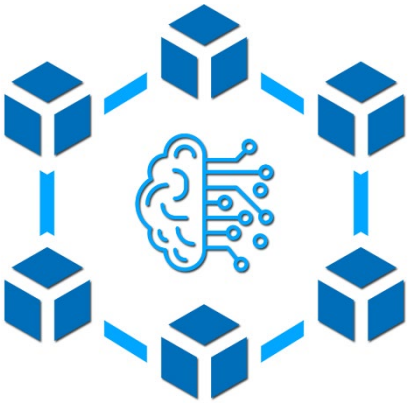


An Artificial Intelligent Aided Unified Network for Secure Beyond 5G Long Term Evolution
[GA: 101096456]

Deliverable 2.2

NANCY Experimental-Driven Modelling

Programme: HORIZON-JU-SNS-2022-STREAM-A-01-06

Start Date: 01 January 2023

Duration: 36 Months

Co-funded by the European Union

6GSNS

NANCY project has received funding from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101096456.

Document Control Page

Deliverable Name	NANCY Experimental-Driven Modelling
Deliverable Number	D2.2
Work Package	WP2 - Usage Scenario and B-RAN Modelling, Network Requirements, and Research Framework
Associated Task	T2.2 - Experimental-driven B-RAN & Attack Modelling
Dissemination Level	Public
Due Date	30 April 2025 (M28)
Completion Date	29 April 2025
Submission Date	30 April 2025
Deliverable Lead Partner	INNO
Deliverable Author(s)	Stylianos Trevlakis (INNO), Lamprini Mitsiou (INNO), Eirini Gkarnetidou (INNO), Vasileios Kouvakis (INNO), Theodoros Tsiftsis (INNO), Javier Vicente (NEC), Panagiotis Sarigiannidis (UOWM), Thomas Lagkas (UOWM), Athanasios Liatifis (UOWM), Dimitrios Pliatsios (UOWM), Sotirios Tegos (UOWM), Nikolaos Mitsiou (UOWM), Vasiliki Koutsioumpa (UOWM), Pigi Papanikolaou (UOWM), Ioannis Makris (MINDS), Nikolaos Ntampakis (MINDS), Dimitrios-Christos Asimopoulos (MINDS), Konstantinos Kyranou (SID), Georgios Michoulis (SID), Thomai Karamitsiou (SID), Ioannis Hadjigeorgiou (SID)
Version	1.0

Document History

Version	Date	Change History	Author(s)	Organisation
0.1	12/5/2023	Table of Contents	Stylianos Trevlakis, Lamprini Mitsiou, Vasileios Kouvakis, Theodoros Tsiftsis	INNO
0.2	12/1/2025	Updated ToC	Stylianos Trevlakis, Lamprini Mitsiou, Eirini Gkarnetidou, Vasileios Kouvakis, Theodoros Tsiftsis	INNO
0.3	4/3/2025	Section 4.1.1.	Javier Vicente	NEC
0.4	19/3/20225	Section 5 & 6	Stylianos Trevlakis, Lamprini Mitsiou, Eirini Gkarnetidou, Vasileios Kouvakis, Theodoros Tsiftsis	INNO

0.5	21/3/2025	Section 4.1.2.	Panagiotis Sarigiannidis, Thomas Lagkas, Athanasios Liatifis, Dimitrios Pliatsios, Sotirios Tegos, Nikolaos Mitsiou, Vasiliki Koutsioumpa, Pigi Papanikolaou/Ioannis Makris, Nikolaos Ntampakis, Dimitrios-Christos Asimopoulos	UOWM/MINDS
0.6	28/3/2025	Section 4.1.1.	Konstantinos Kyranou, Georgios Michoulis, Thomai Karamitsiou, Ioannis Hadjigeorgiou	SID
0.7	1/4/2025	Section 1 & 7, Editing and first complete draft	Stylianos Trevlakis, Eirini Gkarnetidou, Theodoros Tsiftsis	INNO
0.8	24/4/2025	Editing and review	Dimitrios Pliatsios	UOWM
	25/4/2025	Editing and review	Dimitris Manolopoulos	UBI
0.9	25/4/2025	Addressing internal reviewer comments	Stylianos Trevlakis, Eirini Gkarnetidou	INNO
1.0	29/4/2025	Final Version After Quality Check	Anna Triantafyllou, Dimitrios Pliatsios	UOWM

Internal Review History

Name	Organisation	Date
Dimitrios Pliatsios	UOWM	24/4/2025
Dimitris Manolopoulos	UBI	25/4/2025

Quality Manager Revision

Name	Organisation	Date
Anna Triantafyllou, Dimitrios Pliatsios	UOWM	28/4/2025

Table of Contents

List of Figures 7
List of Tables 8
List of Acronyms 9
Executive Summary 10
1. Introduction 11
1.1. Purpose of the Document 11
1.2. Relation to other Tasks and Deliverables 11
1.3. Structure of the Document 11
2. The role of B-RAN modelling in NANCY 14
2.1. Relation to the NANCY usage scenarios 15
2.1.1. Fronthaul network of fixed topology 15
2.1.2. Advanced coverage expansion 16
2.1.3. Advanced connectivity of mobile nodes 16
2.2. Network Topology 16
3. B-RAN Modelling 18
3.1 Markov Chain Model 19
3.2 Hierarchical B-RAN Model 21
3.3 Performance Evaluation 22
3.4 Numerical Results 27
4. Attack Modelling 35
4.1. Identified Attacks 35
4.1.1. Attacks on the Blockchain 35
4.1.2. Attacks on the Network 40
4.2. Simulated Attacks 43
4.2.1. 51% attack modelling 43
4.2.2. Sybil attack modelling 44
5. Unified B-RAN and Attacks Modelling 47

5.1. Performance Evaluation 47
5.1.1. 51% attack 47
5.1.2. Sybil attack 49
5.2. Numerical Results 51
5.2.1. 51% attack 51
5.2.2. Sybil attack 54
6. Intelligent Optimisation 56
6.1. Reinforcement Learning Framework 56
6.2. Performance Evaluation 58
7. Conclusion 63
Bibliography 65

List of Figures

Figure 1. Network topology 15
Figure 2. B-RAN architecture 18
Figure 3. B-RAN Markov chain model 19
Figure 4. Hierarchical blockchain architecture 21
Figure 5. State space scenario for k = 2 and r = 1 25
Figure 6. Transition rate matrix 26
Figure 7. Latency vs N for multiple k and ρ combinations 28
Figure 8. Latency vs N in single blockchain 29
Figure 9. Latency vs traffic intensity for single blockchain for different k values 30
Figure 10. Latency vs traffic intensity of the HB-RAN 31
Figure 11. Performance comparison across various traffic scenarios and k values 33
Figure 12. Latency vs N of the hierarchical B-RAN with multiple intensities 34
Figure 13. Selfish Mining Attack 36
Figure 14. DoS attacks against O-RAN 41
Figure 15. Reconnaissance attacks at O-RAN component 42
Figure 16. MITM attack between the two RICs 43
Figure 17. Procedural illustration of an alternate history attack in B-RAN 44
Figure 18. Probability of successful attack vs the attacker's mining power for multiple combinations of Ng and N 52
Figure 19. Security vs latency for different s and k configurations 53
Figure 20. Probability of a successful Sybil attack as a function of β 54
Figure 21. RL Framework 58
Figure 22. Latency over time for different RL agency activation delay scenarios 59
Figure 23. Latency over time for multiple traffic scenarios 62

List of Tables

Table 1. Performance metrics with and without HSM 39

List of Acronyms

Acronym	Explanation
RANs	Radio Access networks
MC	Markov Chain
B-RAN	Blockchain-Radio Access Network
HB-RAN	Hierarchical Blockchain-Radio Access Network
M/M/1	Single-server queueing model
M/M/s	Multi-server queueing model
MEC	Multiple-access Edge Computing
AI	Artificial Intelligence
UE	User Equipment
BSs	Base Stations
CoMP	Coordinated Multi-Point
V2V	Vehicle-to-Vehicle
p2p	point-to-point
PoW	Proof of Work
PoS	Proof of Stake
BFT	Byzantine Fault Tolerance
SLA	Service Level Agreement
IU	Intermediate User
RL	Reinforcement Learning
PPO	Proximal Policy Optimization
MITM	Man-in-the-Middle
O-RAN	Open Radio Access Network
DOS	Denial-of-Service
DDOS	Distributed Denial-of-Service
CPU	Central Processing Unit
GPU	Graphics Processing Unit
RAM	Random-Access Memory
OS	Operating System
HSM	Hardware Security Module
TX	Transistorized Experimental
ARP	Address Resolution Protocol
P2P	Peer-to-Peer

Executive Summary

This deliverable, namely "D2.2 – NANCY Experimental-Driven Modelling", presents the theoretical modelling frameworks for the Blockchain-Radio Access Network (B-RAN) architecture within the NANCY project. The core objective is to rethink the roles of connectivity providers and service consumers, where a consumer can also act as a provider. The document focuses on three NANCY usage scenarios: fixed fronthaul network, advanced coverage expansion, and advanced connectivity of mobile nodes.

