*Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference*
*Edited by* Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen
 *Published by* Research Publishing, Singapore.
doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P5793-cd

# Optimizing 5G Network Availability: Comparison Between Reliability Block Diagrams and Markov Chains Approaches

Ikram Temericht, Bertrand Decocq, Guillaume Fraysse, Anne Barros, Yiping Fang, Zhiguo Zeng

As essential services increasingly rely on digital platforms, ensuring high availability has become a critical priority. The virtualized architecture of 5G networks provides enhanced flexibility and scalability, but it also introduces challenges in maintaining system availability, particularly for sectors such as industrial IoT, where even brief service interruptions can lead to significant costs. This paper addresses the challenge of designing 5G network architectures that meet stringent availability requirements while adhering to budget constraints. We introduce a model based on Markov Chains to assess system availability, noting that although Markov Chains are rigorous in probabilistic modeling, they can become computationally complex in large, interconnected systems. To address this limitation, we introduce Reliability Block Diagrams (RBDs) as an alternative approach.
Our study evaluates both models, highlighting that RBDs offer a more scalable, easier-to-use approach for virtualized networks. The analysis reveal that strategic redundancy placement significantly enhances system resilience in 5G networks and beyond.

*Keywords*: 5G Networks, Network Resilience, Virtualization, Availability, Reliability Block Diagrams (RBDs), Markov Chains, Redundancy, Service Chains, Availability Optimization

## 1. Introduction

5G networks are central to digital transformation, enabling critical services in sectors such as the Industrial Internet of Things (IIoT), business-to-business (B2B) applications, and smart infrastructure. Their fully virtualized architecture offers flexibility and scalability, allowing dynamic adaptation of resources to meet demand. However, virtualization also presents challenges, particularly in maintaining network resilience and service availability [1].

Availability is a crucial metric for evaluating 5G network resilience, as even brief service interruptions can have significant financial and operational consequences. In sectors like healthcare, Industry 4.0, and critical infrastructure supervision (e.g., energy and transportation), network failures can lead to production downtimes or data loss. Ensuring high availability in complex virtualized systems while respecting budgetary constraints is therefore critical [2]. Recent studies have explored advanced reliability approaches, such as predictive maintenance and intelligent fault detection, which can be leveraged to enhance availability analysis [3].

This study aims to evaluate the availability of 5G networks using two modeling techniques: Markov Chains[4] and Reliability Block Diagrams (RBDs)[5]. Markov Chains provide a detailed probabilistic model but can be computationally intensive for large, interconnected systems like 5G networks. RBDs, on the other hand, offer a simpler, more scalable approach, making them well-suited for virtualized architectures. The main contribution of this work is a comparative analysis of Markov Chains and Reliability Block Diagrams (RBDs) to assess their effectiveness in modeling 5G network availability. By evaluating their computational complexity, scalability, and accuracy, this study provides insights into the suitability of each technique for analyzing virtualized communication systems, helping researchers and network operators make informed choices in reliability modeling. Additionally, this work explores various 5G architecture scenarios and use cases, offering a comprehensive evaluation of their impact on network reliability and performance.

The structure of this paper is organized as follows. Section 2 introduces the system model and architecture scenarios, detailing various network configurations and redundancy setups. Section 3

outlines the methodology, including parameters, assumptions and the use of Markov Chains and Reliability Block Diagrams (RBD) for availability analysis. Section 4 presents the results of our study,demonstrating the impact of redundancy and replicas on system availability. Finally, Section 5 concludes the paper with insights and recommandations for optimizing 5G network resilience.

