length limit. Although some packets are not rejected, the packets of the latency-sensitive service, service 1, can not afford a long waiting time during the congestion, and its delivery time exceeds the latency limit.

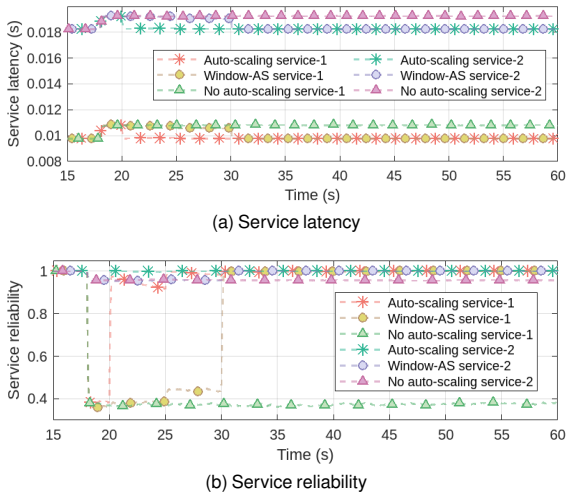


Fig. 9. Service latency and reliability under a long-term traffic variation (pattern a) with different management strategies (auto-scaling, stabilization window-based auto-scaling, and no auto-scaling).

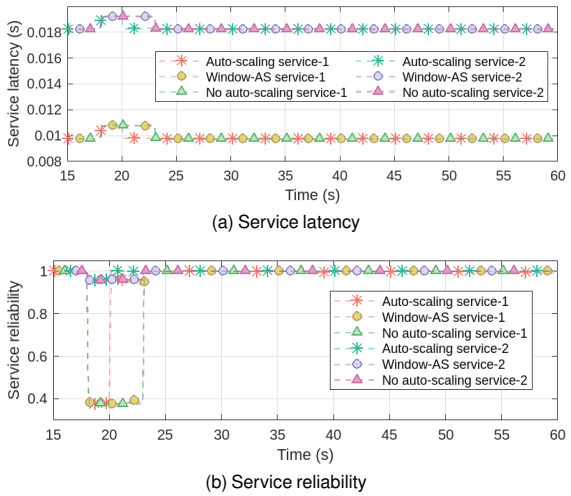


Fig. 10. Service latency and reliability under a short-term traffic variation (pattern b) with different management strategies (auto-scaling, stabilization window-based auto-scaling, and no auto-scaling).

Fig. 9 shows how service latency evolves with time. The ΔT is 0.1 seconds. We collect the packet delay d_i of each packet x_i during this ΔT and compute the corresponding Delay(t) of each interval according to Equation (1). In the long traffic change, Basic AS strategy immediately adds a necessary number of micro-service instances to keep the network service load at an acceptable level at 20 s. The window-based strategy takes a relatively long time but eventually relieves the congestion. While not taking any scaling action results in a large resilience loss in the service, especially for service 1, since it is more sensitive to latency. The model captures the service latency and the resilience loss evolution, as presented in Fig. 9.

Fig. 10 shows how service reliability evolves with time. We obtain the y_i by verifying if the latency requirement is satisfied for each packet x_i during this ΔT interval and then compute the corresponding service reliability SR(t) of each interval according to Equation (3). For a short-term traffic variation, Win.AS and No AS perform almost the same since the scaling decision is neglected in the former, and no scaling action is required in the latter. This leads to a congestion of the network for about 5 seconds. However, due to the randomness of packet arrival rates, a high resource utilization may occur from time to time and triggers window-based auto-scaling, causing a slightly high resource cost than No AS scenario. Basic AS reduces congestion time to two seconds. The resilience loss of both services is reduced, but it uses about a quarter more resources than other management strategies. The latency and reliability of the two services are compared in Fig. 10.

For the less fluctuating sinusoidal superposition traffic variations, Basic AS strategy makes a decision every 5 seconds to adapt to the traffic. Win.AS considers the traffic change during the last 15 seconds and is thus more "rigorous" to avoid frequent scaling in and out. The three strategies are compared in Fig. 11. The resilience loss of Basic AS is less at the beginning of traffic variation, but it performs even worse than No AS mechanism at the end of the simulation (at the third traffic peak). The resilience loss of Win.AS is almost the same as No AS case at the beginning, but it gradually performs better. The total resilience loss of Win.AS is less than Basic AS and No AS. Taking resource cost into consideration, Win.AS is the most economical solution to improve service resilience with a few additional cost.

In a more fluctuating traffic situation, the threshold-based Basic AS algorithm may not provide a satisfying solution. Indeed, the auto-scaling fails to make the correct decision as the expected scale at each decision moment changes. The Win.AS would prefer to decide not to change the scale during the fluctuation. As shown in Fig. 12, the differences in resource cost and resilience loss for the scenarios Win.AS and No AS are not much. The resilience of Basic AS is worse than No AS, and it costs the most. Basic AS takes the hazard of scaling out and in quickly but fails to provide enough service instances if there is a traffic increase just after a scaling-in triggered by a short-sighted decision. In fact, a scaling-in action would freeze the removed instance's resource for a while before being entirely killed to make sure all packet treatments are done before removing the instance. This results in a large resource cost and reduces the total available resources in the shared server that other micro-services can allocate. In this scenario, Win.AS performs the best in resilience but it is close to No As situation. Basic AS has the lowest resilience and the highest resource cost. If the fluctuation or irregularity of the traffic kepng increasing, it is possible that Win.AS performs worse than No AS, as the it may not always provide a suitable scale.

These strategies seem to perform differently under different traffic environments. Indeed, it is possible to implement artificial intelligence in Kubernetes so that the HPA parameters can be optimized according to the real-time traffic to get a better service performance. In our model, Kubernetes is assumed to be reliable throughout the simulation. However, in actual network installation, if Kubernetes fails, the HPA function becomes unavailable. In such a scenario, the Basic AS and Win.AS will perform the same as No AS.

Although this study focuses on short timescale traffic variation, it can be extended to evaluate network service resilience under a long timescale traffic variation. The long-timescale traffic variation can be seen as slices of short-timescale traffic variation, but the traffic often fluctuates less in each time slot. Therefore, the auto-scaling can better adjust to the traffic, and the network service is thus more resilient to a long timescale traffic variation.

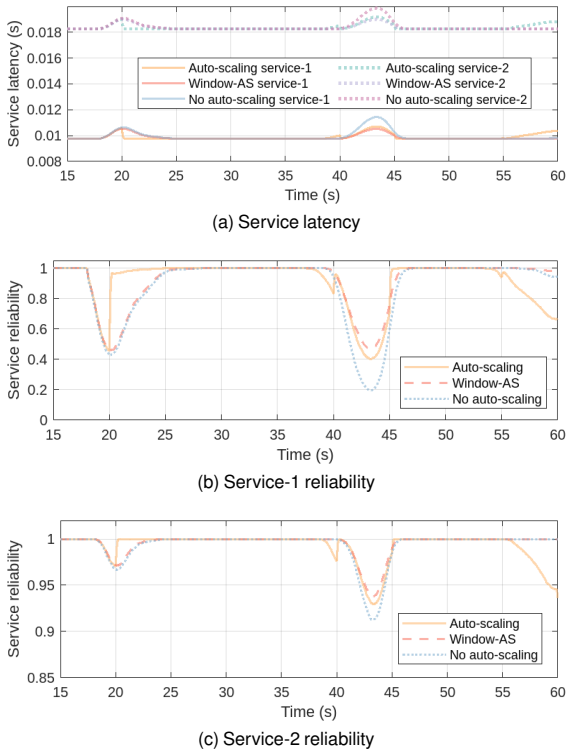


Fig. 11. Service latency and reliability under sinusoidal superposition traffic variation (pattern c) with different management strategies (auto-scaling, stabilization window-based auto-scaling, and no auto-scaling).

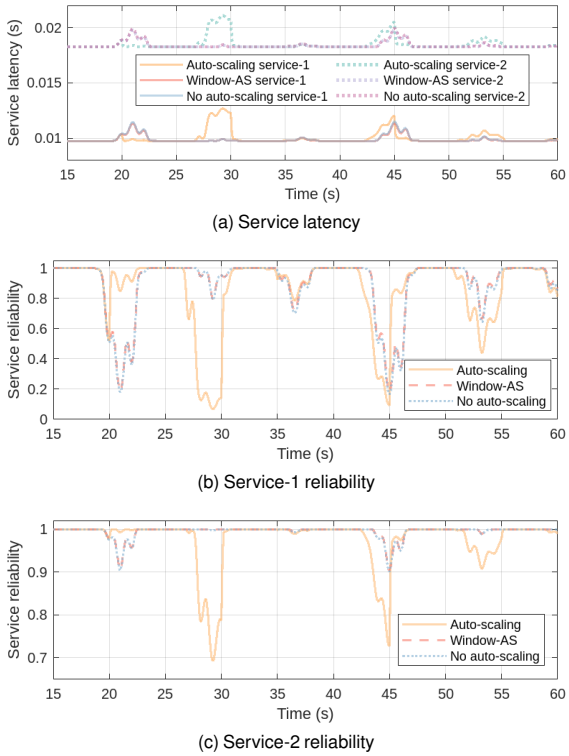


Fig. 12. Service latency and reliability under sinusoidal superposition traffic pattern variation (pattern d) with different management strategies (auto-scaling, stabilization window-based auto-scaling, and no auto-scaling).

B. Resilience with network service isolation

Without isolation, the network resources are shared by all network services. By introducing network slicing, network

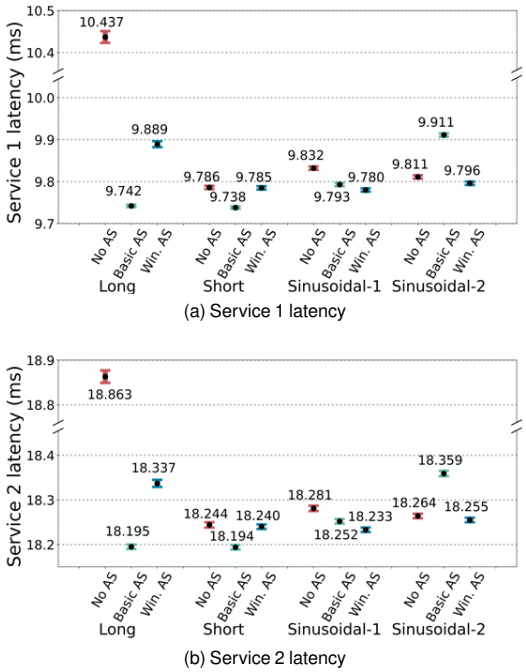


Fig. 13. Service 1 (a) and Service 2 (b) latency values and confidential intervals in case 1.

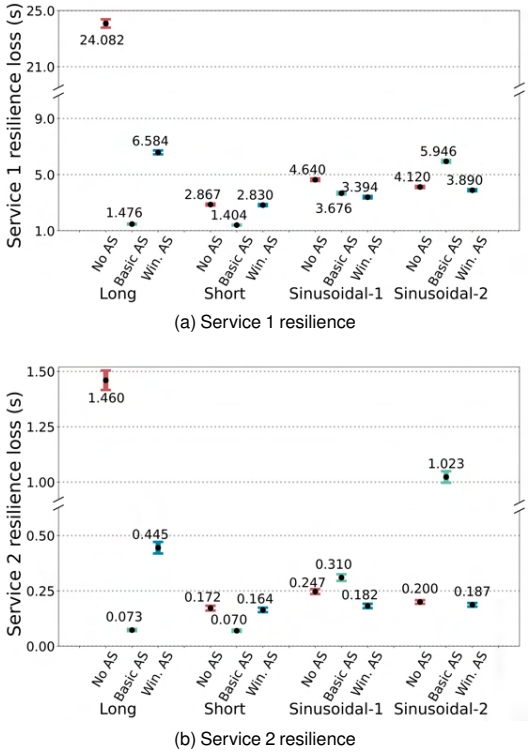


Fig. 14. Service 1 (a) and Service 2 (b) resilience loss values and confidential intervals in case 1.

resources are sliced. They are assigned to different usages so that different services use the customized VNFs belonging to their slice. When the end user starts a communication, the PDU session establishment is informed of which VNF instances are used when delivering data packets.

