Estimating 5G network service resilience against short timescale traffic variation

Rui Li, Bertrand Decocq, Anne Barros, Yi-Ping Fang, Zhiguo Zeng

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

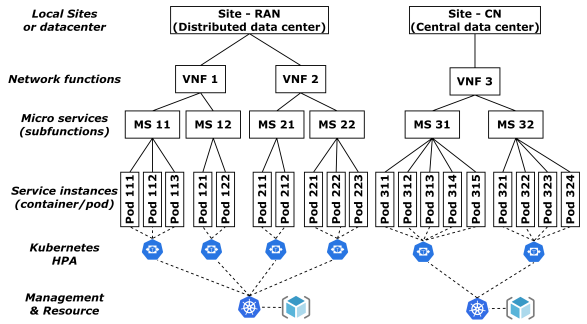


Fig. 1. 5G container-based NFV hierarchy topology example with one local site and one centralized data center site.

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

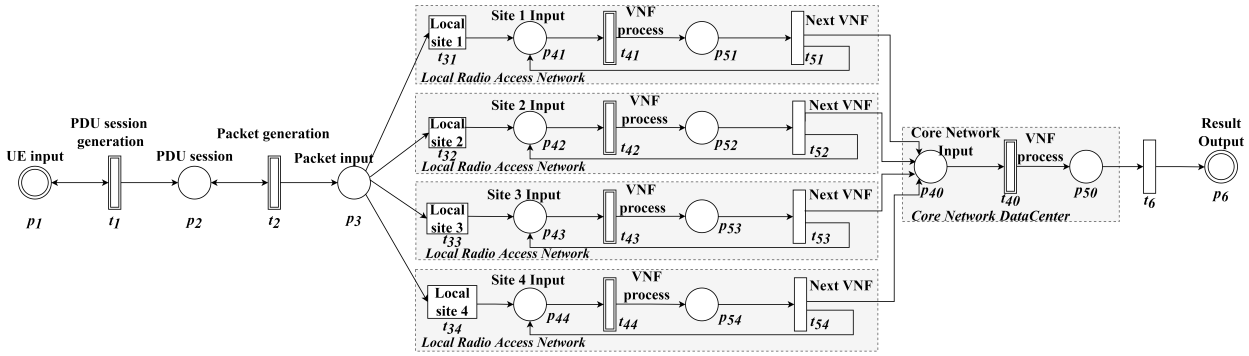


Fig. 2. Service delivery level Petri Net. Example with four radio cells and one core network data center.

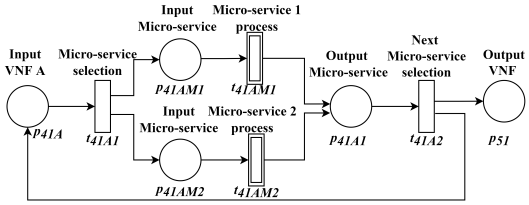


Fig. 3. VNF processing level Petri Net. Example of VNF A.

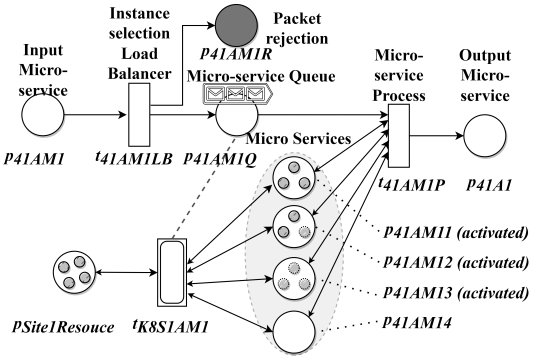


Fig. 4. Packet processing level Petri Net. Example of micro-service of the first VNF in t_{41}, VNF A.