## 2. System Model and Architecture Scenarios

The communication system modeled in this study consists of a chain of Virtual Network Functions (VNFs), represented as shown in Figure 1 :

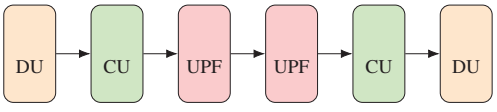



Fig. 1. VNF Chain

Each VNF plays a critical role in the 5G network architecture:

- **Distributed Unit (DU):** The DU processes lower-layer Radio Access Network (RAN) protocols, managing real-time tasks such as scheduling, error correction, and signal modulation. It interfaces directly with radio hardware, ensuring efficient real-time data delivery and user access.
- **Centralized Unit (CU):** The CU manages higher-layer RAN protocols, including Radio Resource Control (RRC) and Packet Data Convergence Protocol (PDCP). By centralizing non-real-time functions, it optimizes resource allocation, traffic flow, and user mobility.
- **User Plane Function (UPF):** The UPF, part of the 5G core, is responsible for data packet forwarding and routing. It supports Quality of Service (QoS) enforcement, traffic steering, and session handling.

In this 5G network architecture, User Plane Functions (UPFs) reside in the core, Central Units (CUs) at the Edge, and Distributed Units (DUs) at the access level. Two UPFs manage data traffic across coverage areas, reducing latency and ensuring seamless communication. The network relies on microservices which are small, independent software components deployed as containers orchestrated by Kubernetes (K8s) for efficient management, scalability, and fault tolerance. These containers run within pods, with multiple replicas ensuring redundancy and load balancing. To provide isolation and resource control, Kubernetes pods are deployed inside Virtual Machines (VMs) running on bare-metal servers. This hybrid approach leverages the flexibility of containers and the security of VMs, ensuring a robust, scalable, and resilient 5G network. Kubernetes automates deployment and scaling across VMs, optimizing performance while maintaining high availability.

The system is subject to potential failures and repairs which are essential for modeling system resilience. The values for these rate are taken from the study conducted in [4]. These rates are provided in Table 1.

Table 1. Repair and Failure rates of the components

| | Repair rate ($h^{-1}$) | Failure rate ($h^{-1}$) |
|---|---|---|
| Virtual machine | 360 | 0.005 |
| Baremetal server | 1 | 0.0002 |

### 2.1. *Architecture Scenarios and Analysis*

This subsection presents the scenarios used to evaluate the impact of various configurations on system availability. Each scenario combines a description of the setup and its analysis, providing a cohesive view of how different configurations affect system performance.

#### 2.1.1. *Scenario 1: Reference Communication System (Base Case)*

**Description and Analysis:**

This scenario models a basic communication system consisting of a chain composed of a single instance of each VNF described in Fig. 1, each composed of a user-defined number of microservices. The system lacks redundancy or replicas, making it highly vulnerable to failures. In this

configuration, a single component failure results in a complete system failure due to the series nature of the architecture. The system is expected to exhibit low availability, which will be confirmed through testing.

#### 2.1.2. *Scenario 1 bis: Fully Redundant End-to-End VNF Chain Without Microservice Replicas*

**Description and Analysis:**

This scenario is derived from Scenario 1 by introducing full redundancy across the entire VNF chain. Microservice replicas are not included in this configuration. Despite the absence of microservice-level replication, the system is expected to achieve very high availability due to the redundancy at the VNF level, ensuring operational continuity in the face of component failures which will be verified during testing.

#### 2.1.3. *Scenario 2: Communication System with Microservice Replicas*

**Description and Analysis:**

In this scenario, each microservice within a VNF is equipped with a user-defined replicas to enhance fault tolerance. This parallel configuration ensures that the system remains operational as long as at least one replica per microservice remains functional. The scenario is proposed in order to have an improved availability comparing to Scenario 1, which will be confirmed after testing.

#### 2.1.4. *Scenario 3: Microservice Replicas + Core VNF Redundancy*

**Description and Analysis:**

Building on Scenario 2, this setup introduces redundancy to critical core VNFs, (UPF), alongside microservice replicas. Redundancy at the core level ensures that the system can maintain critical networl functions even if a core VNF instance fails. The system is made in place to show a marked improvement in availability, which will be confirmed through testing.