Case study 2 considers a no-autoscaling 5G system composed

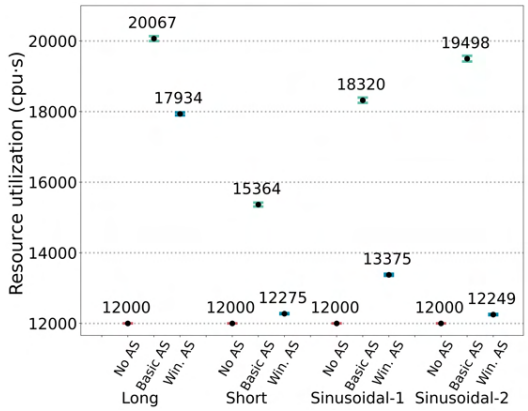
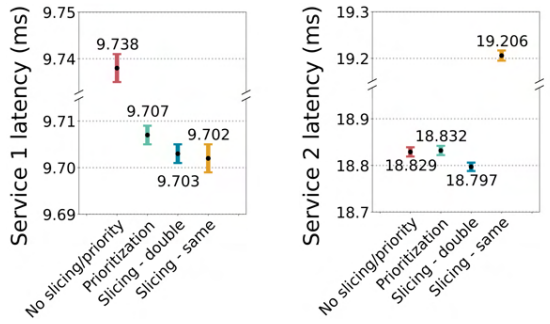
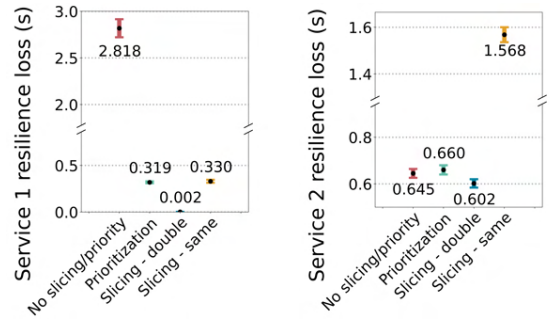


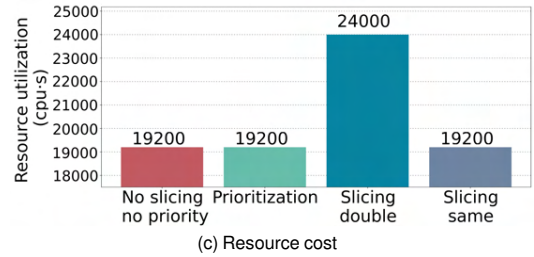
Fig. 15. Resource cost values and confidential intervals in case 1.



(a) Service 1 and Service 2 latency



(b) Service 1 and Service 2 resilience loss



(c) Resource cost

Fig. 16. Service latency (a), reliability (b) and resource cost (c) in case 2.

of four identical distributed local RAN (1-4) and a centralized CN (5). In zone 1, only Service 1 end users are connected and always generate regular traffic. In zones 2, 3, and 4, only Service 2 end users are connected, and they start to change the traffic arrival rate by triple (short traffic variation for 10 seconds). If no network slice is applied, in RAN, each service has its own VNF since, geographically, they use different physical infrastructure. They share the same UPF instance in the centralized CN. If priority is applied, then the latency-sensitive service-1 packets are treated with priority in the shared VNF. If slicing is applied, then in CN, each service has its UPF instance, and they are managed separately. These UPF instances are assigned to end users when building PDU session for the connection between user and the network.

Four scenarios are compared: no slicing or priority network, prioritization network, and two sliced networks. We consider two slicing partitions. The first partition is to create two separate UPF instances for services 1 and 2, each using the same amount of resources as in the shared UPF. Therefore, we double the initial resource. The second partition is to create two different sized UPF instances with different resource allocations according to the initial service traffic. The total resources of the two UPFs equal the single shared UPF.

Fig. 16 shows the latency and resilience results. Prioritization helps largely reduce critical service resilience loss without allocating more resources as it treats the latency-sensitive packets first so that most of them do not exceed time limit. Dedicated slices also keep the latency-sensitive service from anomalies from services. When failure is injected into service-2 end users, service-1 is protected by virtual isolation. If each service has its UPF instance the same size as a shared one, then the performance of both services is better than without slicing, even under adverse traffic change. However, it takes relatively more resources (about a quarter in Case 2). If we keep the initial resource the same, the resource margin for each service in normal operation mode is less than in a shared network. Service-1 has more chance to overload the slice by the randomness of the packet arrival. This explains a greater service-1 resilience loss than the doubled initial resource slicing. For service-2, as the resource margin is reduced, it is more congested than the no slicing scenario during traffic variation, resulting in a greater resilience loss.

According to the results of case 2, with a generous budget, the doubled initial resource slicing is preferred during a traffic variation. Otherwise, prioritization is favored.

VII. Conclusion

This paper presents the hierarchical Petri Net-based model to estimate 5G network service resilience performance. This model is capable of capturing the virtualized network characteristics and dynamic behaviors. We introduce how we apply it to quantify network resilience by combining the aspect of service latency and service reliability. Traffic changes are selected as the primary threats to network service resilience. Kubernetes-based management and orchestration systems, network slicing, and prioritization are studied as potential solutions to increase service resilience. A resilience analysis is carried out by Monte Carlo simulation. The results show that: 1) auto-scaling can improve resilience during some traffic variations by dynamically changing the scale of the network setup, but the algorithm or strategy should be carefully designed to cope with the different patterns of traffic anomalies; 2) network slicing, though requires more resources, can effectively protect a network service from incidents happening outside the slice; 3) service priority can be applied to guarantee the overall network resilience of all network services with limited resource allocation budget. To the best of our knowledge, this is the first model to estimate service resilience in a short timescale. This model gives valuable

information on network design, operation, and control from a resilience perspective to the service providers and operators.

Although some existing simulators may also estimate the service performance, the Petri Net-based approach we propose in this work, which by focusing on stochastic processes, queue models, and priority queue models, is tailored and adapted to the specific problem and allows to represent and capture the dynamic behavior and the relationship between different network elements. These existing simulators consider the whole message process for each VNF and link. They could be less efficient for simulating and estimating the congestion and management problem than our approach. Besides, the 5G model they propose will not necessarily be the same as the 5G installation chosen by operators. Finally, to test the performance using existing simulators, additional parts such as a traffic generator and a K8S model will be needed.

In future work, more precise parameters will be collected to simulate a use case from the vertical industry to evaluate the resilience based on the real service requirements. Certain parameters may be challenging to obtain directly from simulations or experiments. For example, extracting the processing time of each network element from an end-to-end test may not be easy due to various limitations. In addition, the management parameters can also differ from one service provider to another, which can impact service resilience. Nevertheless, we can modify these parameters in the model to assess their impact on the overall system resilience, e.g., for determining the most contributing parameters to the service resilience. This is usually conducted with global sensitivity analysis methods [52] and is outside the scope of the present study.

A control plane network model will be considered to simulate the network signaling, which is critical in evaluating the network service resilience in use cases such as high-speed train services where frequent signaling requests are expected. Although the proposed model is currently used for off-line resilience estimation to provide suggestions to anticipate traffic change, it is possible to implement or integrate the model with operational intelligence, such as NWDAF in 5G CN for real-time deployment. By doing so, the model could estimate the network service resilience based on real-time metrics collected from the system and provide feasible and efficient management suggestions for enhancing resilience.

Since our approach can also be applied to all types of 5G/6G networks that will be installed, future work will also undertake performance testing using an actual virtualized telecommunication network, once the fully virtualized commercial or experimental network becomes available.

REFERENCES

[1] NGMN Alliance, "Perspectives on Vertical Industries and Implications for 5G," Jun, 2016.

[2] D. Jiang, and G. Liu, "An Overview of 5G Requirements," *5G Mobile Communications*, pp.3–26, 2017.

[3] F. Z. Yousaf, M. Bredel, S. Schaller and F. Schneider, "NFV and SDN—Key Technology Enablers for 5G Networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2468–2478, Nov. 2017.

[4] J. Jiang, J. Lu, G. Zhang and G. Long, "Optimal Cloud Resource Auto-Scaling for Web Applications," in *proc. the 13th IEEE/ACM Int. Symp. on Cluster, Cloud and Grid Computing*, 2013, pp. 58—65.

[5] L. Toka, G. Dobreff, B. Fodor and B. Sonkoly, "Adaptive AI-based auto-scaling for Kubernetes," in *proc. the 20th IEEE/ACM Int. Symp. on Cluster, Cloud and Grid Computing(CCGRID)*, 2020, pp. 599–608.

[6] G. Pintér and I. Felde, "Analyzing the Behavior and Financial Status of Soccer Fans from a Mobile Phone Network Perspective: Euro 2016, a Case Study," *Information*, vol. 12, no. 11, p. 468, Nov. 2021.

[7] D. Rico and P. Merino, "A Survey of End-to-End Solutions for Reliable Low-Latency Communications in 5G Networks," *IEEE Access*, vol. 8, pp. 192808–192834, 2020.

[8] D. Hutchison and J. P. Sterbenz, "Architecture and design for resilient communication systems," *Comput. Commun.*, vol. 131, pp. 13–21, 2018.

[9] J. P. Sterbenz et al., "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Comput. Netw.*, vol. 54, no. 8, pp. 1245—1265, 2010.

[10] J. Rak and D. Hutchison, *Guide to disaster-resilient communication networks*,Cham, Switzerland: Springer, 2020.

[11] C. Esposito et al., "On the Disaster Resiliency within the Context of 5G Networks:The RECODIS Experience," 2018.

[12] A. Dutta and E. Hammad, "5G Security Challenges and Opportunities: A System Approach," 2020 IEEE 3rd 5G World Forum (5GWF), Bangalore, India, 2020, pp. 109-114.

[13] A. Mauthe et al., "Disaster-resilient communication networks: Principles and best practices," in *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, Halmstad, Sweden, 2016, pp. 1-10.

[14] D. Santos, A. De Sousa, C. Mas-Machuca and J. Rak, "Assessment of Connectivity-Based Resilience to Attacks Against Multiple Nodes in SDNs," *IEEE Access*, vol. 9, pp. 58266-58286, 2021.

[15] A. De Sousa, "Improving the Connectivity Resilience of a Telecommunications Network to Multiple Link Failures Through a Third-Party Network," in *2020 16th International Conference on the Design of Reliable Communication Networks DRCN 2020*, Milan, Italy, 2020, pp. 1-6.

[16] M. K. Awad, A. A. M. R. Behiry and E. A. Alrashed, "A Robust and Resilient Load Balancing Framework for SoftRAN-Based HetNets With Hybrid Energy Supplies," *IEEE Trans. Netw. Service Manage*, vol. 17, no. 3, pp. 1403-1417, Sept. 2020.

[17] C. Liu, Y. Xie, H. Li, Y. Wang and Y. Zhang, "A Framework for Assessing the Resilience of 5G Mobile Communication Networks," 2022 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2022, pp. 1077-1081.