The deliverable introduces a novel B-RAN architecture and attack modelling approach. It employs Markov chain theory to model the probabilistic transitions between different system states in both single-chain B-RAN and a Hierarchical B-RAN (HB-RAN) architecture designed for coverage expansion scenarios. The HB-RAN model utilizes nested blockchains with intermediate users to extend network coverage securely.

Performance evaluation, primarily focusing on latency, is conducted for both B-RAN and HB-RAN using queueing theory (M/M/1 and M/M/s models) and Markov chain analysis. Numerical results demonstrate the impact of parameters like the number of confirmations (N), block capacity (k), and traffic intensity (ρ) on latency. The HB-RAN analysis reveals a performance trade-off due to the introduction of the secondary blockchain.

The deliverable also investigates the security landscape of B-RAN, identifying and modelling prominent attacks, particularly the 51% attack and the Sybil attack. Closed-form expressions for the probability of successful attacks are derived based on the attacker's mining power and network parameters. Numerical results highlight the relationship between security (attack probability) and latency, demonstrating an inherent trade-off. Increasing the number of confirmations generally enhances security but increases latency.

Finally, the deliverable explores intelligent optimisation of B-RAN using a Reinforcement Learning (RL) framework. A Proximal Policy Optimization (PPO) based RL agent is developed to dynamically adjust blockchain parameters at runtime to maintain target latency while optimizing resource utilization. Simulation results demonstrate the RL agent's effectiveness in mitigating traffic surges and stabilizing network performance. The findings underscore the potential of AI-based techniques for enhancing the efficiency and resilience of B-RAN systems.

1. Introduction

1.1. Purpose of the Document

This deliverable, namely "D2.2 – NANCY Experimental-Driven Modelling", presents a new model in which service consumers may also act as connectivity providers, and provides theoretical modelling frameworks for the B-RAN architecture, which revolutionizes wireless networks, by allowing a service consumer to simultaneously be a service provider. It focuses on the three NANCY usage scenarios, i.e., fixed fronthaul network, advanced coverage expansion, and advanced connectivity of mobile nodes. D2.2 also investigates the security landscape of B-RAN, recognizes the 51% and Sybil attacks as the most impactful, and incorporate them in the theoretical modelling framework. It capitalises on Markov-Chain theory and artificial intelligence, and extracts closed-form expressions for the average latency and probability of successful attack. Finally, AI-based techniques are provided for the intelligent optimisation of B-RAN.

1.2. Relation to other Tasks and Deliverables

This deliverable builds upon previous research and findings within the NANCY project as follows:

- “D2.1 – NANCY Requirements Analysis” provides a strong foundation for the construction of the experimental-driven modelling in terms of the NANCY’s vision, use cases, usage scenarios, and requirements.
- “D3.1 – NANCY Architecture Design” describes the NANCY architecture along with details about its various components. A preliminary analysis of the experimental-driven modelling approaches of NANCY is included in D3.1.

1.3. Structure of the Document

The rest of the document is structured as follows:

- **Section 2 – The role of B-RAN modelling in NANCY:** This section addresses the necessity of providing intelligence, energy efficiency, and security at the network's edge, leading to the transformation of Radio Access Networks (RANs). It highlights the potential of incorporating blockchain into RANs (B-RAN) as a promising countermeasure to security risks by offering decentralized tamper-proof solutions. The section discusses how B-RAN enables secure, private, and dependable collaboration among service providers and users by integrating blockchain with virtualization, Multiple-access Edge Computing (MEC), and Artificial Intelligence (AI). It also details the relation of B-RAN modelling to the three NANCY usage scenarios: fronthaul network of fixed topology, advanced coverage expansion, and advanced connectivity of mobile nodes. The section further describes the network topology, considering both point-to-point and multi-hop

connectivity scenarios, and introduces the concept of a hierarchical B-RAN (HB-RAN) architecture for advanced coverage expansion.

- **Section 3 – B-RAN modelling:** This section presents the theoretical modelling frameworks for the B-RAN architecture. It employs a Markov chain model to capture the probabilistic transitions between different system states and analyses their dynamic behaviour during B-RAN operations. The section also introduces an extension of the single-chain B-RAN model, the HB-RAN model, designed for assessing the performance of coverage expansion scenarios in terms of security and reliability. Furthermore, this section delves into performance evaluation by mathematically modelling latency using queuing theory (M/M/1 and M/M/s queues) to represent request processing in blockchain blocks and service initiation. Finally, the section presents numerical results obtained from the proposed B-RAN model, discussing its performance and highlighting valuable design guidelines through simulations.
- **Section 4 – Attack modelling:** This section investigates the security landscape of B-RAN by first presenting the most prominent attacks, focusing on both blockchain and network aspects. It identifies and describes attacks such as the 51% attack, selfish mining, Man-in-the-Middle (MITM) attack, collusion attacks, and private key compromise in validator nodes within the blockchain domain. In the network domain, it discusses threats to Open Radio Access Network (O-RAN) such as Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks, reconnaissance attacks, MITM attacks, and eavesdropping. The section then details and models the 51% attack and Sybil attack as the most impactful threats. Finally, it prepares these selected attacks for integration with the B-RAN theoretical model.
- **Section 5 – Unified B-RAN and attacks modelling:** This section focuses on the integration of the selected attacks (51% and Sybil) with the theoretical B-RAN modelling presented earlier. It emphasizes the security of B-RAN and evaluates it by analysing the probability of a successful attack, highlighting the equilibrium between performance and security. The section includes a performance evaluation of B-RAN under the 51% attack and the Sybil attack, providing closed-form expressions for the probability of successful attacks. It then presents numerical results collected based on the alternate history attack model, discussing the interactions among various B-RAN parameters and comparing the achieved security with other works to provide design guidelines for robust protection.
- **Section 6 – Intelligent Optimisation**: This section explores the use of AI-enhanced algorithms for intelligent optimization in blockchains, aiming to improve performance, security, and scalability. It discusses how AI and machine learning can help solve issues like low transaction processing speed, energy efficiency, security, and network congestion by enhancing consensus mechanisms and resource utilization. The section presents a RL framework designed to optimize the performance of a B-RAN by dynamically adjusting blockchain parameters to maintain target latency while optimizing resource usage. Finally, it presents the performance of this RL framework

through simulations, demonstrating its effectiveness in mitigating network congestion and restoring performance under dynamic traffic conditions.

- **Section 7– Conclusions**: This section provides a summary of the key findings and insights derived from the theoretical modelling, attack analysis, and intelligent optimization strategies discussed throughout the deliverable. It will likely reiterate the significance of the research in the context of secure Beyond 5G Long Term Evolution networks and potentially outline directions for future work within the NANCY project.

2. The role of B-RAN modelling in NANCY

Recent technological developments and the vision for the next generation of wireless communications have brought to the forefront the need to provide intelligence, energy efficiency, and security at the far edge of the network. As a consequence, radio access networks (RANs) need to be transformed to enable flexible, efficient, and reliable connectivity to a wide variety of devices [1]. High security and privacy assurances must be implemented as RAN technologies develop to reduce newly emerging risks and vulnerabilities [2]. This observation has motivated a great amount of research effort that aims to identify and prevent security issues in the RAN have been intensified [3, 4]. Potential threats to the confidentiality and integrity of communications include illegal access, data breaches, and network outages. The incorporation of blockchain into RANs presents a promising countermeasure to the above risks [5]. Although blockchain had initially developed for cryptocurrency [6], its effectiveness in decentralized tamper-proof solutions was been extensively validated [7, 8, 9]. Through decentralization, the blockchain ensures that no single entity controls the network; thus, mitigating the risks associated with centralized points of failure and unauthorized access [10].