#### 2.1.5. *Scenario 4: Microservice Replicas + Redundancy of Core and Edge VNFs*

**Description and Analysis:**

This scenario extends redundancy to both core VNFs (UPFs) and edge VNFs (CUs), alongside microservice replicas. Redundancy enhances fault tolerance for critical system functions. The system is expected to have significantly improved resilience and availability, which will be validated during testing.

#### 2.1.6. *Scenario 5: Fully Redundant End-to-End VNF Chain Except for DU*

**Description and Analysis:**

This scenario introduces a single point of failure at the Distributed Unit (DU) level. All other VNFs, including CUs and UPFs, remain fully redundant with microservice replicas. The system is tested to maintain a higher level of availability compared to Scenario 1, which will be confirmed through testing.

#### 2.1.7. *Scenario 6: Fully Redundant End-to-End VNF Chain*

**Description and Analysis:**

This scenario represents the most resilient configuration, applying full redundancy across the entire VNF chain, including DUs, CUs, and UPFs. Each microservice includes replicas, and all VNFs have redundant instances. The system is expected to achieve near-perfect availability, demonstrating the maximum resilience achievable within the modeled architecture. These expectations to be confirmed through testing.

### 2.2. *Scenario Configurations and Considerations*

The parameters for each scenario are detailed below. These include the number of microservices, replicas, and redundancies configured to test the impact of different fault tolerance strategies.

#### 2.2.1. *Scenario 1: Reference Communication System (Base Case)*

For scenario 1 each VNF is composed of two microservices. In this configuration, no replicas are instantiated for the microservices, and no redundancy is applied to the VNFs. The scenario is represented in figure 2.

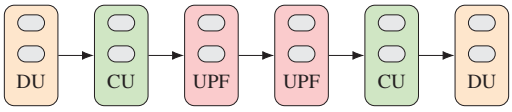
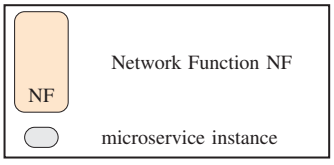



Fig. 2. Scenario 1: Reference communication system, architecture




Legend

#### 2.2.2. *Scenario 1 bis: Fully Redundant End-to-End VNF Chain Without Microservice Replicas*

This scenario is an extension of Scenario 1, where full redundancy is applied to all VNFs in the chain with two redundant instances. The scenario is represented in figure 3 .

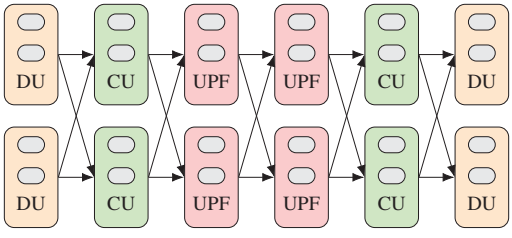



Fig. 3. Scenario 1 bis: Fully redundant end-to-end VNF chain without microservice Replicas

#### 2.2.3. *Scenario 2: Communication System with Microservice Replicas*

Scenario 2 introduces microservice-level replication. Each VNF consists of two microservices, with each microservice deployed as separate pods in Kubernetes (K8S). Each microservice pod is instantiated with two replicas for high availability and fault tolerance. The scenario is represented in figure 4 .

#### 2.2.4. *Scenario 3: Microservice Replicas + Redundancy of Core VNFs*

In addition to microservice replication (as in Scenario 2), each core VNF(UPF) is configured with two redundant instances, implemented as separate Kubernetes deployments. The scenario is represented in figure 5.