[18] F. Nakayama, P. Lenz and M. Nogueira, "A Resilience Management Architecture for Communication on Portable Assisted Living," *IEEE Trans. Netw. Service Manage*, vol. 19, no. 3, pp. 2536-2548, Sept. 2022.

[19] Kang, F. He and E. Oki, "Resilient Virtual Network Function Allocation with Diversity and Fault Tolerance Considering Dynamic Requests," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, Budapest*, Hungary, 2022.

[20] N. Hyodo, T. Sato, R. Shinkuma and E. Oki, "Resilient Virtual Network Function Placement Model Based on Recovery Time Objectives," in *2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR)*, Newark, NJ, USA, 2020, pp. 1-7.

[21] M. Di Mauro, G. Galatro, M. Longo, F. Postiglione and M. Tambasco, "Comparative Performability Assessment of SFCs: The Case of Containerized IP Multimedia Subsystem," *IEEE Trans. Netw. Service Manage*, vol. 18, no. 1, pp. 258–272, Mar. 2021.

[22] H. Farooq, M. S. Parwez and A. Imran, "Continuous Time Markov Chain Based Reliability Analysis for Future Cellular Networks," in *proc. IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1–6.

[23] J. Bai, X. Chang, F. Machida, L. Jiang, Z. Han and K. S. Trivedi, "Impact of Service Function Aging on the Dependability for MEC Service Function Chain," *IEEE Trans. Dependable Secure Comput.*, early access. doi: 10.1109/TDSC.2022.3150782.

[24] S. Agarwal, F. Malandrino, C. F. Chiasserini and S. De, "VNF Placement and Resource Allocation for the Support of Vertical Services in 5G Networks," *IEEE/ACM Trans. Netw.* vol. 27, no. 1, pp. 433–446, 2019.

[25] Q. Ye, W. Zhuang, X. Li and J. Rao, "End-to-End Delay Modeling for Embedded VNF Chains in 5G Core Networks," *IEEE Internet of Things J.*, vol. 6, no. 1, pp. 692–704, Feb. 2019.

[26] J. Li, W. Shi, Q. Ye, N. Zhang, W. Zhuang and X. Shen, "Multiservice Function Chain Embedding With Delay Guarantee: A Game-Theoretical Approach," *IEEE Internet of Things J.*, vol. 8, no. 14, pp. 11219–11232, Jul. 2021.

[27] U. Singh et al., "Coalition Games for Performance Evaluation in 5G and Beyond Networks: A Survey," *IEEE Access*, vol. 10, pp. 15393–15420, 2022.

[28] N. -T. Dinh and Y. Kim, "An Efficient Reliability Guaranteed Deployment Scheme for Service Function Chains," *IEEE Access*, vol. 7, pp. 46491–46505, 2019.

[29] K. S. Ghai, S. Choudhury, and A. Yassine, "A stable matching based algorithm to minimize the end-to-end latency of edge NFV," *Procedia Comput. Sci.*, vol. 151, pp. 377—384, 2019.

[30] L. Dong, N. L. S. da Fonseca and Z. Zhu, "Application-Driven Provisioning of Service Function Chains Over Heterogeneous NFV Platforms," *IEEE Trans. Netw. Service Manage*, vol. 18, no. 3, pp. 3037–3048, 2021.

[31] Y. Wu, W. Zheng, Y. Zhang and J. Li, "Reliability-Aware VNF Placement Using a Probability-Based Approach," *IEEE Trans. Netw. Service Manage*, vol. 18, no. 3, pp. 2478–2491, Sep. 2021.

[32] P. K. Thiruvasagam, A. Chakraborty, A. Mathew and C. S. R. Murthy, "Reliable Placement of Service Function Chains and Virtual Monitoring Functions With Minimal Cost in Softwarized 5G Networks," *IEEE Trans. Netw. Service Manage*, vol. 18, no. 2, pp. 1491–1507, June 2021.

[33] S. Schneider, A. Sharma, H. Karl and H. Wehrheim, "Specifying and Analyzing Virtual Network Services Using Queuing Petri Nets," in *proc. IEEE/IFIP Netw. Oper. Manag. Symp (IM)*, 2019, pp. 116–124.

[34] L. Rui, X. Chen, Z. Gao, W. Li, X. Qiu and L. Meng, "Petri Net-based reliability assessment and migration optimization strategy of SFC," *IEEE Trans. Netw. Service Manage*, vol. 18, no. 1, pp. 167-–181, Mar. 2021.

[35] X.-Y. Li, Y. Liu, Y.-H. Lin, L.-H. Xiao, E. Zio, and R. Kang, "A generalized Petri net-based modeling framework for service reliability evaluation and management of cloud data centers," *Rel. Eng. Syst. Saf.*, vol. 207, Mar. 2021, Art. no. 107381.

[36] R. Li, B. Decocq, A. Barros, Y. Fang and Z. Zeng, "Petri Net-Based Model for 5G and Beyond Networks Resilience Evaluation," in *Proc. 25th Conf. Innov. Clouds, Internet Netw. (ICIN)*, Mar. 2022, pp. 131–135.

[37] R. Li, B. Decocq, A. Barros, Y. Fang and Z. Zeng, " A Petri Net-based model to Study the Impact of Traffic Changes on 5G Network Resilience," in *Proc. Eur. Saf. Rel. Conf. (ESREL)*, Dublin, Ireland, Sep. 2022.

[38] T. Taleb, I. Afolabi, K. Samdanis and F. Z. Yousaf, "On Multi-Domain Network Slicing Orchestration Architecture and Federated Resource Control," *IEEE Network*, vol. 33, no. 5, pp. 242–252, Sep. 2019.

[39] S. D. A. Shah, M. A. Gregory and S. Li, "Cloud-Native Network Slicing Using Software Defined Networking Based Multi-Access Edge Computing: A Survey," *IEEE Access*, vol. 9, pp. 10903–10924, 2021.

[40] *Description of Network Slicing Concept, NGMN 5G P1 Requirements & Architecture, Work, Stream End-to-End Architecture, Version 1.0*, NGMN Alliance, Jan. 2016.

[41] P. Rost et al., "Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 72–79, May. 2017.

[42] R. Ferrus, O. Sallent, J. Pérez-Romero, and R. Agusti, "Management of network slicing in 5G radio access networks: Functional framework and information models," 2018, *arXiv:1803.01142*.

[43] M. Ersue, "ETSI NFV management and orchestration - An overview," in *Proc. 88th IETF Meeting*, 2013.

[44] R. Mijumbi, J. Serrat, J. -l. Gorricho, S. Latre, M. Charalambides and D. Lopez, "Management and orchestration challenges in network functions virtualization," *IEEE Commun. Mag.*, vol. 54, no. 1, pp. 98–105, 2016.

[45] *5G; Service requirements for the 5G system (Release 16)*, document 3GPP TS 22.261, 3GPP, Jan. 2022.

[46] *Service requirements for enhanced V2X scenarios (Release 17)*, document 3GPP TS 22.186, 3GPP, Apr. 2022.

[47] *Mobile communication system for railways (Release 16)*, document 3GPP TS 22.289, 3GPP, Nov. 2020.

[48] *Management and orchestration; 5G performance measurements (Release 16)*, document 3GPP TS 28.552, 3GPP, Oct. 2022.

[49] *Disaster resilience: A national imperative* 2012. The National Academies Press, Washington, DC, USA, 2012.

[50] M. Bruneau, S. Chang, R. Eguchi, G. Lee, T. D. O'Rourke, A. M. Reinhorn, M. Shinozuka, K. Tierney, W. A. Wallace, and D. Von Winterfeld, "A Framework to Quantitatively Assess and Enhance the Seismic Resilience of Communities," *Earthq. Spectra*, vol. 19, pp. 733-–752, Nov. 2003.

[51] *Study on new radio access technology: Radio access architecture and interfaces (Release 14)*, document 3GPP TR 38.801, 3GPP, Mar. 2017.

[52] B. Iooss and P. Lemaître, "A review on global sensitivity analysis methods," in *Uncertainty Management in Simulation-Optimization of Complex Systems.* Boston, MA, USA: Springer, 2015, pp. 101–122.

Rui Li received the degree in engineering (CTI) from Ecole Centrale Paris (Dual degree) in Feb 2020 the a M.S. degree in industrial engineering from Beihang University in Jan 2020.

He is currently pursuing the Ph.D. degree in complex system engineering at CentraleSupélec, University of Paris-Saclay, Gif-sur-Yvette, France. He is also a CIFRE fellow working in Orange Innovation. His current research interests include complex networked system modeling, telecommunication network system performance and resilience analysis, 5G network service optimization.

Bertrand Decocq received the Ph.D. degree in Computer Science and Operations Research, and ENSIIE engineering school in 1997. He joined France Telecom (now Orange) since 1997.

He is currently team and project manager and is a member of Orange expert community on future networks. He started to work on mobile network resilience in 2014 and is now working with a chair of Centrale Supelec on "Risks and Resilience of Complex Systems" dealing mainly with interdependencies between critical infrastructures from resilience and maintenance perspectives. He is also in charge of a partnership with EDF R&D on cross resilience between energy and telecommunication networks.

Anne Barros received the Ph.D. degree from the University of Technology of Troyes, Troyes, France, in 2003. She held a professorship position with the University of Technology of Troyes from 2003 to 2014. She was a Full-Time Professor with NTNU, Trondheim, Norway, from 2014 to 2019.

She is currently a Professor of reliability and maintenance modeling with the Ecole CentraleSupélec, University of Paris-Saclay, Gif-sur-Yvette, France. She is also the Head of the research group and an Industrial Chair with the CentraleSupélec, with the ambition to prove reliability assessment and maintenance modeling methods for complex systems. Her research interests include degradation modeling, prognostics, condition based, and predictive maintenance.

Yiping Fang received the Ph.D. degree in industrial engineering from École Centrale Paris, Paris, France, in 2015.

He is currently an Assistant Professor with the Chair Risk and Resilience of Complex Systems, Laboratoire Génie Industriel, CentraleSupélec, Université Paris-Saclay, Paris. He was the Postdoc Research Fellow with ETH, Zurich, Switzerland, from 2015 to 2017. His research interests include the study and development of advanced computational methods for risk, reliability, and resilience analytics of critical cyber-physical systems (including smart grids, intelligent transportation, and 5G-and-beyond-systems), stochastic and robust optimization, risk and decision analysis, and machine learning.

Zhiguo Zeng received the Ph.D. degree in reliability engineering from Beihang university in 2016.

He is currently an Assistant Professor at CentraleSupélec, Université Paris-Saclay, France. His research focuses on the characterization and modeling of the failure/repair/maintenance behavior of components, complex systems and their reliability, maintainability, prognostics, safety, vulnerability and security. Dr. ZENG is an author/co-author of more than 50 papers in highly recognized international journals and conferences (including 32 journal papers indexed in Web of Science). He is editorial board member of International Journal of Data Analysis Techniques and Strategies, and the leading guest editor of the special issue on "Dependent failure modeling" of the journal Applied Science.