The investigation of how blockchain can enhance security across various network areas has been widely explored making it a focal point in the evolution towards the sixth generation (6G) wireless systems. Additionally, the Markovian chain technique is a versatile and powerful method that can be applied across various fields, such as marketing [11], weather forecasting [12], and chemical engineering [13]. An important example of its application is in blockchain technology, where it has proven effective, as demonstrated in previous studies [14, 15]. More specifically, Markovian chains have been applied in blockchain within RAN scenarios, as seen in research studies [16, 17]. Based on these examples, we conclude that Markov chain models are appropriate due to their proven effectiveness in network and blockchain performance analysis. For example, the authors of [18] have conducted a comprehensive examination of the integration of blockchain into RANs, proposing a framework of secure B-RAN tailored for 6G networking. Additionally, they have outlined a framework for the analysis of block-structured Markov processes, adding phase-type service periods and transaction arrivals to the existing models. In [19, 20], the authors have taken advantage of Markov chain (MC) models to investigate B-RAN systems performance in terms of latency and security capabilities. A similar modelling approach has been followed in [21], where the authors have presented a dual-hop B-RAN architecture and have analysed its performance in terms of probability of delay and average latency.

Several recent studies have focused on the specification of the ideal block size [22, 23, 24, 25]. In particular, in [22], the authors have described the building and mining process with a focus on performance assessment, while the authors of [23] have studied the latency model of blockchain with a variety of timers and forks. The authors of [24] have presented block access control as a remedy to blockchain forking problems in wireless networks. This method effectively controls block transfer and improves transaction

throughput. In the same work, an evaluation of the network performance in terms of transaction throughput and saving computational power has been conducted. Batch service queuing has been utilized in [25] in order to reduce the impact of delay on system stability.

All the aforementioned works consider very constrained blockchain models that assume singular transaction per block and do not consider the possibility of block rejections. To cover this gap, this deliverable introduces a novel B-RAN architecture and attacks modelling that allows both individual and commercial intermediary nodes to act as wireless access providers, regardless of their ownership. At the same time, the reduced complexity of the proposed model constitutes a versatile tool for conducting performance assessment and extracting design guidelines before actually deploying a B-RAN system.

2.1. Relation to the NANCY usage scenarios

B-RAN enables collaboration among different service providers and users in a secure, private, and dependable manner. This is achieved by blending blockchain with virtualization, multiple-access edge computing (MEC), and AI functionalities. B-RAN opens the door to a number of attractive usage scenarios, such as fronthaul network of fixed topology, advanced coverage expansion, and advanced connectivity of mobile nodes. In the rest of the section, we document the aforementioned scenarios.

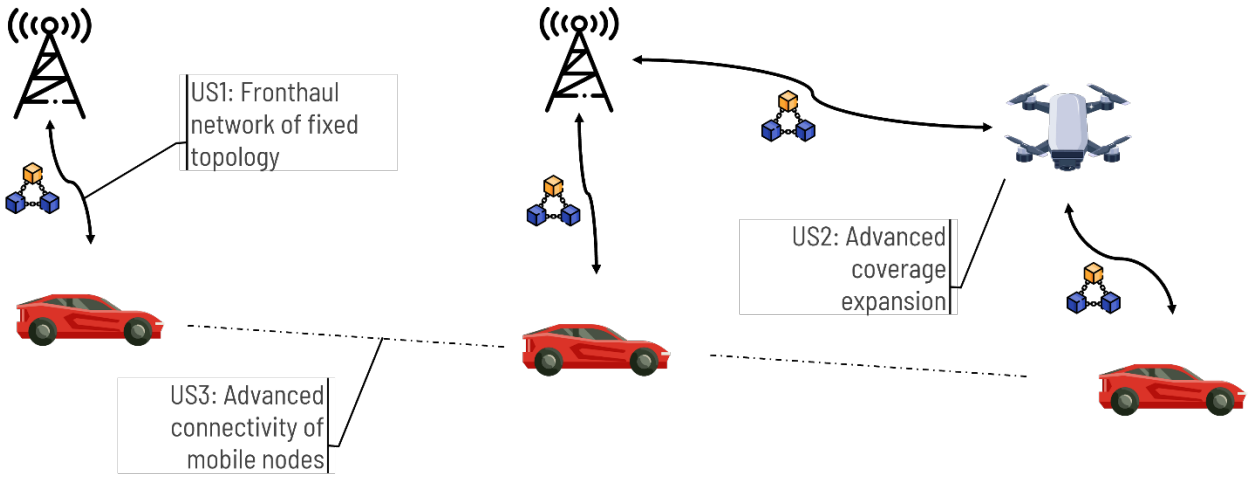


Figure 1. Network topology

2.1.1. Fronthaul network of fixed topology

In the fronthaul network of fixed topology, each user equipment (UE) performs tasks that demand significant computing power and time sensitivity, such as navigation, video streaming, or virtual reality. It is assumed that the base stations (BSs), which may belong to different service providers, are equipped with MEC capabilities. BSs have computing resources and can carry out AI tasks. UEs with resources can offload tasks to various edge infrastructures using resource allocation strategies, such as those optimized

for cooperative transmission in C-RAN environments [26]. Additionally, coordinated multi-point (CoMP) connectivity, managed through joint precoding and resource allocation strategies, can improve system reliability and energy efficiency in scenarios involving multiple BSs [27]. The need for B-RAN stems from the absence of trust-based interactions between UEs and BSs.

2.1.2. Advanced coverage expansion

The concept of advanced coverage expansion plays a significant role in future communication networks in order to address the increasing demand for reliable high-speed connectivity in various environments. It involves utilizing state-of-the-art approaches, like using infrastructure as relay nodes [28], implementing node structures, and employing efficient connectivity models to enhance network performance [29]. These methods not only ensure expanded and higher quality network coverage but also significantly improve energy efficiency. B-RAN is envisioned to augment security in coverage expansion scenarios through the integration of blockchain to strengthen confidentiality and trust in communications. By incorporating these elements, B-RAN is capable to meet the complex and evolving needs of modern connectivity, while also emphasizing the importance of adaptability, security, and efficiency in network expansion.

2.1.3. Advanced connectivity of mobile nodes

This scenario promotes communication between vehicles and BSs as well as between different vehicles. Information can be shared among vehicles, where one vehicle acts as the intermediate node to forward the data to the BS. Establishing a network that supports both vehicle-to-vehicle (V2V) [30] and vehicle-to-BS communication is necessary [31, 32]. In addition, an important goal is to ensure wide coverage and efficiency in terms of latency, throughput, and energy consumption. Although vehicular communications rely on ground-based infrastructure for V2V transmission, the growing demands of services call for more stringent requirements that these traditional methods cannot fulfil. For instance, when vehicles travel far from the BS, ensuring latency for real-time applications requires synchronization among vehicles and becomes challenging if communication must go through the BS. V2V communications address this by enabling direct information exchange without relying on centralized infrastructure. While eliminating the reliance on the BS can offer advantages, it also comes with drawbacks such as the lack of a centralized entity responsible for network security management. B-RAN aims to ensure security and privacy by using pseudonyms when sharing data since trust cannot be assumed. Moreover, B-RAN can enable support for multi-hop communications among vehicles to minimize connectivity gaps and extend the coverage of the network.

2.2. Network Topology

The realization of the aforementioned usage scenarios is founded upon not only point-to-point (P2P) but also multi-hop connectivity. If we take for instance a mobile connectivity use case of a vehicle moving

inside the coverage area of the network, p2p connectivity can suffice for providing network services to the UE. However, when the vehicle reaches the limits of the network's coverage area, P2P links can no longer provide adequate quality for the service. In this case, an ad-hoc network must be instantiated by a node that is located close to the UE can extend the network coverage and providing connectivity. This intermediate node is connected at the same time to the mobile UE and the BS; thus, providing the service though multi-hop connectivity. This high-mobility use case is depicted in Figure 1, which illustrates the different B-RAN usage scenarios as a vehicle moves through the network and eventually exits its coverage to be served by the intermediate node. Specifically, the fronthaul network of fixed topology usage scenario is applicable to direct connectivity cases inside the coverage area of the network, the advanced connectivity of mobile nodes describes the entire movement of the vehicle, and the advanced coverage expansion scenario is applicable after the vehicle moves beyond the limits of the fixed infrastructure and an ad-hoc network is deployed by the intermediate node.

In this deliverable, we model the B-RAN network dynamics through Markov-chain theory. To achieve this, we split the theoretical modelling into two network topologies. The first assumes a direct connectivity scenario with the UE connected directly to the BS; thus, providing the service through the primary blockchain of the network. Of note, there are three main blockchain architectures, i.e., public, private and consortium, that can be used in an implementation [33]. Each blockchain type has different characteristics, specifications, and requirements; however, the flexible and adaptable nature of the presented model makes it applicable in all possible scenarios regardless of the blockchain type. The second topology tackles the multi-hop connectivity case of the advanced coverage expansion scenario; thus requires the establishment of a hierarchical B-RAN architecture that deploys a secondary blockchain between the intermediate node and the UE to provide the service outside of the network's primary coverage area.