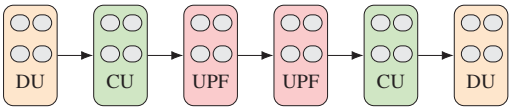



Fig. 4. Scenario 2: Communication system with microservice replicas

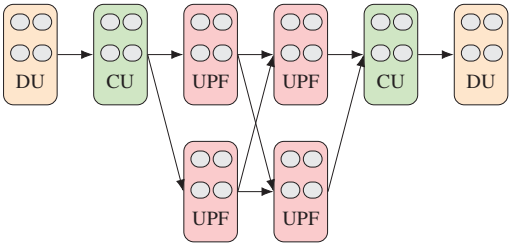



Fig. 5. Scenario 3: Communication system with microservice replicas and redundancy of core VNFs (UPF Redundancy)

#### 2.2.5. *Scenario 4: Microservice Replicas + Redundancy of Core and Edge VNFs*

Building upon Scenario 3, Scenario 4 adds redundancy to both core VNFs (UPF) and edge VNFs (CU). The scenario is represented in figure 6.

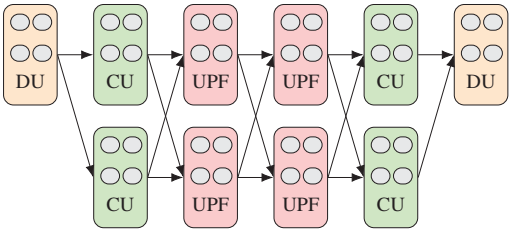



Fig. 6. Scenario 4: Communication system with microservice replicas and redundancy of core and Edge VNFs (CU redundancy)

#### 2.2.6. *Scenario 5: Fully Redundant End-to-End VNF Chain Except for One DU*

Each VNF consists of two microservices, deployed as Kubernetes pods with two replicas, and

all VNFs (CU, one DU and UPF) are configured with two redundant instances via Kubernetes deployments.The scenario is represented in figure 7.

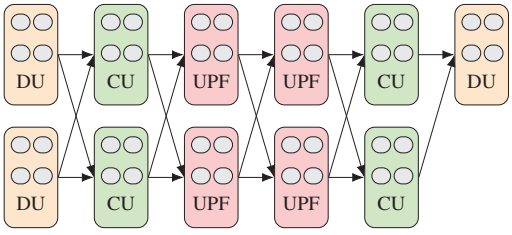



Fig. 7. Scenario 5: Fully redundant end-to-end VNF chain except for one DU

#### 2.2.7. *Scenario 6: Fully Redundant End-to-End VNF Chain*

Each VNF consists of two microservices with two replicas, and all VNFs (CU, DU, and UPF) are configured with two redundant instances. The scenario is represented in figure 8.

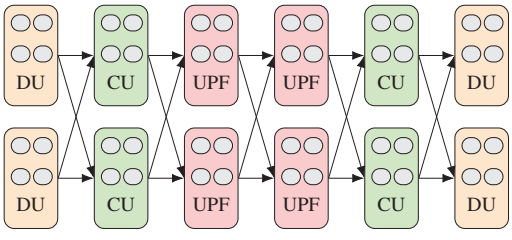



Fig. 8. Scenario 6: Fully redundant end-to-end VNF chain

## 3. Methodology

In this section, we outline the methodology for evaluating the availability of the system in Scenario 1 using two analytical methods: Markov Chains and Reliability Block Diagrams (RBDs). We assume that each VNF in the chain comprises two microservices, with no replicas or redundancy.

### 3.1. *Markov Chain Method*

The Markov Chain method[4] models the system as a set of states representing the operational and failed conditions of its components. Markov Chains are stochastic processes that undergo transitions from one state to another within a finite or countable number of possible states, where the probability of each transition depends solely on the current state. In reliability engineering, this method is particularly useful for modeling systems with components that have constant failure and repair rates, allowing for the analysis of system behavior over time .

For demonstration, we compute the availability of the first Virtual Network Function (VNF) in the chain (DU) using a Markov Chain model. The VNF consists of two microservices ($MS_1, MS_2$), each running on a dedicated Virtual Machine ($VM_1, VM_2$) hosted on a Baremetal (BM) server. The methodology is generalized to the entire chain of VNFs later.