Reliability challenges of 5G and Beyond networks applications in high-speed trains

Rui Li, Bertrand Decocq

Orange Innovation, Orange Labs, France. E-mail: {rui.li;decocq.bertrand}@orange.com

Yiping Fang, Zhiguo Zeng, Anne Barros

Chair Risk and Resilience of Complex Systems, Laboratoire Génie Industriel, Centralesupélec, Université Paris-Saclay, France. E-mail: {yiping.fang;zhiguo.zeng;anne.barros}@centralesupelec.fr

5G and Beyond networks are expected to be reliable solutions to support new and complicated wireless communication scenarios. As high-speed railway systems are booming all around the world, they bring about novel challenges to the 5G and Beyond networks to support high mobility usage. Railway communication functionality has higher performance requirements than other use cases. These requirements will be satisfied by providing an ultra-reliable 5G and Beyond system and seamless handover procedures under high mobility. On the one hand, the system faces failures from its virtual and physical layers. On the other hand, high mobility creates radio issues on handover and interrupts network services. Network service reliability performance can be guaranteed by continuous end-to-end user plane connectivity. This connectivity is maintained by successful handover during radio zone changes. Handover is a signaling process in the control plane. Therefore, the railway network service reliability analysis requires a combined perspective of user and control planes. This paper investigates the possible challenges of high-speed railway network service reliability and examines the impacts of various factors. By using discrete event simulation, we calculate the onboard network communication service reliability during its mission. The impacts of different telecommunication network deployments on network and service reliability are compared. Simulation results provide insights into estimating service performance and propose feasible solutions to improve service continuity and reliability for railway operators and network providers.

Keywords: Mobile network, 5G and Beyond, Reliability, High-speed trains, Discrete event simulation.

1. Introduction

For more than 20 years, ground-to-train communication has relied on the GSM-R system based on 2G. The International Union of Railway (UIC) decides to launch a new system, Future Railway Mobile Communication System (FRMCS), to replace it. As pointed out by UIC (2020), the goal is to usher in 5G for rail networks. GSM-R, often reinforced with redundancy in the application, has been, so far, one of the most reliable systems (He et al. (2016)). Although GSM-R is still a universal solution for the communication between the train and control center, there are many reasons to upgrade this system, such as the end of the GSM-R system life-cycle and the need to improve the quality of service and quality of experience (Masur and Mandoc (2009)).

5G and Beyond is undoubtedly the most advanced telecommunication system that will enhance the quality of railway services. The 5G New Radio (5G NR) extends to a higher spectrum band (Niu et al. (2015)), enabling a higher data transfer rate. The 5G Core will be fully virtualized (Bonati et al. (2020)), providing a flexible and tailored network to train services.

Nevertheless, just as GSM needs to be upgraded with further enhancements specific to the requirements to become GSM-R, 5G and Beyond networks need to be carefully implemented and designed to adjust to the specific requirements of railroad operation.

According to 3GPP (2022), seamless communication is crucial for train control service as it conveys important signals guaranteeing the operation of trains. Onboard, seamless communication is also required to provide high-quality services.

However, communication in the high-speed railway scenario faces many challenges. As discussed by Fan et al. (2016), most of these challenges could be grouped under four categories:

accurate channel estimation, advanced signal processing, optimized network deployment, and effective mobility management. Since this work addresses reliability-related issues, we focus mainly on network deployment and mobility management. The failure of the network facility is one of the main reasons a train loses its communication service since it would need to connect to different base stations during its movement. The faster a train moves, the faster it needs to change the anchoring base stations, thus the more network elements it uses during a given time. In network management, HandOver (HO) procedure can be another crucial reliability challenge. As 5G and Beyond networks introduce a high spectrum band, the dense small-cell (Al-Falahy and Alani (2017)) layout increases HO frequency for high mobility end-users. HO signaling procedure reliability becomes thus more important for providing a seamless connection to high-speed trains.

Some works have addressed the 5G reliability problem, considering low-mobility or non-mobility users (Farooq et al. (2015); Qu et al. (2018); Thiruvasagam et al. (2022)). Some works have investigated the HO process management under high mobility and sought to find a better way to avoid wrong HO, failed HO, or missed HO (Song et al. (2014); El Banna et al. (2020); Sönmez et al. (2020); Tanveer et al. (2022)). Nevertheless, little attention has been paid to the impact of network infrastructure failure and HO procedure failure on the reliability and availability of high-speed train communication service.

This paper aims to take up the challenges of 5G and Beyond reliability analysis in high-speed train applications. We developed a 5G and Beyond network element model and a moving train model. Combined together, these two models reflect the real communication-related problems a train could encounter during its mission. The reliability and availability of 5G and Beyond network and train telecommunication service are estimated by carrying out discrete event simulations. The main contributions of this work are the following:

- Main challenges in high-speed train communication are discussed
- Moving train model and network component model are developed to represent their state changes
- Handover procedure and re-establishment procedure are both considered for high-speed train scenario
- The perspectives of reliability and availability from the network operator and high-speed train service user are compared

The paper has been organized in the following way. We briefly introduce the high-speed train service problem in section 2. In section 3, we present the 5G and Beyond network model and the train model. A high-speed train mission scenario is presented, and the simulation results are given in section 4. Finally, section 5 concludes the work with some remarks and outlines future works.

2. Problem statement

We consider a generic 5G and Beyond network composed of the Radio Access Network (RAN) and the Core Network (CN). The network architecture is presented in Figure 1. RAN, which transmits, receives, converts and processes the signal, comprises a set of gNB base stations, and each is composed of Radio Units (RUs), Distributed Units (DUs) and Central Units (CUs). The CN, consisting of different Virtual Network Functions (VNFs), that take charge of aggregation, authentication, service control, etc., is divided into the User Plane (UP) with User Plane Function (UPF), and the Control Plane (CP), including VNFs such as Access Management Function (AMF), Session Management Function (SMF), Data Management (UDM), Authentication Server Function (AUSF), etc. As an end-user, a train will connect to the RU with the best signal that covers the area it passes via a 5G NR air interface. Once the train is registered to the network, it will request a Protocol Data Unit (PDU) session to start an end-to-end UP connectivity between the UE and Data Network (DN). This connectivity is supported by User Plane, that is, RU, DU, CU-UP, UPF, and the links between them.

The main problem addressed in this work is the reliability and availability-related challenges

of communication services applied to high-speed trains. More precisely, a train is considered connected to the internet if the user is registered to the network and it has initiated a PDU session and the whole user plane allocated by the PDU session is reachable and available to the train. We distinguish in the paper two kinds of connection failure: the failure related to User Plane failure and the failure related to reachability.

2.1. *User Plane failure*

When a train starts to travel on the railway, we assume that it is already registered to the 5G and Beyond network. While the train is running, failures from different parts of the network will impact the communication service in different ways:

- If the gNB facility (including RU, DU, and CU-UP) fails, the train directly loses the connection to DN. There are two possible solutions to reconnect to the DN. If there is another available gNB covering the train, then the train will try to re-establish the connection via this available gNB by a re-establishment procedure. Otherwise, the train becomes unconnected and untraceable. Communication service is stopped. The train will wait until the gNB is repaired or until it enters an available gNB coverage area.
- If the UP in CN fails, i.e., UPF-UP fails, the end-to-end communication service is interrupted, yet the train is still connected to the gNB. The communication service resumes after the recovery of CN UP.

The Re-establishment procedure (3GPP (2021)) is simplified by considering the call flow involving only the RU, DU, CU, AMF, and UPF.

2.2. *Reachability failure*

Since the train is in high mobility, the RU to which it connects can only serve a specific area, as shown in the radio layout example in Figure 2. To guarantee a seamless connection, the train regularly changes the connected RU by HO process at the overlapping covered by multiple RUs. There are different types of HO regarding the implementation and layout of 5G (3GPP (2021)). In the scope of this work, we consider two of them:

- Inter gNB-DU and Intra gNB-CU Handover: In this HO procedure, the new and old gNB-DUs are connected to the same CU. The signaling message will not necessarily be sent to CN. This procedure will involve messaging over the source and target RUs, DUs, and their CU.
- Inter gNB-CU Handover: In this HO procedure, the signaling will involve messaging over the source and target gNBs (including RUs, DUs, CU), AMF, and UPF.

If the HO procedure fails, the train stays connected to the previous RU. When the RU is no longer reachable to the train, the train will be disconnected from the network and need to re-establish the connection to resume the communication service.

2.3. *Availability and reliability*

To analyze the reliability challenges, the reliability-related terms should be well defined. For the considered network, we define the availability and reliability from both network and high-speed train communication service perspectives:

- We define network availability as the percentage value of the amount of time the network operator can provide end-to-end service and response to CP signaling messages everywhere by using the 5G and Beyond network deployed in a considered area, divided by the total considered time.
- We define network reliability as the ability of the 5G and Beyond network to provide end-to-end connection and response to CP signaling messages everywhere in a considered area. We measure network reliability using the Mean Time To Failure (MTTF) of the considered network system.
- We define train network communication service availability as the percentage value of the amount of time the end-to-end communication service is delivered, divided by the amount of time the train network communication service is expected to be delivered.
- We define network communication service reliability as the ability of the communication service to perform as required for a given time in-

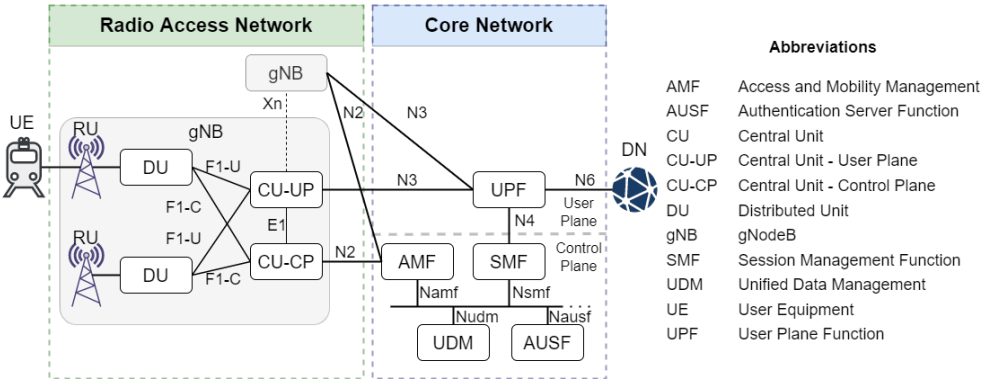


Fig. 1. 5G network architecture.

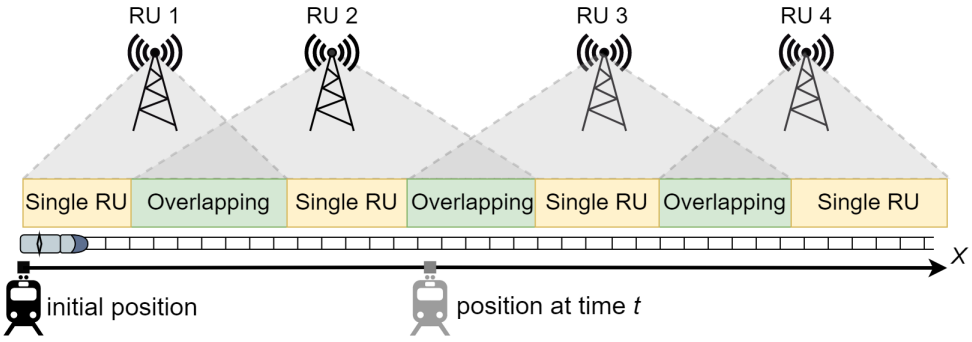


Fig. 2. An example of 5G gNB RU layout along a section of railway.

terval under given conditions. We describe network communication reliability using MTTF of the train communication service.

3. Discrete event simulation model

We separate the considered system into two parts: the network facility, "Telecommunication network", and the service user, "high-speed train". A telecommunication network is a set of network functions composed of virtualized applications and physical resources. The train, whose position is known at a given moment, will consume the service the reachable network functions provide.