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

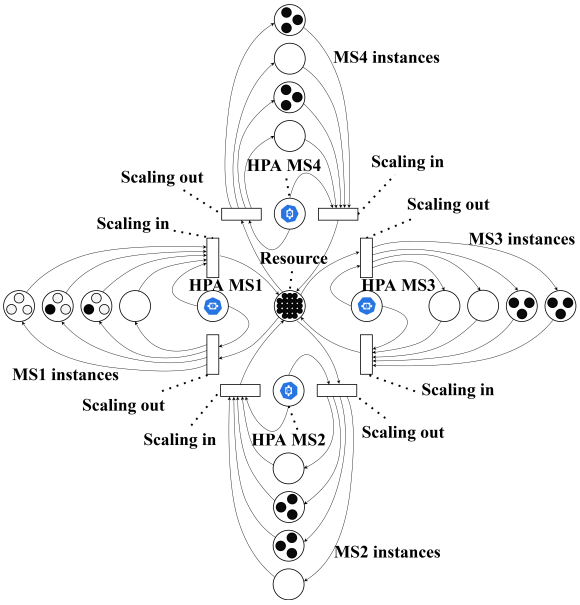


Fig. 5. Micro-service management level Petri Net. Example of a site with four micro-services.

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

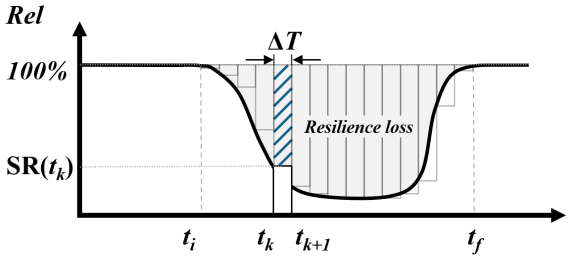


Fig. 6. The resilience triangle. The incident takes place at t_i. The system recovers at t_f. The gray part represents resilience loss of the k-th time slot.

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

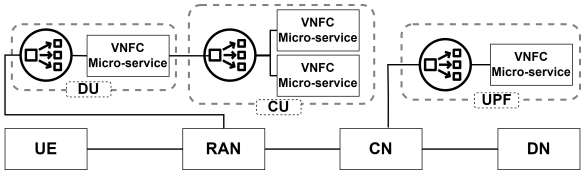


Fig. 7. Service function chain including 3 VNFs.

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
Maximal waiting time	1000 ms	reject if time out

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

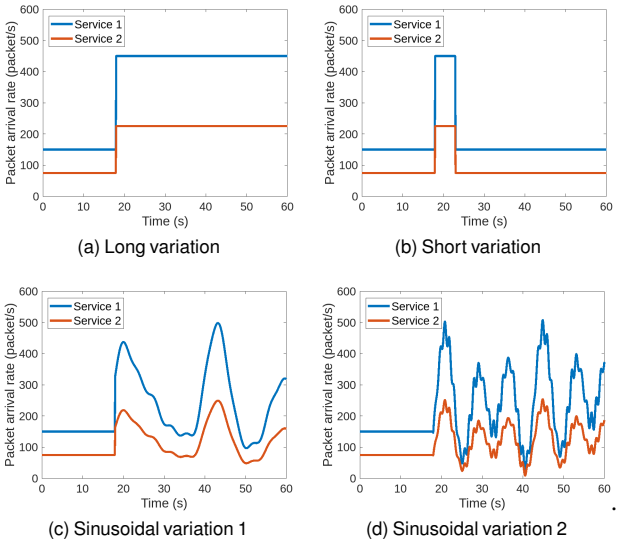


Fig. 8. Four traffic patterns with different arrival rate variations after t = 18s. (a) Long-term constant variation pattern, approximate entropy: 0.0108. (b) Short-term constant variation pattern, approximate entropy: 0.0207. (c) Sinusoidal (superposition) variation pattern 1, approximate entropy: 0.1019. (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:

$$\text{New scale} = \lceil \frac{\text{Current utilization}}{\text{Desired utilization}} \rceil \cdot \text{Current scale}. \quad (7)$$

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.