#### 3.1.1. *State Space Definition*

Each component ($BM_1, BM_2, VM_1, VM_2, MS_1, MS_2$) can be either **Operational** (1) or **Failed** (0). Given that there are two Baremetal servers, two Virtual Machines, and two Microservices, there are $2^6 = 64$ possible states. However, we assume that a microservice fails only when the underlying VM hosting it fails since microservices are software components that rely on Virtual Machines for execution. If a VM fails, the microservices running on it also fail. Therefore, we do not explicitly model the failure and repair rates of microservices separately; they are implicitly included in the VM states., reducing the number of effective states. Under this assumption, the state space is defined by the four main components: two Baremetal servers and two Virtual Machines, resulting in $2^4 = 16$ states.

Each state represents the status of the entire VNF, and we list them in Table 2:

#### 3.1.2. *Transition Matrix*

The transitions between states depend on the failure and repair rates of each component:

- $\lambda_{\text{BM}}$: Failure rate of a Baremetal server.
- $\mu_{\text{BM}}$: Repair rate of a Baremetal server.
- $\lambda_{\text{VM}}$: Failure rate of a Virtual Machine.

Table 2. State representation of VNF availability

| State | $BM_1$ | $BM_2$ | $VM_1$ | $VM_2$ | Status of VNF |
|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | Failed |
| 2 | 0 | 0 | 0 | 1 | Failed |
| 3 | 0 | 0 | 1 | 0 | Failed |
| 4 | 0 | 0 | 1 | 1 | Failed |
| 5 | 0 | 1 | 0 | 0 | Failed |
| 6 | 0 | 1 | 0 | 1 | Failed |
| 7 | 0 | 1 | 1 | 0 | Failed |
| 8 | 0 | 1 | 1 | 1 | Operational |
| 9 | 1 | 0 | 0 | 0 | Failed |
| 10 | 1 | 0 | 0 | 1 | Failed |
| 11 | 1 | 0 | 1 | 0 | Failed |
| 12 | 1 | 0 | 1 | 1 | Operational |
| 13 | 1 | 1 | 0 | 0 | Failed |
| 14 | 1 | 1 | 0 | 1 | Operational |
| 15 | 1 | 1 | 1 | 0 | Operational |
| 16 | 1 | 1 | 1 | 1 | Operational |

- $\mu_{\text{VM}}$: Repair rate of a Virtual Machine.

Since we assume that microservices fail only when their respective VM fails, the microservice failure rate does not appear explicitly in the transition matrix. The transition matrix ($\mathbf{Q}$) is a $16 \times 16$ matrix, where:

$$Q_{ij} = \begin{cases} -\sum_{k \neq i} Q_{ik}, & \text{if } i = j, \\ \text{Transition rate from state } i \text{ to } j, & \text{if } i \neq j. \end{cases}$$

Here is a partial representation of the matrix:

$$\mathbf{Q} = \begin{bmatrix} -\sum_j Q_{1j} & \lambda_{BM} & \lambda_{VM} & \cdots & 0 \\ \mu_{BM} & -\sum_j Q_{2j} & \lambda_{VM} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \mu_{VM} & -\sum_j Q_{16j} & \end{bmatrix}$$

#### 3.1.3. *System Availability*

To compute the system availability:

(1) Solve the balance equations for the steady-state probabilities:

$$\boldsymbol{\pi} \cdot Q = 0, \tag{1}$$

$$\sum_{i=1}^{16} \pi_i = 1. \tag{2}$$

(2) Sum the probabilities of all operational states:

$$A_{VNF} = \sum_{\text{Operational states}} \pi_i. \tag{3}$$

For a chain of 6 VNFs, assuming independence:

$$A_{\text{System}} = A_{VNF}^6. \tag{4}$$

Substituting the computed availability of a single VNF ($A_{VNF} \approx 0.9983249$):

$$A_{\text{System}} = (0.9983249)^6 \approx 0.98999148. \tag{5}$$

### 3.2. *Reliability Block Diagrams (RBD) Method*

The RBD method is a graphical representation used to analyze the reliability and availability of a system by modeling the logical relationships between its components. In an RBD, system components are represented as blocks connected in series or parallel configurations, where series connections indicate that the failure of any single component leads to system failure, while parallel connections provide redundancy, improving overall system reliability [5]. The system is represented as follow:

- Each microservice ($MS_1, MS_2$) is modeled as a **series system** of its VM and the hosting baremetal server.
- Each VNF is modeled as a **series system** of two microservices.
- The entire chain of VNFs is modeled as a **series system**.