3.1. *5G network model*

The 5G and Beyond network comprises different elements, such as DU, CU, and AMF in Figure. 1. We assume they all have similar behavior as shown in Figure. 3. They all start from a working state (W) and may fall into a failed state (F) due to software and hardware reasons. This failure will be detected and identified (N). Finally, it will be either fixed automatically in the case of software and application issues or repaired manually (R). When the element is not in the state (W), all end-users relying on this element fail to use the element, leading to a service connection or a signaling procedure (re-establishment or HO) failure.

3.2. *Train model*

From an end-user's perspective, the train is always in a moving situation. We divide the train's mission into a series of rounds. Each round is represented by Figure. 4. A round starts from the state where the train is initially connected to i^{th} RU.

If the train runs into a Single RU area, it will stay at the connected state unless the connection fails (some of the network elements it uses are

in states (F)). If the failure is due to UPF-UP, the train can return to the connected state when UPF-UP is repaired. If the gNB fails, the train will try to re-establish the connection to i^{th} RU if the failed gNB is repaired, and the train then goes back to the connected state. If the train fails to re-establish the connection, it will remain disconnected until a successful re-establishment to j^{th} RU when entering an overlapping zone, where $j \neq i$.

If the train runs into an overlapping area, it can request HO when a better signal is found. If the HO procedure succeeds, the train will connect to j^{th} RU, where $j \neq i$. If the HO procedure fails, the train will retry HO until the train runs outside of the i^{th} RU covering zone. Then the train will re-establish the connection instead of requiring HO. In this area, the connection is also at risk of facility failure. As the train runs in an overlapping area, another RU always exists. Should i^{th} RU fails, it would immediately try to re-establish the connection to the other RU, j^{th} RU, where $j \neq i$.

Both the HO and re-establishment processes change the state of a train by generating a call flow. The re-establishment process changes a train from a non-connected state to a connected state. The HO process allows a connected train to be handed over to another available RU. The train remains connected throughout the HO process.

3.3. *Interactions between two models*

The two models work together in the simulations. When a train starts either a HO or a re-establishment process, it informs the corresponding network elements that they will be needed or no longer be needed by the train. When a network element changes its state from (W) to (F), for instance, it will inform the train of the failure. If the train is already connected to the network, it will be disconnected and request a re-establishment process.

4. Simulation and results

We implement the proposed models in section 3 with the SimPy environment. We consider a railway line of 100 km with locally distributed RAN and one aggregated CN. The gNBs in RAN consist of co-located RUs and DUs at the edge data center and one aggregated CU at the gNB level data center. RUs are assumed to be purely physical equipment and are equally spaced alongside this 100 km line. First RU is at the starting point of the railway, and the last RU is at the endpoint. The RUs in this study can cover an area with a radius of 5 km using the spectrum it can provide. The failure process of the network system is given in Table 1, according to the data provided by the network service suppliers. The composition of our envisioned 5G and Beyond network is given in Table 2. Throughout the simulation, one train runs every hour from the start to the end of the line at a fixed speed of 200 km/h. All network links in this study are assumed ultra-reliable.

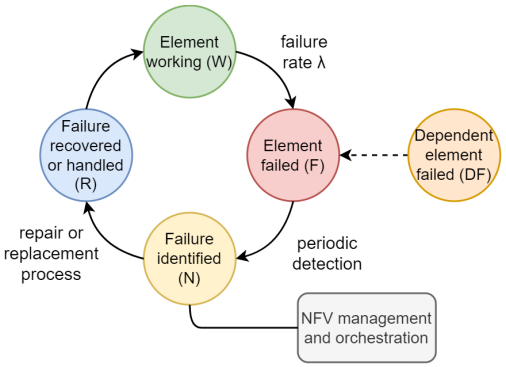


Fig. 3. 5G and Beyond network element model.

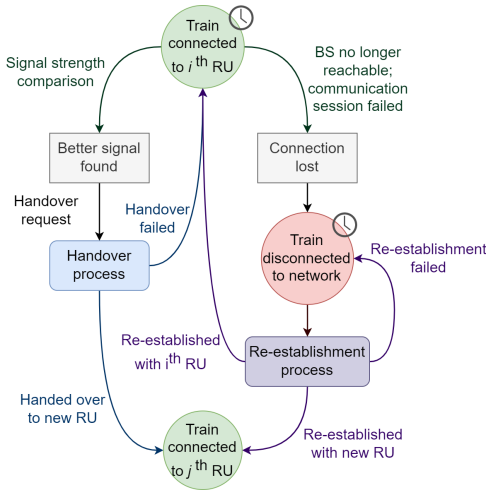


Fig. 4. High-speed train model.

Table 1. Failure processes of network system.

Item	MTTF	repair time
RU	50 years constant failure rate	1 hour fixed repair time
Virtual application (container)	52 days constant failure rate	10 s $U(0, 10)$ continuous uniform distributions
Server	1 year constant failure rate	1 hour fixed repair time

Table 2. Components of network system.

Items	Instances	Description
RU	Variable	Physical equipment
DU	1 for 1 RU	1 app and 1 server
CU	1 pair for 8 DUs	2 apps and redundant servers
UPF	1 in total	2 apps and redundant servers
AMF	1 in total	1 app and redundant servers

4.1. *Unreliable Radio Unit*

In the first scenario, we simplified the network elements to better explain the different perspectives from the network and the train. We consider that only RUs will fail in the network, and the rest of the system is highly reliable. We investigate how the density of radio installations may impact the network and service communication reliability.

From the network operator's perspective, the network availability and reliability are strictly defined by considering the capability to provide end-to-end connection and signaling message response at every position (including both single RU zones and overlapping zones) in the considered area. From the train's perspective, the system we consider is changing between a single RU system and an overlapping system dynamically as it travels.

We simulate the trains traveling through the railway for 100 000 hours (about 11 years) and estimate the availability and the MTTF of train network communication service. Via Monte-Carlo simulation, we compared the impact of different numbers of RUs, varying from 12 to more than 20. Figure. 5 and 6 show the availability and reliability metric MTTF for network and service. A direct computation of the series system helps us validate this result.

Obviously, neither availability nor reliability from these two perspectives is the same. For operators, when the number of RUs is below 20, some parts of the railway are always covered by a single RU. The more RU installation is dense, the larger the number of these single RU zones. The network availability and MTTF thus decrease with the number of RUs. However, if when the number of RUs is more than 20, there is a sudden jump. In fact, the RU setup is considered fully redundant everywhere, covered by at least two RUs (this redundant layout, in reality, is often not affordable for a network operator). The network service availability obtains nine nines (99.9999999%), and the MTTF is largely improved.

For train service, it only considers the RUs it can connect to at its position. A failed RU far from where the train is would not impact end-to-end service delivery for the train. At the overlapping zone, the re-establishment procedure helps the train to resume the connection if one of the

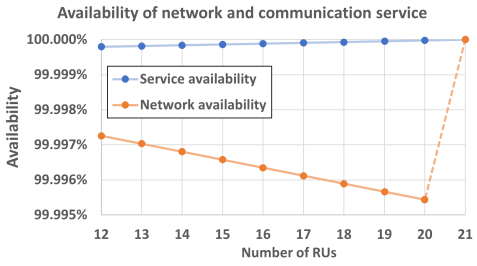


Fig. 5. Number of RUs' impact on network and service availability.

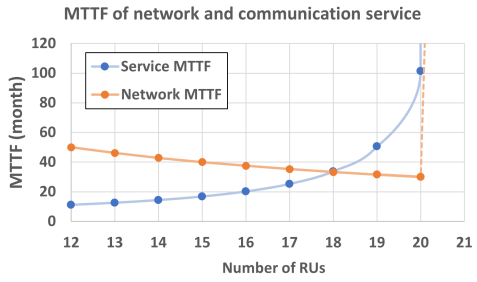


Fig. 6. Number of RUs' impact on network and service MTTF.

RU in the overlapping zone fails. Therefore, the more RUs installed, the less time it spends in a single RU area, and the more service can be guaranteed by at least two RUs in the overlapping area. Then the train communication service availability increases with the density of RU installation. With more than 20 RUs in the railway, the communication service availability reaches even 11 nines. However, the Radio Unit is expensive, and it is hard to do maintenance as they are often distributed. With a limited budget, one of the possible solutions could be deploying RUs according to geographical information of the train route and upgrading the existing 3G/4G facility.

4.2. *Random failures*

In the second scenario, we remove the assumption of high reliability on the rest of the network. All elements in gNBs and the CN can fail. Then the system becomes more complex. Still, we compare different Radio Unit densities alongside the railway. The simulation time is 100 000 hours to generate enough failure in the system.

For the network operators, the system is considered available when all network elements work as initially expected to provide end-to-end service, re-establishment request, and HO request anywhere in the considered railway network. The time to fail is the time from when at least one network element fails to when all the failed network elements are repaired.

For the high-speed train, the service is considered available when its connection is established, and all the UP functions it uses work. HO procedure provides seamless connection as it induces no service interruption and thus enhances service reliability. On the other hand, the re-establishment procedure helps an end-user reconnect to the network from either UP or HO failure. Re-establishment can not maintain a connection and always comes with a service interruption. Therefore, unlike HO, the re-establishment procedure can only enhance service availability but does not contribute to service reliability.

The estimated reliability and availability for the network and service from the simulation are shown in Table 3. Similar to the previous scenario, while we increase the number of RUs, the network availability and reliability decrease. However, for communication service, there are more failures during a train's mission, especially minor failures when the number of RUs increases. The re-establishment procedure can guarantee availability since the overlapping area gets larger. Nevertheless, as the number of failures still increases, the MTTF gets shorter, resulting in less reliable communication service. A possible solution for enhancing reliability could be adding redundant items, which may be energy-consuming and expensive for train and network operators.

Table 3. Performance with random failures

Number of RUs	Network availability	Network MTTF (hours)	Service availability	Service MTTF (hours)
12	99.86058%	55	99.99456%	359
13	99.84895%	52	99.99512%	344
14	99.83789%	50	99.99571%	333
15	99.82612%	48	99.99628%	319
16	99.81485%	46	99.99686%	308
17	99.80219%	44	99.99742%	298
18	99.79151%	42	99.99801%	288
19	99.78031%	41	99.99859%	279
20	99.76875%	39	99.99917%	270

5. Conclusion

This paper discussed the reliability of 5G and Beyond network applications on high-speed trains from two different angles. Service operators often focus on the overall system availability and reliability to provide end-to-end connection and signaling requests for the end-users everywhere in the network. In comparison, a high-mobility end-user focuses only on local issues. That is why high-speed train service has a different estimation of reliability and availability than the telecommunication network itself.

We also modeled both the 5G and Beyond network and the high-speed train to simulate how high-speed train interacts with the network by re-establishment and HO procedures. The discrete event simulation helps us understand the differ-

ent perspectives of network operators and service users on reliability and availability. The result also shows how they change with the density of the Radio Unit facility alongside the railway.

Our assumptions on the radio interface are ideal. Many aspects, such as weather conditions and moving speed, can cause other types of failures during the re-establishment and HO procedures. The failure rates of the system are assumed to be constant. When considering aging systems, degradation models should be applied. However, our current work has already provided valuable information on the reliability challenges of 5G and Beyond networks for high-speed train services.

The continuation of this work will focus on building an analytical model of the complex network system to validate our proposed approach and compare the performance with the discrete-event simulation. Further cooperation with railway companies will help refine the model by including additional information, such as railway geographical coordinates and train schedules, which will add more value to the approach.

References

3GPP (2021, Jan). TS 38.300 V16.4.0 5G; NR; NR and NG-RAN Overall description; Stage-2.

3GPP (2022, May). TS 22.289 V17.0.0 LTE; 5G; Mobile communication system for railways.