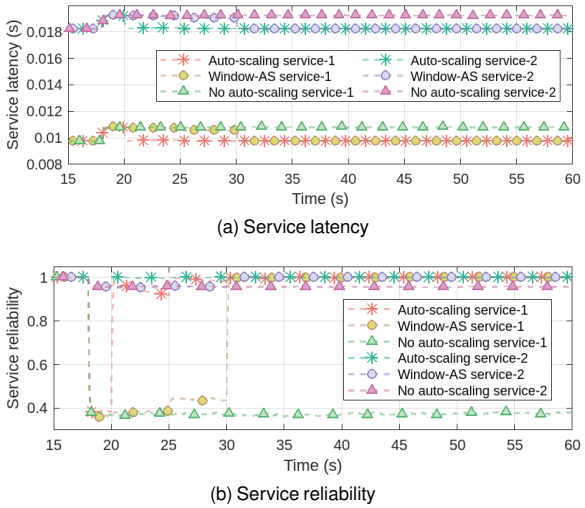


Fig. 9. Service latency and reliability under a long-term traffic variation (pattern a) with different management strategies (auto-scaling, stabilization window-based auto-scaling, and no auto-scaling).

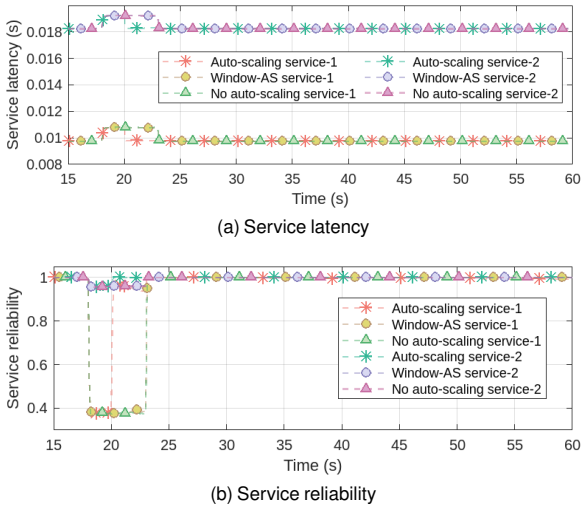


Fig. 10. Service latency and reliability under a short-term traffic variation (pattern b) with different management strategies (auto-scaling, stabilization window-based auto-scaling, and no auto-scaling).

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.

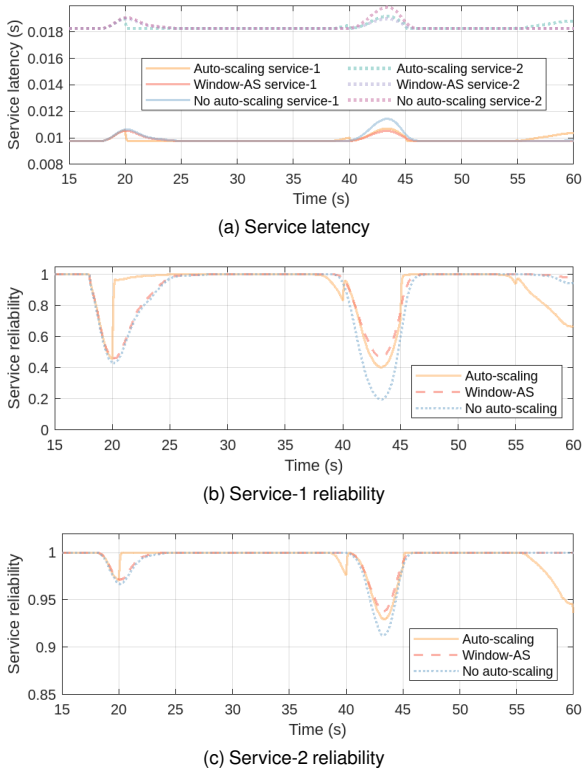


Fig. 11. Service latency and reliability under sinusoidal superposition traffic variation (pattern c) with different management strategies (auto-scaling, stabilization window-based auto-scaling, and no auto-scaling).