Alongside the blockchain architecture, it is essential to mention the consensus mechanisms. According to relevant studies [34, 35] the consensus algorithms can be categorized into a variant of types, with the most used to be Proof of Work (PoW), Proof of Stake (PoS) and Byzantine Fault Tolerance (BFT) each of them with unique attributes. For example, a characteristic feature of PoW is the fact that is resource intensive, as it requires nodes to solve difficult problems, a feature that offers a strong security level but is highly demanding on energy consumption. On the other hand, PoS completes its role based on the validator's stakes, thus consuming less energy. Finally, the BFT algorithm is ideal for distributed models where fault tolerance has an important role in the overall model. In conclusion, we can distinguish that each consensus mechanism has its strengths and weaknesses by trying to satisfy both the overall speed of the model and its security. The ideal choice depends on the blockchain model and the attributes (e.g., security, latency times, energy consumption) that would like to address the most in every scenario. The proposed framework can model different deployments of blockchain through the employed Markov-chain theory-based approach that depends on the probabilistic behaviour of the blockchain.

3. B-RAN Modelling

In this section, we present the B-RAN model. We employ a Markov chain model to capture the probabilistic transitions between different system states and delve into their dynamic behaviour during the operations of B-RAN. Additionally, we introduce an extension of the single-chain B-RAN model, which is characterized by nested blockchains that create a HB-RAN architecture. This HB-RAN model provides a novel solution for assessing the performance of coverage expansion scenarios, like ad-hoc deployments and cell-free network access in terms of security and reliability. Through this approach, our aim is to illustrate the operational dynamics of B-RAN, while offering essential insights for its optimization and improvement.

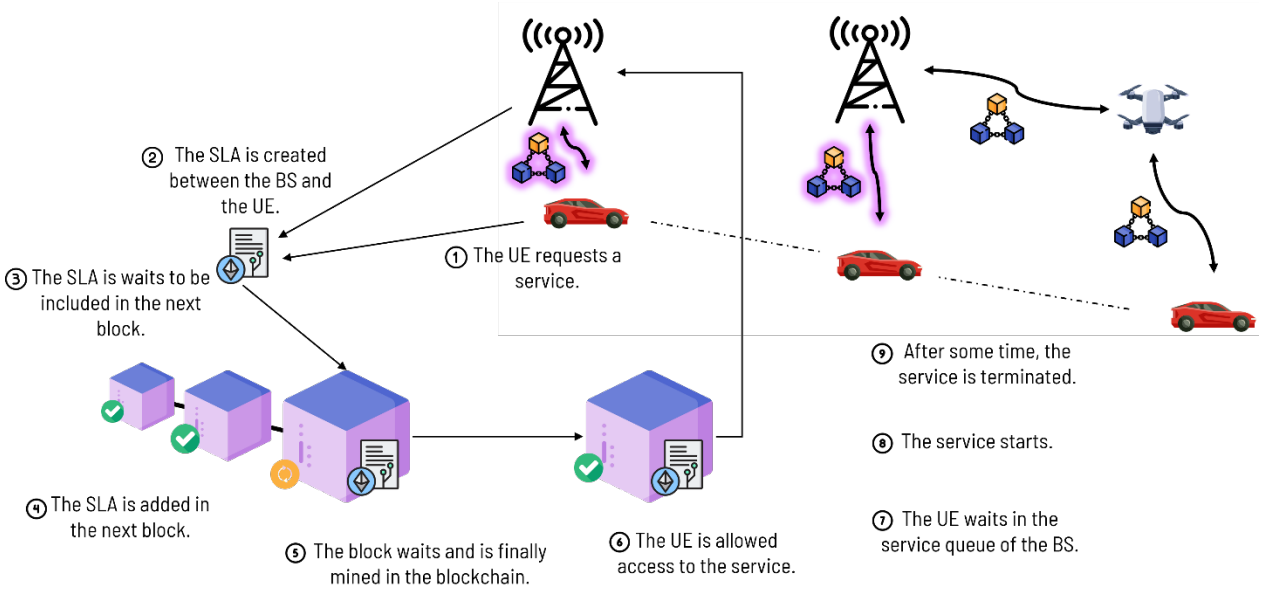


Figure 2. B-RAN architecture

The B-RAN model is depicted in Figure 2 and illustrates its operation through the utilization of two queues. The first queue models service requests that wait to be included in a blockchain block, while the second queue models the confirmed requests that wait to be serviced by the network. In more detail, the first queue operates based on the principles of a $M/M/1$ queue, in which requests arrive with a Poisson distribution with rate $R_a \in \mathbb{R}_+$, and their processing times are governed by memoryless exponential distributions with a rate of $R_m \in \mathbb{R}_+$. The Poisson process is ideal to model network traffic since the aggregation of multiple i.i.d. processes tend to a Poisson process for a sufficient number of events. Also, its simplicity and mathematical tractability enable efficient simulation and analysis, making it straightforward to generate and evaluate traffic patterns. In addition, the superposition and splitting properties allow complex network scenarios-such as multiplexed traffic or routed paths-to be modelled with minimal computational overhead. Each of the request is processed within blockchain blocks that may

contain a maximum of k number of requests per block. Additionally, the second queue is modelled as a $M/M/s$ queue, with s representing the maximum number of access links. Requests arrive based on a Poisson distribution, while their processing times are characterized by memoryless exponential distributions. Based on the above model, at any time t, the system is fully described by two non-negative integers, namely i and j. The former denotes the number of requests currently pending inclusion in the next blockchain block. The latter specifies the number of (already mined) requests sitting in the service-ready queue. The $i-j$ pairs are packaged into one "state," termed $E[i,j] \in S = \{(i,j) \mid i = 0,1,2,\dots; j = 0,1,2,\dots\}$. For example, in the case that the servicing stage has only s servers but infinite buffer, $j \in [0,\infty)$ with up to s "in service" simultaneously. Therefore, $E[i,j]$ is a 2-dimensional integer vector in the product state-space of the two queues. Every state $E[i,j] \in \mathbb{N}_0 \times \mathbb{N}_0$ fully specifies how many jobs are "mining-pending" (i) and how many are "service-pending" (j).

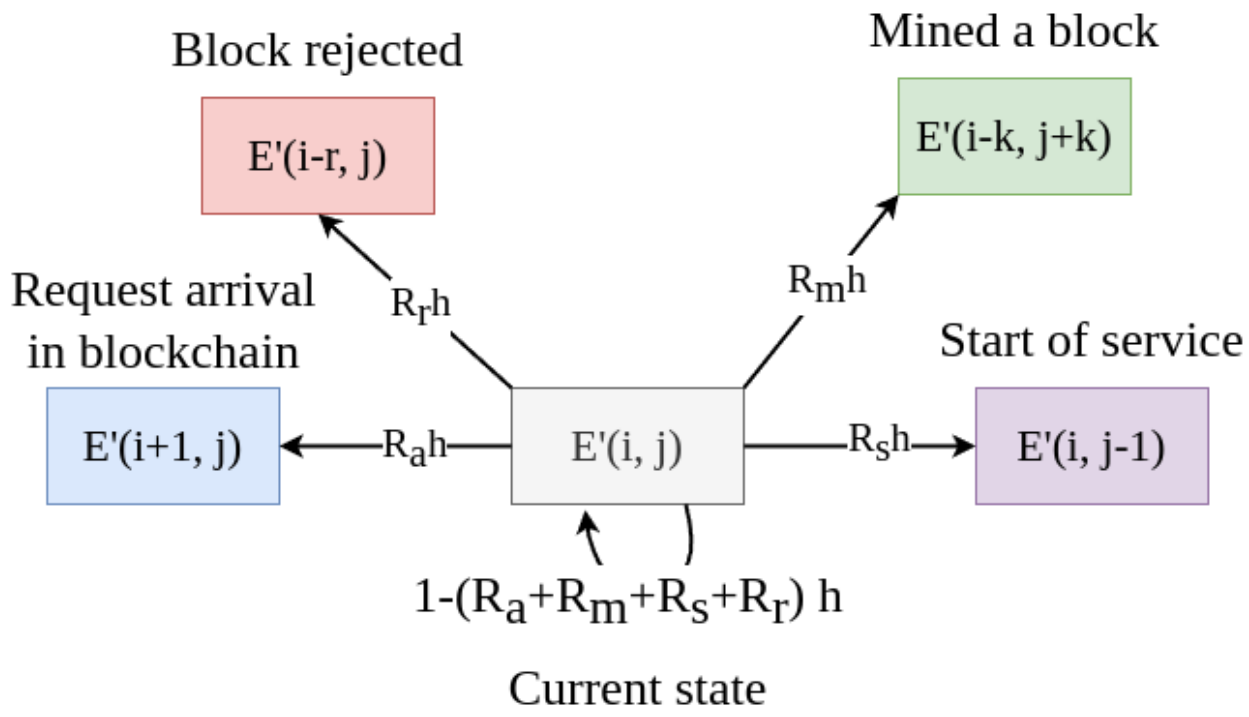


Figure 3. B-RAN Markov chain model

3.1 Markov Chain Model

The possible states can be aptly portrayed as a continuous time-homogeneous Markov process; thus, embodying all the defining characteristics inherent to a Markov chain [20]. As depicted in Figure 3, the Markov chain model is defined by its current state, $E[i,j]$, at time t, and five discrete states that capture various configurations of the system. The transitions between these states take place over minimal time interval $h \to 0$.