Al-Falahy, N. and O. Y. Alani (2017). Technologies for 5g networks: Challenges and opportunities. *IT Professional 19*(1), 12–20.

Bonati, L., M. Polese, S. D'Oro, S. Basagni, and T. Melodia (2020). Open, programmable, and virtualized 5g networks: State-of-the-art and the road ahead. *Computer Networks 182*, 107516.

El Banna, R., H. M. EL Attar, and M. Aboul-Dahab (2020). Handover scheme for 5g communications on high speed trains. In *2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC)*, pp. 143–149.

Fan, P., J. Zhao, and C.-L. I (2016). 5g high mobility wireless communications: Challenges and solutions. *China Communications 13*(2), 1–13.

Farooq, H., M. S. Parwez, and A. Imran (2015). Continuous time markov chain based reliability analysis for future cellular networks. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6.

He, R., B. Ai, G. Wang, K. Guan, Z. Zhong, A. F. Molisch, C. Briso-Rodriguez, and C. P. Oestges (2016). High-speed railway communications: From gsm-r to lte-r. *IEEE Vehicular Technology Magazine 11*(3), 49–58.

Masur, K. D. and D. Mandoc (2009, November). Lte/sae – the future railway mobile radio system: Long-term vision on railway mobile radio technologies. UIC Technical Report.

Niu, Y., Y. Li, D. Jin, L. Su, and A. V. Vasilakos (2015, Nov). A survey of millimeter wave communications (mmwave) for 5g: opportunities and challenges. *Wireless Networks 21*(8), 2657–2676.

Qu, L., M. Khabbaz, and C. Assi (2018). Reliability-aware service chaining in carrier-grade softwarized networks. *IEEE Journal on Selected Areas in Communications 36*(3), 558–573.

Song, H., X. Fang, and L. Yan (2014). Handover scheme for 5g c/u plane split heterogeneous network in high-speed railway. *IEEE Transactions on Vehicular Technology 63*(9), 4633–4646.

Sönmez, Ş., I. Shayea, S. A. Khan, and A. Alhammadi (2020). Handover management for next-generation wireless networks: A brief overview. *2020 IEEE Microwave Theory and Techniques in Wireless Communications (MTTW) 1*, 35–40.

Tanveer, J., A. Haider, R. Ali, and A. Kim (2022). An overview of reinforcement learning algorithms for handover management in 5g ultra-dense small cell networks. *Applied Sciences 12*(1).

Thiruvasagam, P. K., V. J. Kotagi, and C. S. R. Murthy (2022). A reliability-aware, delay guaranteed, and resource efficient placement of service function chains in softwarized 5g networks. *IEEE Transactions on Cloud Computing 10*(3), 1515–1531.

UIC (2020, December). Frmcs and 5g for rail: challenges, achievements and opportunities. Publication of UIC rail system department.

High-mobility 5G communication service: availability and reliability analysis

Rui Li*†, Bertrand Decocq*, Anne Barros†, Yiping Fang†,Zhiguo Zeng†
*Orange Innovation
Châtillon, France
{rui.li, bertrand.decocq}@orange.com
†Laboratoire Génie Industriel
CentraleSupélec, Université Paris-Saclay
Gif-sur-Yvette, France
{rui.li, anne.barros, yiping.fang, zhiguo.zeng}@centralesupelec.fr

***Abstract*—5G, the latest generation of cellular technology, is designed to support the various use cases of multiple industries. Railway transport is one of the most challenging usage scenarios the 5G system encounters. The telecommunication network service provided by 5G is crucial to guarantee the quality and safety of train traffic. Therefore, estimating the availability and reliability of such network service is necessary. This article separates spatially and temporally the 5G network into subsystems. This article also provides methods and calculation expressions for evaluating the availability and reliability of subsystems as well as the overall network service.**

***Index Terms*—5G, network service, service availability, reliability, railway.**

I. Introduction

With the fast worldwide development of railway systems, especially high-speed railways, the demand for mobile services has dramatically grown during the last decade. It is necessary to provide stable and reliable railway services to the massive passenger flow. The current railway-oriented telecommunication system, GSM-R, faces a decommission issue and can no longer satisfy the demanding requirements of high-speed transport service [1]. A new telecommunication network based on 5G is designed to be the successor of the GSM-R [2].

The unavailability and unreliability of network service are the main factors that seriously impact train service. Once the connection is lost, the train cannot be tracked, and signal transmission with the control center and information exchange will be interrupted. As communication networks become more complex, finding a practical way to estimate and improve the network service availability and reliability for railway communication service usage before replacing the current telecommunication system with 5G is essential.

This paper introduces a method to spatially and temporally regroup the 5G system into subsystems for high-mobility communication services. The network service availability and reliability are evaluated by combining series-parallel system availability and reliability analysis of the subsystems.

The structure of this paper is organized as follows. The telecommunication network for high-mobility users is introduced in Section II. Section III presents the regrouping of the network system and the method to assess network availability and reliability. Section IV focuses on series-parallel subsystem availability and reliability analysis. Numerical evaluation is showcased in Section V. Finally, Section VI concludes the paper with some remarks.

II. 5G network for railway communication

In a 5G network, an End-to-End (E2E) service connection is established through a Radio Access Network (RAN), which connects devices to other parts of a network via radio connections, and a Core Network (CN), which manages the connection to the service platform or the internet. By introducing Network Function Virtualization (NFV) and Software Defined Networking (SDN) technologies [3] into 5G, the network is virtualized and can be flexibly deployed and easily managed. After virtualization, RAN and CN are virtualized, for example, as microservices in containers hosted on physical infrastructure.

When a 5G network is applied to high-mobility scenarios, some components may need multiple instances and be distributed along the railway track due to the radio coverage distance constraints and service latency requirements. A user will only connect to the RAN components covering it, as shown in Fig. 1. Therefore, at a given position, the train only establishes an E2E connection via the reachable local RAN components. The local RAN is connected to an aggregated CN. The CN components are often located in a data center far from the RANs. Sometimes, a train is reachable by multiple RAN components, like in Zone 2 in Fig. 1. This zone is also called an overlapping area, covered by multiple radio antennas. Sometimes, the train runs in the zone covered by only one RAN radio antenna, as Zone 1 in Fig. 1.

The 5G network is supposed to provide various high-availability and high-reliability railway communication services, including voice and data communication. These services are based on user plane E2E communication. This E2E communication requires all user plane network components to operate correctly. Inspired from [4], under the context of 5G for high-mobility trains, the network service availability can be defined as the probability that the E2E connection is available at any instant. Reliability is often used to characterize if a

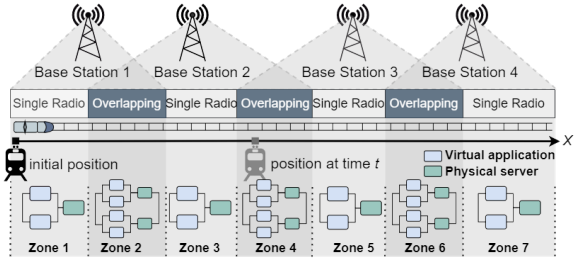


Fig. 1. Example of 5G network along a railway track.

system is appropriately working during a specific period of time [5]. The 5G network we consider is a repairable system. We use Mean Time To Failure (MTTF), the average time the E2E connection lasts, and Mean Time To Repair (MTTR), the average time to repair the E2E connection, to estimate the network service reliability. When the time moment in the availability definition t tends to infinity, the steady-state availability equals MTTF/(MTTF + MTTR) [6].

III. System decomposition

Since not all network components are usable for the train at a given position and time, it is possible to simplify the 5G system by considering different subsystems when assessing the network service availability and reliability.

We set the length of the considered railway as S. Alongside this rail line, N Base Stations $BS = \{bs_1, bs_2, ..., bs_N\}$ are evenly distributed from the start $x = 0$ to the end $x = S$ of the line. Each base station bs_n can effectively transmit radio signals to end users in a zone with a radius r_n. We divide this rail line into M zones $Z = \{z_1, z_2, ...z_M\}$ such that in zones z_i, z_j, $\forall i, j \in \{1, 2, ..., M\}$, for the corresponding effective covering Base Station ensembles $c_i, c_j \subseteq BS$, we have $c_i \neq c_j$. If in zone z_i, $card(c_i) = 1$, it is called a single covering zone. If $card(c_i) \geq 2$, then it is called overlapping zone.

Fig. 1 shows how zones reconstitute a telecommunication network. The whole system comprises virtual and physical components for RAN and CN. At the moment t, the train is in the middle of the train line of zone z_4 and is reachable to the second and third Radio Base Stations. The covering Base Stations are $c_4 = \{bs_2, bs_3\}$. Each Base Station can establish an E2E connection by a series-parallel network function system, which is composed of two virtual applications and one physical server (a simplified demonstrative example). These two series-parallel systems are also in parallel and form a subsystem for zone z_4. When the train enters zone z_5, the only effective covering Base Station is $c_5 = \{bs_3\}$. The subsystem for zone z_5 consists of two virtual components and one physical component. With M zones, the entire system can be regrouped into M subsystems.

The availability of a train network service is the average percentage of available time that the train can connect to the Data Network via at least one subsystem. The reliability of a train network service is the capacity to provide E2E connection without failure, which is characterized by the Mean Time Between Failures (MTBF) of the service in the present study.

The availability and reliability of one subsystem provide the availability and reliability for the communication service of a train running at this specific zone. Although some components could belong to multiple subsystems by this regrouping, the failures of these subsystems are assumed to be independent. For the train use cases, these subsystems are temporally and spatially independent. At a given moment t, the train is located only at one position and connects to only one subsystem. The train service's available time can be computed as the sum (superposition) of the available time of those subsystems it passes.

$$A_{train} = \frac{\sum_{i=0}^{M} A_{subnet_i} \cdot T_i}{T_{total}} \times 100\% \tag{1}$$

Equation 1 calculates the service availability. A_{subnet_i} is the availability of the i-th subsystem. T_i is the train passing time at zone z_i. The train network service availability shows the percentage of time the train can use the E2E communication during the trip.

The number of failures of the train service for a given duration that the train stays in the subsystem can be deduced from the reliability of the subsystem. Then, by assuming these subsystems are independent, it is possible to extract the MTBF for the overall train service.

$$MTBF_{train} = \frac{T_{total}}{\sum_{i=0}^{M} \frac{T_i}{MTTF_i + MTTR_i}} \tag{2}$$

In Equation 2, the sum of $MTTF_i$ and $MTTR_i$ is the MTBF of the i-th subsystem. T_i is the train passing time at zone z_i. The passing time divided by MTBF at zone z_i is the number of failure occurrences when the train passes zone z_i. The train network service reliability indeed describes how often an E2E service interruption may happen during the trip.

IV. Subsystem availability and reliability

A first assumption to simplify the considered subsystems model is that all components, whatever their nature, physical or virtual, their failure processes follow the exponential law, and so do their repair processes. Based on this assumption, we create a state space model of the subsystem. The m-th subsystem is an ensemble of m_k components, $S_m = \{e_1, e_2, ..., e_{m_k}\}$. There will be 2^{m_k} states in total, as each element can be either working or failed.

An example of a subsystem with three elements (two identical virtual component instances and one server) is the subsystem in Zone 1. The two virtual functions are in parallel to provide redundancy. If one virtual component fails, the other keeps the subsystem's virtual part alive. The server and the virtual functions are in series. The whole subsystem fails if the only server or the parallel virtual part fails.