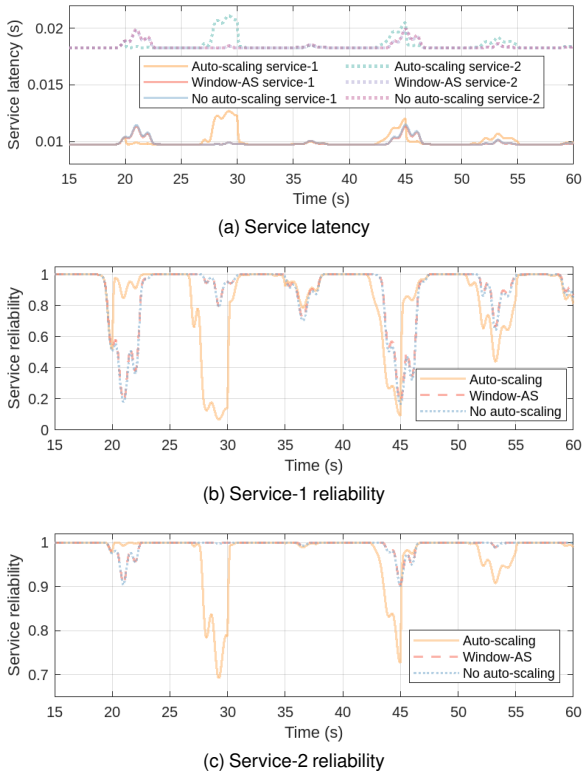


Fig. 12. Service latency and reliability under sinusoidal superposition traffic pattern variation (pattern d) with different management strategies (auto-scaling, stabilization window-based auto-scaling, and no auto-scaling).

B. Resilience with network service isolation

Without isolation, the network resources are shared by all network services. By introducing network slicing, network 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

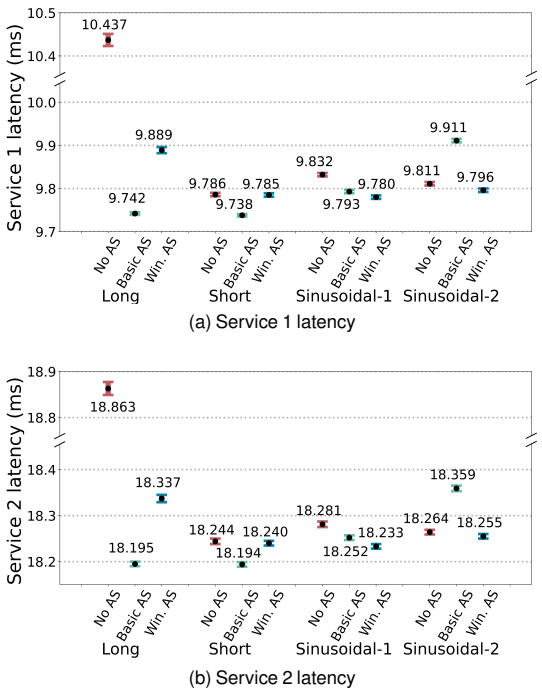


Fig. 13. Service 1 (a) and Service 2 (b) latency values and confidential intervals in case 1.