When a new request is received, the next state is denoted by $E' \rightarrow [i + 1, j]$ and signifies an increase in the number of pending requests for blockchain, i, as an additional request is added to the next block. Of note, in this case, the number of requests waiting service, j, remains unchangeable. This reflects the fact that only one event can take place at any given h. The probability of this transition is defined as

$$p_a = R_a h, \tag{1}$$

where R_a stands for the rate at which an arrival request occurs. Next, the transition to $E' \rightarrow [i - k, j + k]$ describes the case in which a block is successfully mined. The probability of successful block mining can be expressed as

$$p_m = R_m h, \tag{2}$$

with R_m being the mining rate of a block. Note that p_m depends on k, which denotes the maximum number of requests that can be included in a single block. A successful block mining event is related on the number of pending requests and the block size number. If the number of pending requests, i, is equal to or less than the threshold k, then all pending requests are successfully mined in a single block; thus, increasing j by i; the subsequent state is denoted by $E' \rightarrow [0, j + i]$. Conversely, if the number of pending requests surpasses the threshold k, only a maximum of k requests can be mined, while the remaining requests remain pending; in this case, the next state can be written as $E' \rightarrow [0, j + k]$. The transition to the $E' \rightarrow [0, j - 1]$ state models the service of a request and is associated with a probability

$$p_s = R_s h, \tag{3}$$

where R_s is the service rate. This transition indicates a reduction of 1 in the j queue, signifying the commencement of service for the corresponding request. Note that the number of pending requests, i, remains unaffected, as service initiation influences only the queue of blocks and not the pending requests. Additionally, the transition to state $E' \rightarrow [i - r, j]$ characterizes the rejection of a request due to factors like authentication problems, insufficient resources, and so on. In this case, r represents the number of rejected requests. This transition is governed by the rejection probability, which is defined as

$$p_r = R_r h, \tag{4}$$

with R_r standing for the rejection rate. When a rejection event occurs, i decreases by r since the block that contains the rejected request is discarded and the remaining requests need to be included in the next block. Meanwhile, j remains unchanged emphasizing that the rejected block did not advance to the mining stage. Finally, there is a probability that none of the aforementioned events occur; this signifies the idle state. The idle state is denoted by $E' \rightarrow [i, j]$ with its probability being written as

$$p_i = 1 - (p_a + p_m + p_s + p_r), \tag{5}$$

or, after applying (1)–(4),

$$p_i = 1 - (R_a + R_m + R_s + R_r)h. \quad (6)$$

The idle state represents that the system remains unchanged at the given time without moving to any available states. The probability of this scenario captures the possibility of no requests coming in, no requests being rejected, no mining successes, and no service operations being completed.

3.2 Hierarchical B-RAN Model

In order to support the scenarios, which were documented in Section 2, HB-RAN deployments are required. In the scenario of advanced coverage expansion, the end user is unable to directly connect to the BS of its Internet service provider. However, it can establish a direct link with an intermediate user (IU) that is already connected to the BS via the primary blockchain. Therefore, a secondary blockchain is created between the intermediate and the end user to ensure security, privacy, and trust, as seen in Figure 4.

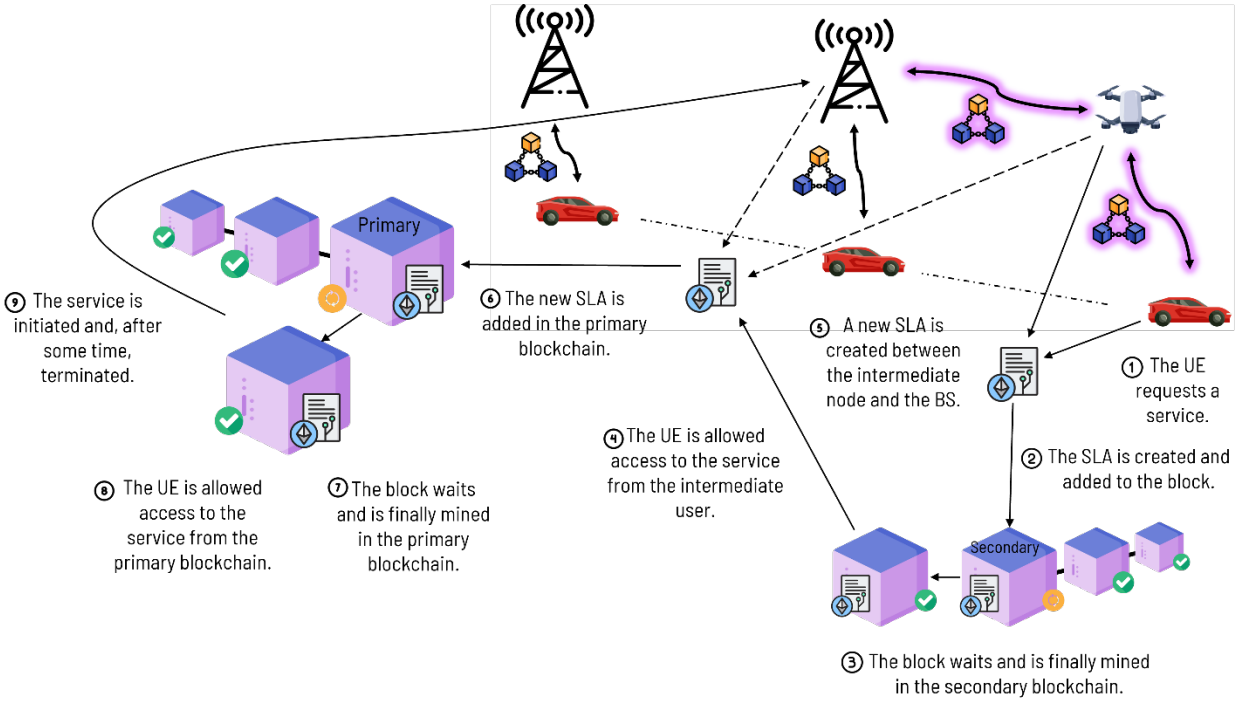


Figure 4. Hierarchical blockchain architecture

This procedure generates a smart contract between the end user and the IU. Consequently, an end-user request from the secondary network, as illustrated in Step 1 of the figure, is initiated to establish a connection and create a Service Level Agreement (SLA) in Step 2, after which it is inserted into the secondary blockchain. At first, this request enters a $M/M/1$ queue and waits to be included in a block; this process is depicted in Step 2 of the figure. Next, the mining phase begins by the blockchain network in order to verify the request. After a successful mining process, the request is forwarded to a second queue, where blocks are waiting to be serviced using a multiple-server queuing model $M/M/s$. Once the request

is validated through the secondary blockchain (Step 4), a corresponding request is formed in the primary blockchain. Upon accessing the primary blockchain by a new SLA thus has been created from the secondary blockchain, as depicted in step 5, it joins the $M/M/1$ queue of the primary blockchain (Step 6) for block inclusion and then waits for N confirmations to validate the block, according to step 7 of the figure. Afterwards, it transitions into a distinct stage that waits in the $M/M/s$ queue of the primary blockchain to begin its service (Step 8). Once the request's service starts on the main blockchain, it switches to the secondary blockchain, and its end-toend (e2e) latency can be measured, reflecting the initiation of service as depicted in Step 9. This refers to the total time spent navigating across both primary and secondary blockchains.

3.3 Performance Evaluation

In this subsection we analyse the performance of the BRAN framework, concentrating on assessing the vulnerabilities and estimating the system latency. We provide important insights into the fundamental concepts of B-RAN by mathematically modelling latency.