In this subsystem, each component is either in the state "Working" or "Failed". TABLE I gives the entire eight subsystem states. The reliability of a repairable series-parallel system can not directly be solved using tools like the Reliability Bloc

Diagram. The state space models are preferred. We build a Markov chain [7] with this list of possible states. The possible transition paths are shown in the figure of Continuous-Time Markov Chains (CTMC) in Fig. 2. λ and μ represent the element failure and repair rate respectively. With the help of the transition rate matrix of the CTMC, the stationary distribution of the CTMC, π can be obtained. We can then deduce the subsystem's availability.

TABLE I
STATES OF THE SUBSYSTEM CONTAINING TWO VIRTUAL COMPONENTS AND ONE SERVER

Chain State	**Component state**			**System state**
	Virtual 1	***Virtual 2***	***Server***	
1 (1,1,1)	Working	Working	Working	Working
2 (0,1,1)	Failed	Working	Working	Working
3 (1,0,1)	Working	Failed	Working	Working
4 (1,1,0)	Working	Working	Failed	Failed
5 (0,0,1)	Failed	Failed	Working	Failed
6 (1,0,0)	Working	Failed	Failed	Failed
7 (0,1,0)	Failed	Working	Failed	Failed
8 (0,0,0)	Failed	Failed	Failed	Failed

The set of chain states $CS \in \{1,2,3,4,5,6,7,8\}$ corresponds to the combination of component states in the subsystem. Simulation results show that the subsystem stays short at a transient state and moves fast to a steady state. A detailed example will be given in Section V-A. Supposing $p(i,t), i \in \{1,2,3,4,5,6,7,8\}$ represents the probability of the subsystem being at state i at time t. In steady-state SS, the probability of the subsystem at states $\{1,2,3\}$ is $p(SS =$ "Working"$) = \lim_{t\to+\infty} p(\{1,2,3\},t)$. This distribution gives us a rapid answer to compute subsystem availability, which is equivalent to the sum of the first three items of π.

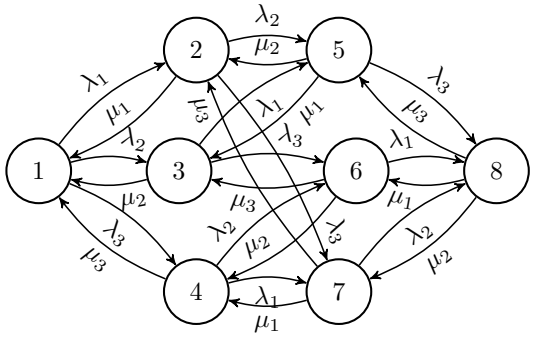


Fig. 2. Subsystem represented by a Continuous-Time Markov Chain.

The stationary distribution of the subsystem states $\pi = \{p_{1\infty}, p_{2\infty}, ..., p_{8\infty}\}$ can be directly computed from the transition matrix of the CTMC. The availability of the series-parallel subsystem by adding the stationary distribution of all "Working" states is:

$$A_{subnet} = \sum_{i=1,2,3} p_{i\infty} \tag{3}$$

However, it could be more complicated when computing the subsystem's reliability. Instead of looking at all changes of states, we consider two Discrete-Time Markov processes: the failure process and the repair process.

For the failure process, we consider the transitions inside "Working" states and from "Working" states to "Failure" states. This process starts from the subsystem's recovery and ends with the state changed to $CS \in \{4,5,6,7\}$, represented by recurrent states, as shown in Fig. 3. The transition probability of this Discrete-Time Markov Chain is deduced from the CTMC. It shows how the subsystem definitively changes from one state to another. Each transition corresponds to a state-changing event. For example, the state-changing probability from state 1 to state 2 in Fig. 3 is the probability of moving to state 2 after the first state-changing event from state 1. We use variables τ_1, τ_2, τ_3 to represent the failure time of component 1, 2 and 3. The state-changing probability from state 1 to state 2 is Prob$\{\min(\tau_1,\tau_2,\tau_3) = \tau_1\} = \frac{\lambda 1}{\lambda 1+\lambda 2+\lambda 3}$.

We describe this Markov Chain of failure process by a stochastic transition matrix P_F. The initial state is the state of the very moment that the subsystem is repaired to "Working" state. The chain has a final state as the failure process always ends with the connection becoming unavailable. That is the state of the very moment that the subsystem for the first time goes into "Failure" state. Since state 8 is not a direct "Failure" state and is only reachable from another "Failure" state $CS \in \{4,5,6,7\}$, state 8 is not enaged during this process. As a result, we define the process initial state distribution π_F^0 and final state distribution π_F^∞. We get the following relations:

$$\pi_F^0 = [p_{F1}^0, p_{F2}^0, p_{F3}^0, p_{F4}^0, p_{F5}^0, p_{F6}^0, p_{F7}^0, p_{F8}^0] \tag{4}$$

$$\pi_F^\infty = [p_{F1}^\infty, p_{F2}^\infty, p_{F3}^\infty, p_{F4}^0, p_{F5}^\infty, p_{F6}^\infty, p_{F7}^\infty, p_{F8}^\infty] \tag{5}$$

$$\lim_{k\to+\infty} \pi_F^0 \times {P_F}^k = \pi_F^\infty \tag{6}$$

where:
$\sum_{i=1}^{8} p_{Fi}^0 = 1$, and $p_{Fi}^0 = 0$ for $i \in \{4,5,6,7,8\}$
$\sum_{i=1}^{8} p_{Fi}^\infty = 1$, and $p_{Fi}^\infty = 0$ for $i \in \{1,2,3,8\}$

For the repair process, we consider the opposite. All start from "Failure" states $CS \in \{4,5,6,7\}$. This process ends by reaching the states $CS \in \{1,2,3\}$, represented by recurrent states, as shown in Fig. 4. The transition probability is also deduced from the CTMC of the subsystem.

We describe this Markov Chain by a transition matrix P_R. The initial state is the state of the very moment that the subsystem failed to "Failure" state. The final state is the state of the very moment that the subsystem, for the first time, goes into "Working" states. Since state 8 is only reachable from another "Failure" state $CS \in \{4,5,6,7\}$, the initial "Failure" state can not be 8. As a result, we define the process initial state π_R^0 and final state π_R^∞.

We get the following relations:

$$\pi_R^0 = [p_{R1}^0, p_{R2}^0, p_{R3}^0, p_{R4}^0, p_{R5}^0, p_{R6}^0, p_{R7}^0, p_{R8}^0] \tag{7}$$

$$\pi_R^\infty = [p_{R1}^\infty, p_{R2}^\infty, p_{R3}^\infty, p_{R4}^0, p_{R5}^\infty, p_{R6}^\infty, p_{R7}^\infty, p_{R8}^\infty] \tag{8}$$

$$\lim_{k\to+\infty} \pi_R^0 \times P_R^k = \pi_R^\infty \tag{9}$$

where:
$\sum_{i=1}^{8} p_{Ri}^{0} = 1$, and $p_{Ri}^{0} = 0$ for $i \in \{1, 2, 3, 8\}$
$\sum_{i=1}^{8} p_{Ri}^{\infty} = 1$, and $p_{Ri}^{\infty} = 0$ for $i \in \{4, 5, 6, 7, 8\}$
When the subsystem is in steady state, we have:

$$\pi_F^{\infty} = \pi_R^0 \tag{10}$$

$$\pi_R^{\infty} = \pi_F^0 \tag{11}$$

By solving Equations 4 - 11, we obtain the initial and state distribution of "Working" and "Failure" states.

Each transition step corresponds to a sojourn time in the Discrete-Time Markov Chain in the CTMC. We define the state sojourn time T_i^s as the mean time between the subsystem entering state i and leaving the state i in the CTMC. We also define the mean state failure time T_i^F as the mean time between the subsystem entering "Working" state i and the first time entering a "Failure" state. It is the sum of a set of transition steps in the Discrete-Time Markov Chain for the failure process. The MTTF we intend to compute is the mean state failure time of all "Working" states.

$$MTTF = \sum_{i \in \{1,2,3\}} \frac{p_{Fi}^0}{\sum_{j \in \{1,2,3\}} p_{Fj}^0} \cdot T_i^F \tag{12}$$

Note that for this case, $\sum_{j \in \{1,2,3\}} p_{Fj}^0 = 1$.

The Markov Chain of the failure process in Fig. 3 gives the following relations:

$$T_1^F = T_1^s + \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3} \cdot T_2^F + \frac{\lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \cdot T_3^F \tag{13}$$

$$T_2^F = T_2^s + \frac{\mu_1}{\mu_1 + \lambda_2 + \lambda_3} \cdot T_1^F \tag{14}$$

$$T_3^F = T_3^s + \frac{\mu_2}{\lambda_1 + \mu_2 + \lambda_3} \cdot T_1^F \tag{15}$$

For Equation 13, the average failure time T_1^F includes the time spent in different steps. In the first step, the subsystem leaves state 1 and spends time T_1^s, the CTMC sojourn time in state 1. According to the state transition probabilities, from state 1, the subsystem may change to state 2, 3, or 4. In the next step, the process ends if the subsystem directly fails to state 4. Otherwise, it will spend time T_2^F and T_3^F accordingly for the rest of the failure process. The average failure time of state 2 and 3 can be represented similarly in Equations 14, 15. Finally, the MTTF is obtained by solving Equations 12 - 15.

The MTTR is the total transition time of a subsystem being repaired during failure. We define the mean state repair time T_i^R as the average time between the subsystem entering a specific "Failure" state i and the first time entering a "Working" state. Therefore, the MTTR is the mean sojourn time at all "Failure" states.

$$MTTR = \sum_{i \in \{4,5,6,7,8\}} \frac{p_{Ri}^0}{\sum_{j \in \{4,5,6,7,8\}} p_{Rj}^0} \cdot T_i^R \tag{16}$$

Note that for this case, $\sum_{j \in \{4,5,6,7,8\}} p_{Rj}^0 = 1$ and $p_{R8}^0 = 0$.

The Markov Chain of the repair process in Fig. 4 gives the following relations:

$$T_4^R = T_4^s + \frac{\lambda_1}{\lambda_1 + \lambda_2 + \mu_3} \cdot T_7^R + \frac{\lambda_2}{\lambda_1 + \lambda_2 + \mu_3} \cdot T_6^R \tag{17}$$

$$T_5^R = T_5^s + \frac{\lambda_3}{\mu_1 + \mu_2 + \lambda_3} \cdot T_8^R \tag{18}$$

$$T_6^R = T_6^s + \frac{\mu_2}{\lambda_1 + \mu_2 + \mu_3} \cdot T_4^R + \frac{\lambda_1}{\lambda_1 + \mu_2 + \mu_3} \cdot T_8^R \tag{19}$$

$$T_7^R = T_7^s + \frac{\mu_1}{\mu_1 + \lambda_2 + \mu_3} \cdot T_4^R + \frac{\lambda_2}{\mu_1 + \lambda_2 + \mu_3} \cdot T_8^R \tag{20}$$

$$T_8^R = T_8^s + \frac{\mu_1}{\mu_1 + \mu_2 + \mu_3} \cdot T_6^R + \frac{\mu_2}{\mu_1 + \mu_2 + \mu_3} \cdot T_7^R + \frac{\mu_3}{\mu_1 + \mu_2 + \mu_3} \cdot T_5^R \tag{21}$$

For Equation 17, the average repair time T_4^R includes the time spent in different steps. In the first step, the subsystem leaves state 4 and spends time T_4^s, the CTMC sojourn time in state 4. According to the state transition probabilities, from state 4, the subsystem may change to state 1, 6, or 7. In the next step, the process ends if the subsystem is directly repaired to state 1. Otherwise, it will spend time T_6^F and T_7^F accordingly for the rest of the failure process. The average failure time of states 5, 6, 7, and 8 can be represented similarly. Finally, the MTTR is obtained by solving Equations 16 - 21.