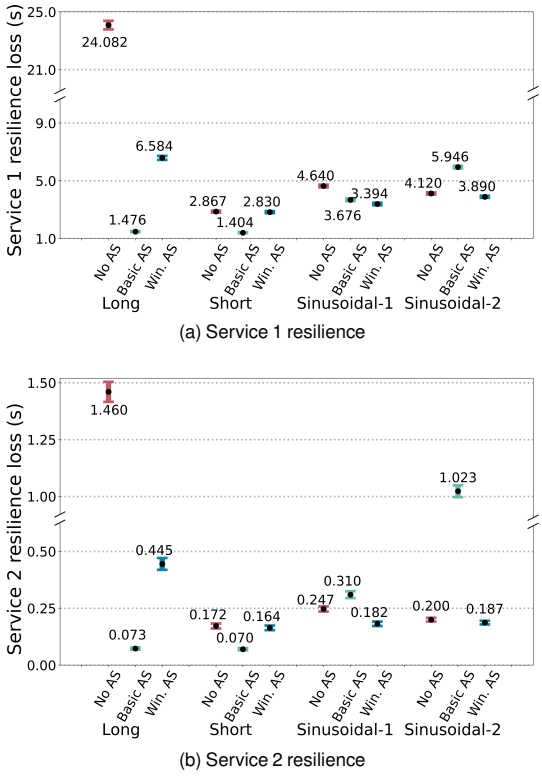


Fig. 14. Service 1 (a) and Service 2 (b) resilience loss values and confidential intervals in case 1.

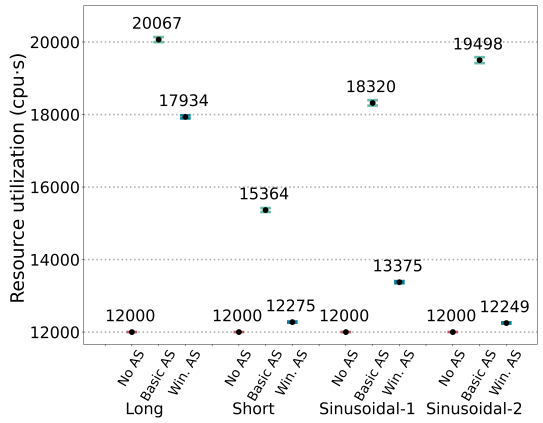
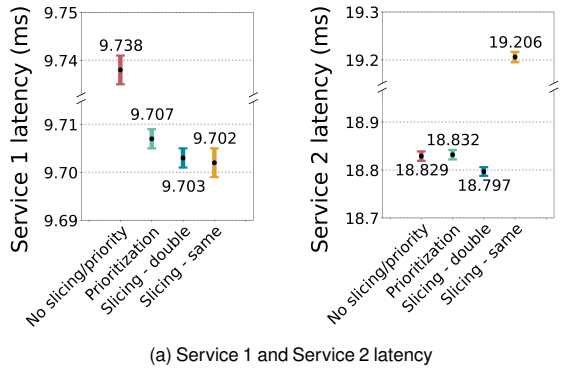
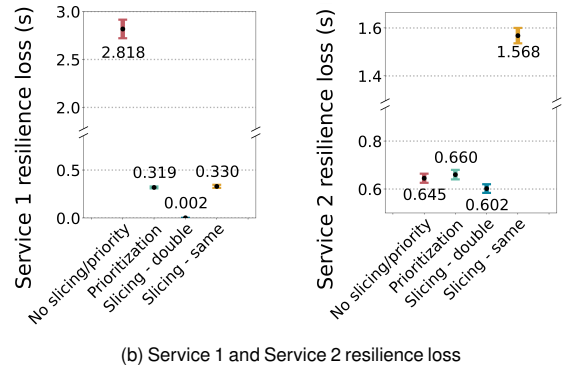


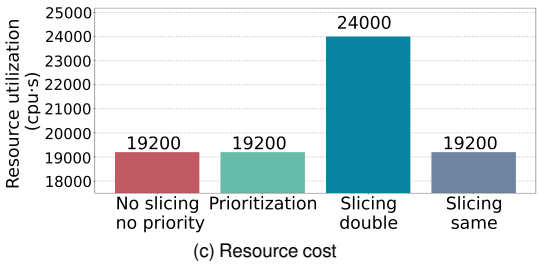
Fig. 15. Resource cost values and confidential intervals in case 1.



(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.

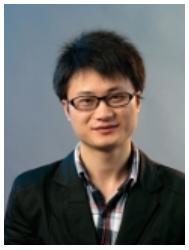
He is currently pursuing the Ph.D. degree in complex system engineering at CentraleSupélec, University of Paris-Saclay, Gif-sur-Yvette, France. He is also a CIFRE fellow working in Orange Innovation. His current research interests include complex networked system modeling, telecommunication network system 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.