The proposed framework utilizes two queues to model the complex dynamics of incoming requests and their processing in blockchain blocks. As explained earlier, a $M/M/s$ queue simulates the latency caused by service initiation and processing, while a $M/M/1$ queue handles requests that are waiting to be included in the blockchain. The end-to-end latency of the system is a result of both queues. The expected value of the waiting time due to the $M/M/1$ queue in the B-RAN model can be analytically expressed as in [36]

$$\tau_1 = \frac{1}{R_m - R_a}, \tag{7}$$

where R_a stands for arrival rate and R_m for service rate. Moreover, the latency generated by the $M/M/s$ queue can be written as in [37]

$$\tau_2 = \frac{C\left(s, \frac{R_a}{R_s}\right)}{sR_s - R_a} + \frac{1}{R_s}, \tag{8}$$

with the first term's nominator expressing the Erlang C formula, which depends on s, R_a, and R_m. Furthermore, the confirmation process of the blockchain also creates some additional delay that can be calculated as

$$\tau_3 = \frac{N-1}{R_m}, \tag{9}$$

where N denotes the number of confirmations and Rm represents the block generation rate. At this point, Little's Law has been applied to establish a relationship between the expected latency and the queue

length. Little's Law asserts that the arrival rate multiplied by the average time an item spends in the system equals the average number of transactions in a stable system. Consequently, the expected sojourn time, τ_s, which quantifies how long each service request remains within its specific system state, can be expressed as

$$\tau_s = \tau_1 + \tau_2 + \tau_3. \tag{10}$$

As a result, the average latency of B-RAN, τ_t, can be evaluated as

$$\tau_t = \tau_s - \frac{1}{R_s}. \tag{11}$$

It is important to highlight that, in the HB-RAN model, the same process is followed for the evaluating the latency. This method calculates the time it takes for a request to be served by the primary blockchain. However, in the coverage expansion scenario, the end-to-end latency is measured by combining the delays incurred by both the primary and secondary blockchains. Based on the aforementioned, we now investigate the BRAN service latency for the single confirmation scenario. In this scenario, the current state is expressed as $E(i,j)$ with i and j denoting the pending requests awaiting assembly into a block and the confirmed requests ready for service, respectively. Let $P_{i,j}(t) = PX(t) = E(i,j)$ denote the probability of the queue being in state $E(i,j)$ at time t. Additionally, we assume that the transition probability P characterizes the queuing model. All transition probabilities are zero except for events of arrivals, mined blocks, rejected blocks, or start of service. The nonzero probabilities of the system are given by

$$\mathrm{P}\{E(i,j) \mid E'(i,j)\} = \frac{R_a}{R_a + R_m + R_r + R_s^j}\left(R_a + R_m + R_r + R_s^j\right)h + \mathcal{O}(h^2), \tag{12}$$

where $\mathcal{O}(h^2)$ denotes higher order terms which vanish sufficiently faster than h. In case a new request arrives in the system, its probability is given by

$$\lim_{h\to 0}\mathrm{P}\{E(i,j) \mid E'(i+1,j)\} = R_a h + \mathcal{O}(h^2). \tag{13}$$

Additionally, when a block gets mined, the probability of this event can be written as

$$\lim_{h\to 0}\mathrm{P}\{E(i,j) \mid E'(i-k,j+k)\} = R_m h + \mathcal{O}(h^2). \tag{14}$$

In case a contract gets rejected from the block, its probability can be expressed as

$$\lim_{h\to 0}\mathrm{P}\{E(i,j) \mid E'(i-r,j)\} = R_r h + \mathcal{O}(h^2). \tag{15}$$

In case a contract gets serviced, the probability of this event is given by

$$\lim_{h\to 0}\mathrm{P}\{E(i,j) \mid E'(i,j-1)\} = R_s^j h + \mathcal{O}(h^2), \tag{16}$$

(16)

The sum of all transition probabilities should be equal to unity, which can be expressed as

$$P_{i,j}(t+h) - P_{i,j}(t) = \left[P_{i-1,j}(t)R_a + P_{i,j+1}(t)R_s^{j+1} - P_{i,j}(t)\left(R_a + R_m + R_s^j + R_r\right)\right]h, \tag{17}$$

where R_s is the completion rate and it's defined by $R_s^j = \min(j,s)\,R_s \; for \; 0 \leq j \leq s$, since at most s can be in service at the same time. Dividing by h equation (17) and taking the limit $h \to 0$, we get

$$\lim_{h \to 0} \frac{P_{i,j}(t+h) - P_{i,j}(t)}{h} \stackrel{\text{def}}{=} \frac{dP_{\{i,j\}}(t)}{dt}, \tag{18}$$

which is the definition of derivative, and thus we obtain the steady-state distribution of B-RAN by setting (18) to zero

$$\frac{dP_{\{i,j\}}(t)}{dt} = P_{i-1,j}R_a + P_{i,j+1}R_s^{j+1} - P_{i,j}(R_a + R_m + R_s + R_r) = 0. \tag{19}$$

Specifically, in the boundary case of $(i = 0)$, (18) can be rewritten as

$$\left(\sum_{\ell=1}^{j} P_{\ell,j-\ell}\right)R_m + P_0^{j+1}R_s^{j+1} - P_{0,j}\left(R_a + R_s^j + R_r\right) = 0, \qquad \forall \mathrm{j} \geq 0, \tag{20}$$

where,

$$P_{0,1}R_s^1 - P_{0,0}R_a = 0. \tag{21}$$

The differential-difference equations (19)-(21) are the forward Kolmogorov equations [38], which can be rewritten more concisely in a probability vector given by

$$\mathbf{P} = \left[P_{0,0}\middle|P_{1,0}P_{0,1}\middle|P_{2,0}P_{1,1}P_{0,2}\cdots\right]^T, \tag{22}$$

Or in matrix notation as

$$QP = \mathbf{0}, \tag{23}$$

with Q being the infinitesimal generator or transition rate matrix. Each entry in Q equals the corresponding transition rate given by $\frac{d}{dh}\mathrm{P\,r}\,X(t) = E|X(t+h) = E'$, depending solely on the B-RAN configuration tuple $\Phi = \{R_a, R_m, R_r, R_s, s\}$. It can be numerically calculated by utilizing the sum probability condition, $1^T P = 1$ as

$$\begin{bmatrix}\mathbf{Q}\\ \mathbf{1}^T\end{bmatrix}\mathbf{P} = \begin{bmatrix}0\\1\end{bmatrix}. \tag{24}$$

The state transition relationships that can be calculated based on the presented analysis for the one confirmation case are presented in Figure 5. From (24), the steady-state distribution, $w(\Phi)$ can be analysed as an implicit function of Φ. Of note the waiting space of B-RAN has no maximum limit, resulting in infinite dimensions for the vector w. In numerical calculations, we approximate the infinite-dimension solution with sufficiently large but finite dimensions as, in practice, the number of UEs cannot be infinite. Thus, the aggressive load R_a must be less than R_s for stability.

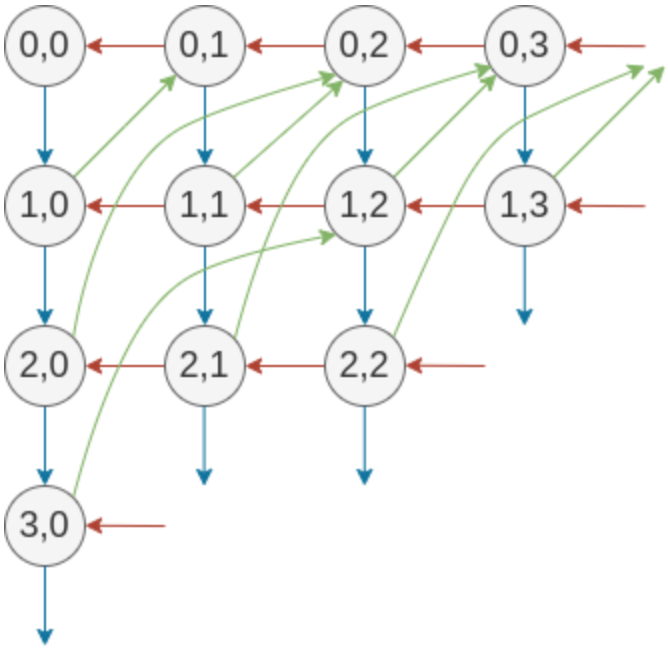


Figure 5. State space scenario for k = 2 and r = 1

In order to analyse the average latency in B-RAN, we consider the limiting distribution $w(\Phi)$ to obtain the average number of waiting requests $N(\Phi)$ as

$$\mathbb{E}N(\Phi) = \sum (i+j) \cdot P_{i,j}(\Phi). \tag{25}$$

Applying Little's Law, we can link the expected queue length and average latency as the expected sojourn time, $L_s(N, \Phi)$, includes both waiting and service latency and can be expressed as

$$\begin{aligned} \mathrm{L}_s(N=1,\Phi) &= \mathbb{E}\mathrm{N}(\Phi)/R_a \\ &= T_a \sum_{i,j} (i+j) P_{i,j}(\Phi), \end{aligned} \tag{26}$$

therefore, the average latency for the one-confirmation scenario can be expressed based on the limiting distribution in Eq. (23) as

$$\mathrm{L}(N=1,\Phi) = T_a \sum_{i,j} (i+j) P_{i,j}(\Phi) - T_s, \tag{27}$$

with T_s denoting the service time.