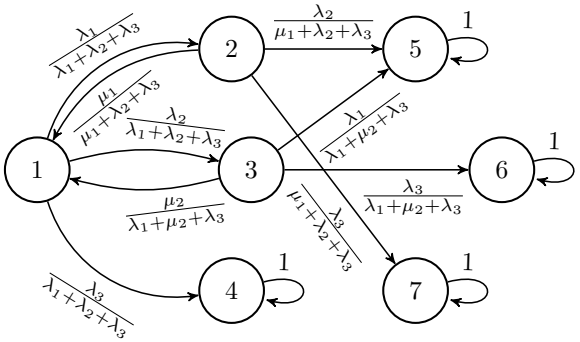


Fig. 3. Failure process represented by a Discrete-Time Markov Chain.

Although only a three-element system is demonstrated, the proposed method can also be applied to any series-parallel system. However, the state space will increase exponentially with the number of considered components.

V. Numerical evaluation

A. Example of a three-element subsystem

Now we consider a system with two virtual components, #1 and #2, and one physical component, #3. The virtual components are the applications that are often threatened by operational failures. The physical component often refers to a physical server where the applications are hosted, which is less likely to fail. Repairing a virtual component takes only

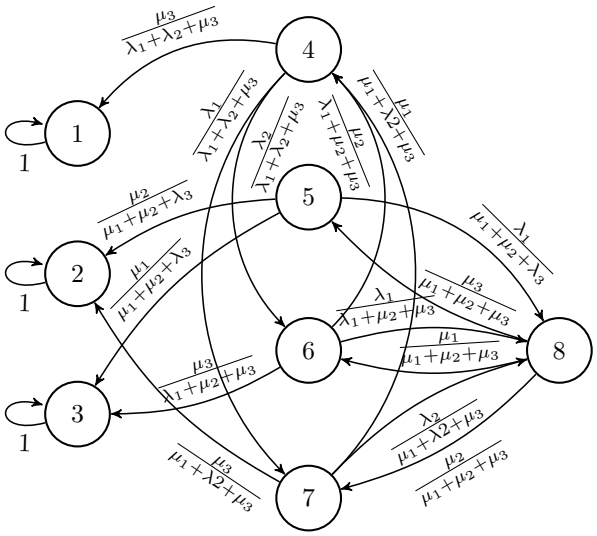


Fig. 4. Repair process represented by a Discrete-Time Markov Chain.

a few seconds by restarting the application. However, when a physical server fails, it must be repaired manually. TABLE II shows the failure and repair rates.

TABLE II
FAILURE AND REPAIR RATES OF COMPONENTS

Failure process			
Component	Symbol	Rate [hour^{-1}]	MTTF
1 - virtual	λ_1	0.005	200 hours
2 - virtual	λ_2	0.005	200 hours
3 - physical	λ_3	0.0002	5000 hours
Repair process			
Component	Symbol	Rate [hour^{-1}]	MTTR
1 - virtual	μ_1	360	10 seconds
2 - virtual	μ_2	360	10 seconds
3 - physical	μ_3	1	1 hour

After building the CTMC model and the transition rate matrix, we calculate the transient availability of such system as shown in Fig. 5. Initially, the brand new subsystem has 100% availability. After a few hours, it gradually drops to the stationary availability around 99.98%. The steady state of this CTMC also gives us a similar result as shown in TABLE III. The availability of the subsystem is $A_{subsystem} = p^1_\infty + p^2_\infty + p^3_\infty = 99.9800\%$. This shows that, at a stationary state, 99.9800% of the time, this subsystem is available to provide application service to the end user.

TABLE III
STATIONARY STATE DISTRIBUTION OF THE SUBSYSTEM

State	**Probability**	**State**	**Probability**
1	9.99772e−1	5	1.92857e−10
2	1.38857e−5	6	2.77715e−9
3	1.38857e−5	7	2.77715e−9
4	1.99954e−4	8	3.85715e−14

As for reliability, two Discrete-Time Markov Chains are built for failure and repair processes. The subsystem reparation processes are assumed to be parallel, i.e., each component can fail or be repaired independently. Equations 4 - 11 give the initial and final states of failure and repair processes as shown in Table IV.

TABLE IV
INITIAL AND FINAL STATE DISTRIBUTION

Failure process		**Repair process**	
p^0_{F1}	9.99278e−1	p^∞_{R1}	9.99278e−1
p^0_{F2}	3.60850e−4	p^∞_{R2}	3.60850e−4
p^0_{F3}	3.60850e−4	p^∞_{R3}	3.60850e−4
p^∞_{F4}	9.99278e−1	p^0_{R4}	9.99278e−1
p^∞_{F5}	6.93943e−4	p^0_{R5}	6.93943e−4
p^∞_{F6}	1.38789e−5	p^0_{R6}	1.38789e−5
p^∞_{F7}	1.38789e−5	p^0_{R7}	1.38789e−5

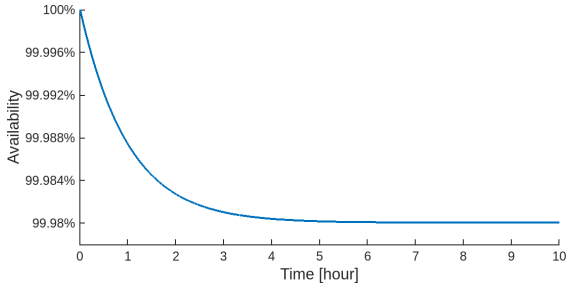


Fig. 5. Transient availability of the subsystem.

Using Equations 12 - 21, we obtain the MTTF and MTTR of the subsystem. MTTF of the subsystem is 4996.53 hours, and MTTR is 0.999307 hours. The physical server failure primarily dominates the subsystem failure time, and the repair time is also dominated by physical server repair because, unlike the virtual components, the physical component is not designed with redundancy in the subsystem. That shows that a possible way to improve the subsystem availability is to reduce physical component failure and repair time.

B. From subsystem to the whole system

The considered railway is 100 km long. A train runs at a constant speed, 200 km per hour. We assume the trains are well-timed and always pass the zone at a fixed time. The information of each zone is given in TABLE V.

It is imagined that along the railway line, all Radio Base Stations with their connected components in Fig. 1 have the same structure. Each of them forms a sub-network as the one in Section V-A. In Zone 1, 3, 5, and 7, the subsystem is the same as in Section V-A. While in Zone 2, 4, and 6, the subsystems are in the form of two sub-networks of Section V-A working in parallel. The reliability and availability of these subsystems are computed following the proposed method.

The average available network service time and average number of network service failures when a train passes each

TABLE V
SUBSYSTEM CHARACTERISTICS OF EACH ZONE

Zone	Expected passing time [min]	Availability	MTBF
1	3.6	99.980004%	4997.5 hours
2	6.6	99.999996%	1426.5 years
3	2.4	99.980004%	4997.5 hours
4	4.5	99.999996%	1426.5 years
5	3.3	99.980004%	4997.5 hours
6	6.0	99.999996%	1426.5 years
7	3.6	99.980004%	4997.5 hours

of the seven zones are given in TABLE VI. By summing up the result in the subsystems, the total mean network service available time of the 100 km route is 29.99742 minutes out of a 30-minute ride. The network service availability is 99.9914%. The total mean number of generated failures is $4.30441e{-5}$. In other words, there will be one failure about every 11616 running hours, which is one failure every 16 months if the train keeps running on this railroad section 24 hours per day.

TABLE VI
MEAN SERVICE AVAILABLE TIME AND MEAN SERVICE FAILURES

Zone	Available time [min]	Failures
1	3.5992801	$1.20059e{-5}$
2	6.5999997	$8.80260e{-9}$
3	2.3995201	$8.00395e{-6}$
4	4.4999998	$6.00177e{-9}$
5	3.2993401	$1.10054e{-5}$
6	5.9999998	$8.00236e{-9}$
7	3.5992801	$1.20059e{-5}$

According to the target requirements from [8], for some rail communication services, for example, the train coupling, the targeted communication service availability is 99.9999%. The network we considered is not yet satisfying, and some potential improvements can be made. The first improvement is adding parallel virtual components to each unitary subsystem connected to the Base Station. Instead of 2 virtual components, the unitary subsystem has been upgraded to 3. The second improvement could be adding a redundant parallel physical server to the unitary subsystem. The service availability and reliability comparison is showcased in Fig. 6. Adding parallel virtual components has less impact on availability and reliability since the virtual element is already redundant. Adding a redundant physical component can vastly improve both availability and reliability. On average, the train can connect to network service for over 10 thousand months without interruption. The availability improved to more than seven nines, largely above the requirement.

VI. CONCLUSION

This paper discussed the method of evaluating the availability and reliability of 5G network communication services applied to high-mobility users. The 5G for the railway is larger and more complex than ordinary 5G networks. We proposed decomposing the 5G communication network into spatially and temporally independent subsystems to simplify

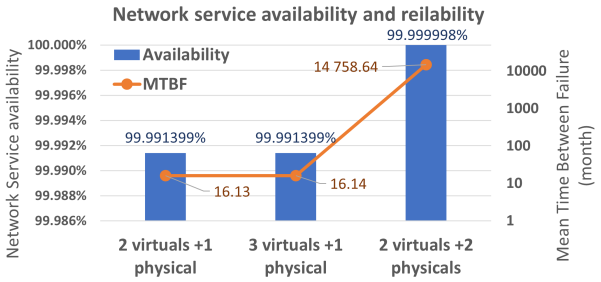


Fig. 6. Service availability and reliability comparison.

the availability and reliability assessment. We use a series-parallel model to estimate the availability and reliability of the subsystems. The overall railway communication service availability and reliability are obtained by combining the evaluation results from subsystems. A numerical example showed that this method could assess a railway communication service's availability and reliability using the network structure and components' properties.

It should be noted that the example we used in the paper is for demonstration. The actual 5G network contains lots of network functions. Even after the regrouping, its subsystems can be more complex than the three-element system. All system states and transitions should be carefully studied for the correct availability and reliability evaluation. Although numerical simulation can be more practical than building Markov Chains, the simulation will take an extremely long time to get accurate results when there are rare events. Besides, using our proposed method, it is easier to change several parameters to compare the network service performance once the Markov Chain is built.

For the next step, the collaboration with railway companies is expected in order to refine and validate the model by considering railway geographical coordinates and train schedules, adding more value to this work.

REFERENCES

[1] R. He et al., "High-Speed Railway Communications: From GSM-R to LTE-R," *IEEE Vehicular Technology Magazine*, vol. 11, no. 3, pp. 49–58, September 2016.

[2] International Union of Railways, "FRMCS and 5G for rail: challenges, achievements and opportunities," December 2020. [Online] https://uic.org/IMG/pdf/brochure_frmcs_v2_web.pdf.

[3] F. Z. Yousaf, M. Bredel, S. Schaller and F. Schneider, "NFV and SDN—Key Technology Enablers for 5G Networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2468–2478, November 2017.

[4] M. L. Shooman, *Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design*. New York, NY, USA: Wiley, 2002.

[5] A. Avizienis, J. -C. Laprie, B. Randell and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, January–March 2004.

[6] B. W. Johnson, *Design and Analysis of Fault Tolerant Digital Systems*. Reading, MA, USA: Addison-Wesley, 1989.

[7] W.J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton, NJ, USA: Princeton Univ. Press, 1994.

[8] 3GPP TS 22.289, "LTE; 5G; Mobile communication system for railways (Release 17)", v17.0.0, May 2022.