To calculate the availability of the system in Scenario 1:

(1) The availability of the Baremetal server ($A_{\text{BM}}$) is given by:

$$A_{\text{BM}} = \frac{\mu_{\text{BM}}}{\mu_{\text{BM}} + \lambda_{\text{BM}}} = 0.999175$$

(2) The availability of the Virtual Machine ($A_{\text{VM}}$) is:

$$A_{\text{VM}} = \frac{\mu_{\text{VM}}}{\mu_{\text{VM}} + \lambda_{\text{VM}}} = 0.9999861$$

(3) The availability of an Instance ($A_{\text{Instance}}$) is the product of the availability of the Virtual Machine and the Baremetal server:

$$A_{\text{Instance}} = A_{\text{VM}} \times A_{\text{BM}} = 0.999162$$

(4) The availability of the Microservice ($A_{\text{MS}}$) is equal to the availability of the Instance:

$$A_{\text{MS}} = A_{\text{Instance}} = 0.999162$$

(5) The availability of the Virtual Network Function ($A_{\text{VNF}}$) is the square of the availability of the Managed Service:

$$A_{\text{VNF}} = A_{\text{MS}}^2 = 0.9983249$$

(6) Finally, the availability of the system ($A_{\text{System}}$) is the sixth power of the availability of the Virtual Network Function:

$$A_{\text{System}} = A_{\text{VNF}}^6 = 0.98999148$$

### 3.3. *Generalization to Other Scenarios*

All calculations in this methodology are based on established analytical formulas for reliability and availability evaluation. To extend these calculations to larger and more complex systems, we developed codes that support both the Markov Chain and RBD methods and automate the computation of availability metrics for systems with numerous components and configurations, ensuring accuracy and scalability.

The methodologies described can be extended to more complex scenarios by modifying the system configuration:

- Adjust the number of microservices per VNF.
- Introduce replicas at the microservice level.
- Add redundancy for core or edge VNFs.

These adjustments affect the structure of the RBD and the transition matrix for the Markov Chain, allowing for flexible modeling of various configurations.

## 4. Results

This section presents the system's availability analysis using both Markov Chains and Reliability Block Diagrams (RBDs).

### 4.1. *Comparative Results*

Table 3 summarizes the availability results obtained with both methods. Although the computed availability values are nearly identical, the execution time (E.T) reveals a major computational advantage for RBDs.

Table 3. Availability results obtained with both methods, including computational time.

| Sc. | Availability (%) | | Unavail. | E.T [s] | |
|---|---|---|---|---|---|
| | Markov | RBD | [h] | Markov | RBD |
| 1 | 98.999148 | 98.999148 | 87.08 | 12.5 | 2.1 |
| 2 | 99.955182 | 99.955183 | 3.93 | 18.2 | 2.8 |
| 3 | 99.970118 | 99.970118 | 2.62 | 25.7 | 3.5 |
| 4 | 99.985056 | 99.985055 | 1.31 | 38.5 | 3.9 |
| 1b | 99.991182 | 99.991182 | 0.77 | 52.4 | 4.6 |
| 5 | 99.999826 | 99.999826 | 0.015 | 120.3 | 5.6 |
| 6 | 99.999997 | 99.999998 | 0.00026 | 210.7 | 7.8 |

While Markov Chains provide a rigorous state-based approach, their complexity increases exponentially with system size, making them impractical for large architectures. In contrast, RBDs follow a more straightforward combinatorial approach, significantly reducing computation time:

- **Computational Complexity:** Markov Chains require solving large state-space models, leading to an exponential increase in computation time ($\mathcal{O}(n^2)$ in basic cases, reaching $\mathcal{O}(2^n)$ for highly interconnected systems). RBDs, however, operate with polynomial complexity ($\mathcal{O}(n)$).
- **Scalability:** The exponential state explosion in Markov models makes them unsuitable for real-time evaluations of large-scale networks, whereas RBDs provide a practical balance between accuracy and efficiency.