So far, only the one-confirmation scenario was considered. When investigating the generic N-confirmation problem, the huge number of variables in eq. (23) makes analysis of a queue with $(N+1)$ −dimensional state space difficult. However, if we consider that once a request is included into a block it must wait for $N-1$ confirmations after the initial confirmation, eq. (27) can be rewritten as

$$\begin{aligned} \mathrm{L}(N,\Phi) &= \mathrm{L}(1,\Phi) + \mathbb{E}\left\{\sum_{n=2}^{N} U_n^m\right\} \\ &= T_a \sum_{i,j} (i+j) P_{i,j}(\Phi) + T_m(N-1) - T_s \end{aligned}. \tag{28}$$

Based on (28), the transition rate matrix, Q, becomes an infinitely dimensional structure that encapsulating all possible rates for every state and is presented in Figure 6.

$$Q = \left[\begin{array}{c|cc|ccc|ccc|ccc} -R_a & R_r & R_s & & & & & & & & & \\ \hline R_a & -(R_a+R_m+R_r) & & R_r & R_s & & & & & & & \\ & R_m & -(R_a+R_s) & & R_r & R_s & & & & & & \\ \hline & R_a & & -(R_a+R_m+R_r) & & & R_r & R_s & & & & \\ & & R_a & & -(R_a+R_m+R_r+R_s) & & & R_r & R_s & & & \\ & & & R_m & R_m & -(R_a+R_s) & & & R_r & R_s & & \\ \hline & & & R_a & & & -(R_a+R_m+R_r) & & & & & \\ & & & & R_a & & & -(R_a+R_m+R_r+R_s) & & & & \\ & & & & & R_a & R_m & & -(R_a+R_m+R_r+R_s) & & & \\ & & & & & & & R_m & R_m & -(R_a+R_s) & & \end{array}\right]$$

Figure 6. Transition rate matrix

Q visualizes the rate at which changes between states occur in a stochastic process. Each element of the matrix represents the rate of transitioning from one state to another. The off-diagonal elements represent the rates of shifting between separate states, while the diagonal elements express the rate of the idle state and ensure that the sum of all elements in a row equals to zero. This matrix is crucial for the calculation of the latency since it determines the system's behaviour throughout the stochastic system. If we take the initial state $E[0,0]$ for example, which denotes no pending requests to be mined and no requests awaiting servicing, according to Figure 6, we have two possible transitions. Either the system stays idle, or a new request arrives with a rate of R_a. Representing the absence of further transitions, the rate for the system to remain inactive is calculated as the negative of the arrival rate R_a. As we get into more complicated stages, the number of possible outcomes grows. For each state, their future possible states can be identified through their corresponding transition rates. For instance, from state $E[0,1]$, there are three possible future states, specifically:

1) A service is completed at a rate of R_s and the state changes into the state $E'[0,0]$ as the number of requests decreases by one.
2) A new request arrives at a rate of R_a, transitioning to state $E'[1,1]$.
3) The system remains idle, with this rate equal to the negative sum of the arrival and service rates $-(Ra + Rs)$.

All in all, by analysing the transition rates in the Q matrix, we can accurately predict all the possible future states from any given state.

3.4 Numerical Results

In this subsection, we present numerical results obtained from the proposed B-RAN model alongside interesting discussions that assess its performance and highlight valuable design guidelines. It is worth noting that our simulation scenarios required substantial process power to manage the complex calculations. The following plots have produced results based on multiple simulations and events (106) to ensure the statistical validity and reliability of the results. To achieve this, we utilized a high-performance computing environment in order to execute our multiple scenarios properly and extract our results. Below is listed the hardware that we have used to fulfil our computational needs: CPU: AMD Ryzen 5 7600X 6-Core Processor 4.70 GHz, GPU: NVIDIA RTX 4070 Ti, RAM: 32 GB, OS: Linux Ubuntu 24.01. By utilizing multiple types of carefully constructed figures, we provide a visual representation of the latency of the model and contrast it with traditional models. These results highlight how closely the BRAN model captures latency while also offering insights into its flexibility and scalability in realistic scenarios with different topologies. Furthermore, we investigate the security features of the proposed B-RAN model and demonstrate its robustness against possible attacks. The presented results serve as a prism that highlights the complexities of B-RAN and helps to derive definitive conclusions about its effectiveness and robustness in realistic usage scenarios.

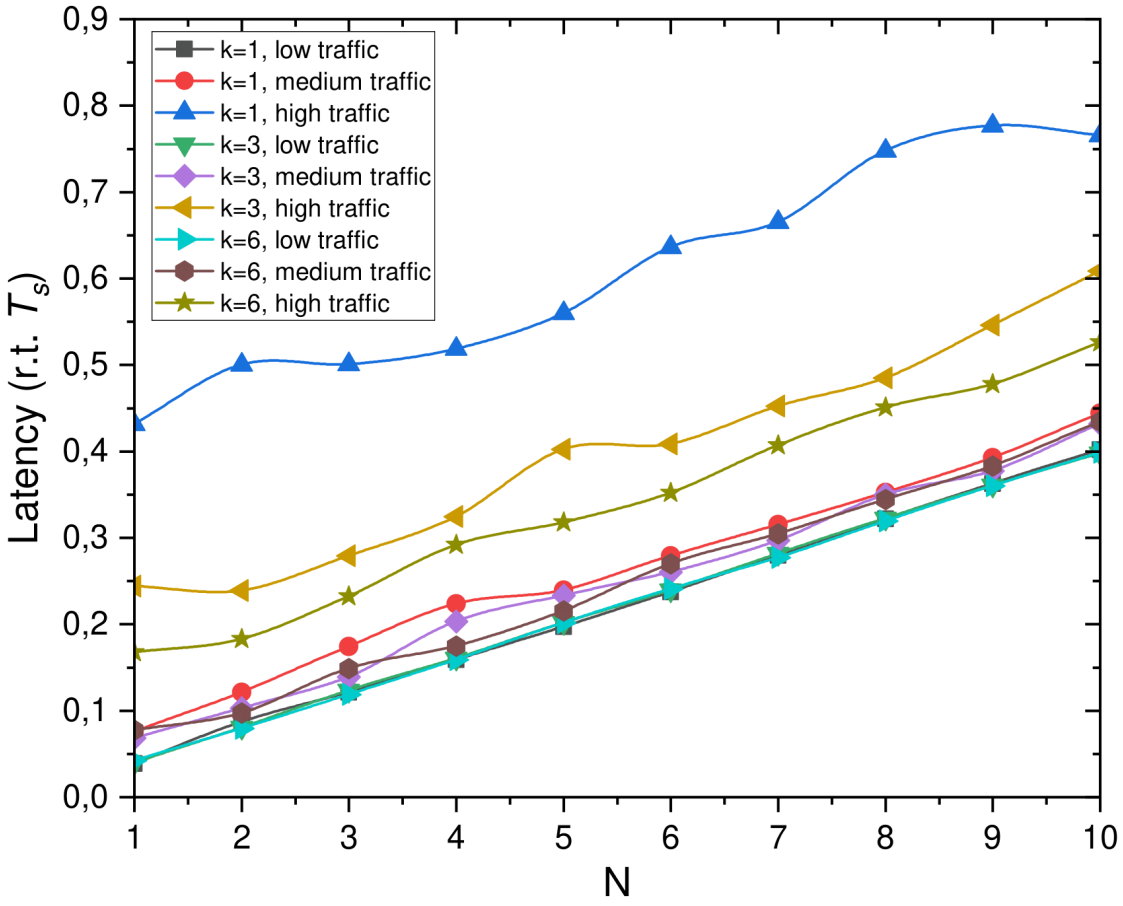


Figure 7. Latency vs N for multiple k and ρ combinations

Figure 7 presents the system's latency as a function of the number of confirmations, N, for different combinations of k and ρ. For all plotted lines, we observe that, as the number of confirmations increases, the achievable latency also increases. Moreover, it becomes obvious that the scenarios with higher k values exhibit better temporal performance. Specifically, for the high traffic regime, the model with $k = 6$ has the lowest latency, while the conventional model with $k = 1$ has the worst. Finally, it is worth noting that in the low-intensity case, all scenarios achieve similar performance with no noticeable changes. This suggests that higher block capacity may appear insignificant when the system performs under reasonable or low traffic but can have a higher impact on latency in high traffic cases.