These results confirm that while Markov Chains offer detailed probabilistic modeling, RBDs are computationally more efficient and scalable, making them the preferred method for large systems.

### 4.2. *Impact of Redundancy and Replication*

The results also highlight the positive impact of redundancy on system availability. Figure 9 illustrates how different redundancy mechanisms, such as microservice replicas and Virtualized Network Function (VNF) redundancy, improve availability and reduce unavailability (1 - availability).

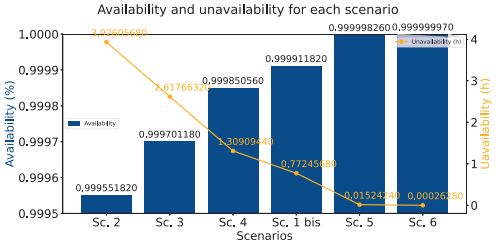



Fig. 9. Availability improvements across scenarios.

Even minimal redundancy, such as microservice replication, significantly enhances fault tolerance by reducing failure likelihood. However, beyond a certain point, additional redundancy yields diminishing returns and may introduce unnecessary costs. A well-balanced redundancy strategy ensures high availability while optimizing resource usage.

This tool enables network planners to make informed decisions, ensuring service reliability while avoiding excessive complexity and resource consumption.

## 5. Conclusion

This study provides a comparative analysis of Reliability Block Diagrams (RBDs) and Markov Chains for evaluating the availability of virtualized communication systems, a critical factor in system resilience. While both methods yield reliable results, RBDs prove more efficient and practical, especially for large-scale and complex systems. Redundancy in microservices and Virtualized Network Functions (VNFs) significantly enhances fault tolerance, ensuring continued operation despite component failures. Although Markov Chains offer detailed probabilistic modeling, RBDs excel in computational simplicity, scalability, and ease of implementation, making them ideal for evaluating complex systems. This approach facilitates faster decision-making and iterative design, emphasizing the need for well-optimized redundancy strategies. By aligning redundancy levels with operational priorities, systems achieve greater resilience without compromising performance. Ultimately, RBDs emerge as a practical and effective tool for modeling availability in virtualized systems like 5G networks. Future research could explore hybrid models that combine the strengths of RBDs and Markov Chains to address the increasing complexity of next-generation network systems.

## Acknowledgment

This work has been partly funded by the European Union's Horizon Europe research and innovation program under grant agreement No. 101095759 (Hexa-X-II).

## References

1. André Cardoso, Pedro R. M. Inácio, Maribel Yasmina Santos, and André Zúquete. Resilient and secure network slicing for 5g systems: A comprehensive survey. *Computers & Security*, 92:101747, 2020.
2. Xin Wang, Pingyi Fan, Hui Tian, and Chenyang Yang. A survey on 5g security: Challenges and solutions. *IEEE Access*, 8:124166–124196, 2020.
3. H. Zhang, Y. Wang, and J. Liu. Process monitoring for tower pumping units under variable operational conditions: From an integrated multitasking perspective. *IEEE Transactions on Industrial Informatics*, 17(3):1951–1962, 2021.
4. Rui Li, Bertrand Decocq, Anne Barros, Yiping Fang, and Zhiguo Zeng. High-mobility 5g communication service: availability and reliability analysis. HAL Id: hal-04313077, 2023. Submitted on 28 Nov 2023.
5. Marvin Rausand, Anne Barros, and Arnljot Hoyland. *System Reliability Theory: Models, Statistical Methods, and Applications*. Wiley, 3rd edition, 2011.