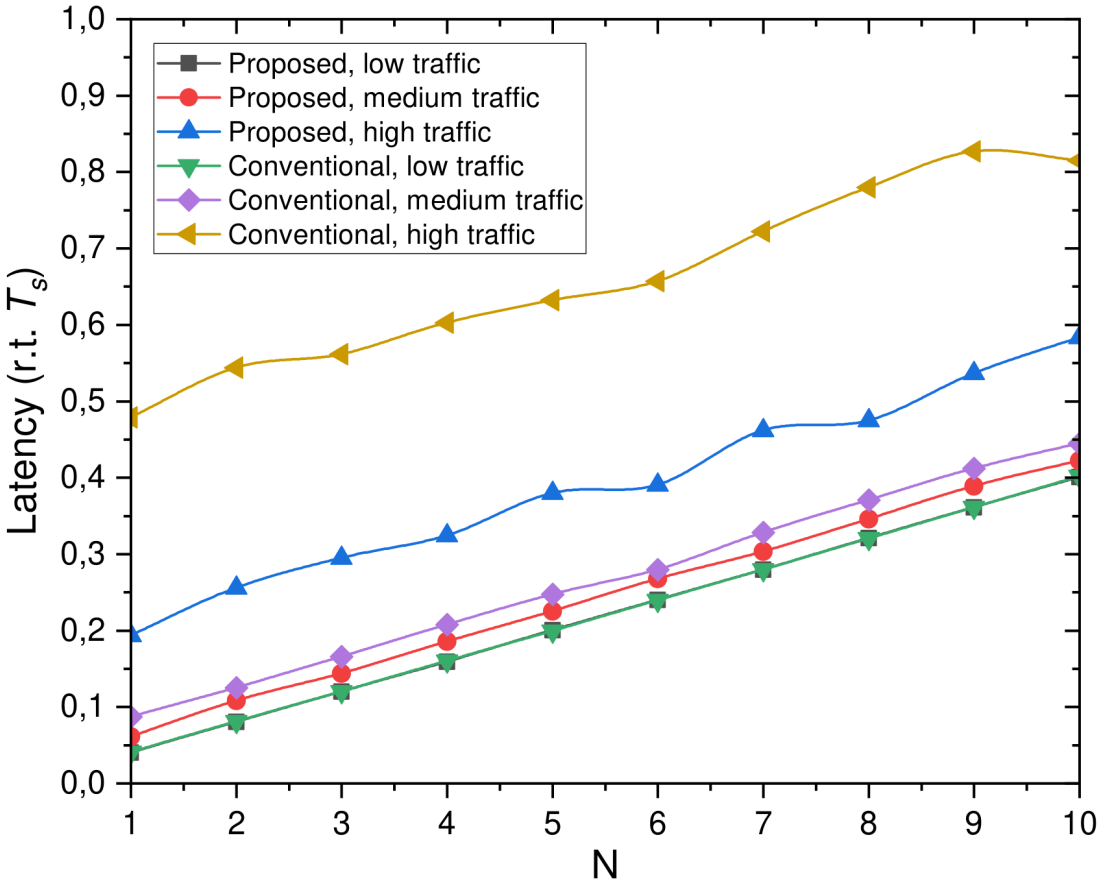


Figure 8. Latency vs N in single blockchain

Figure 8 presents a comparison between the conventional model and the proposed framework with regard to the achievable latency under different traffic intensity scenarios and different N values. By observing this figure, it becomes evident that as the number of confirmations for mining a block increases, the latency increases as well. However, a deeper look reveals some significant differences between the proposed and the traditional models. Despite achieving similar performance in low-traffic scenarios, the traditional model is characterized by higher delay under medium and heavy traffic conditions. This highlights that the proposed framework is more capable of modelling the temporal performance of complex B-RAN systems under a plethora of traffic conditions.

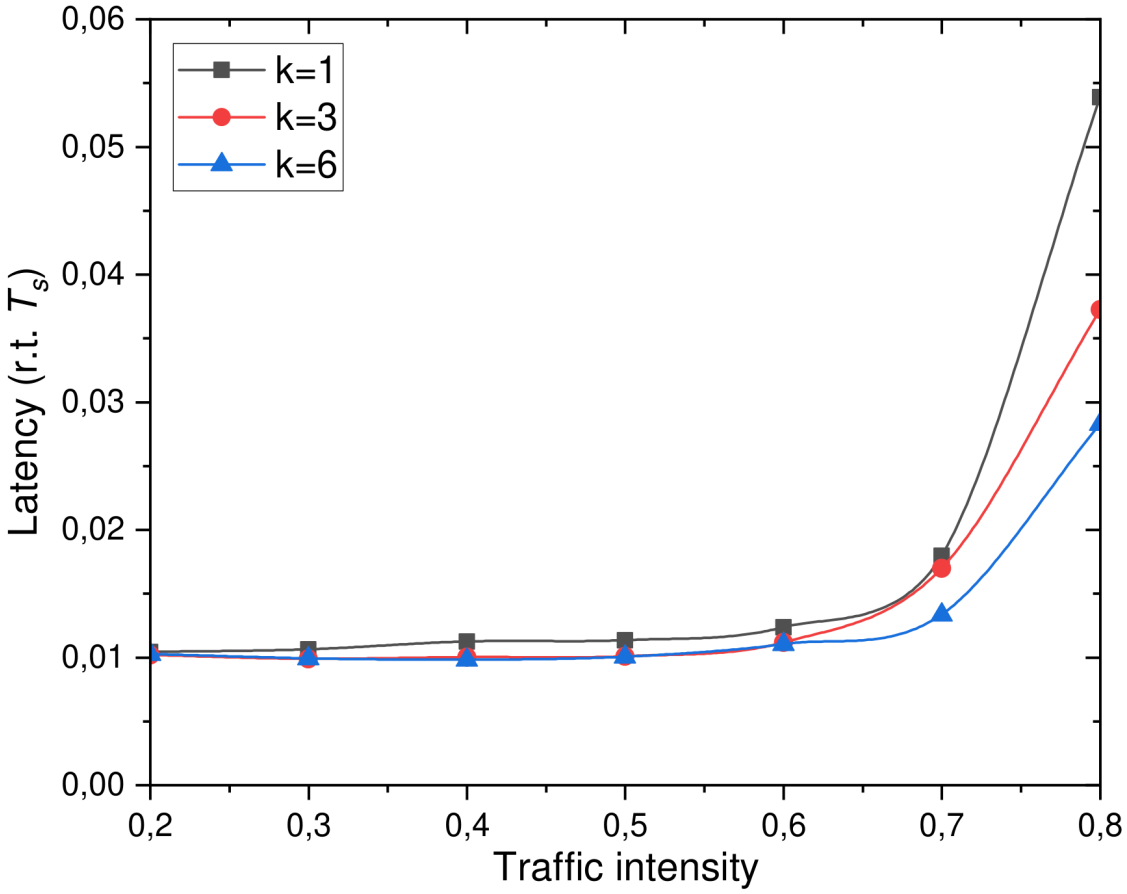


Figure 9. Latency vs traffic intensity for single blockchain for different k values

Figure 9 illustrates the latency as a function of the traffic intensity for different values of k in the single blockchain scenario. From this figure, it becomes evident that the achievable latency for the low traffic regime achieves very close latency independent of the value of k. Moreover, until the threshold of $\rho = 0.5$, all cases exhibit the same behaviour. On the contrary, when traffic increases past this point, it becomes evident that the lines diverge from each other, with $k = 1$ increasing significantly higher latency than the other two cases. For traffic intensity equal to 0.8, we observe the biggest difference between the values. Specifically, the $k = 1$ line achieves the highest latency, while $k = 6$ has the lowest. This suggests that by increasing the number k we achieve significantly lower latency especially in high traffic scenarios. In essence, as long as the intensity remains below a certain threshold, the number of possible transactions per block has no major impact on the system's latency. As the intensity increases, larger values of k allow the system to reach its true potential.

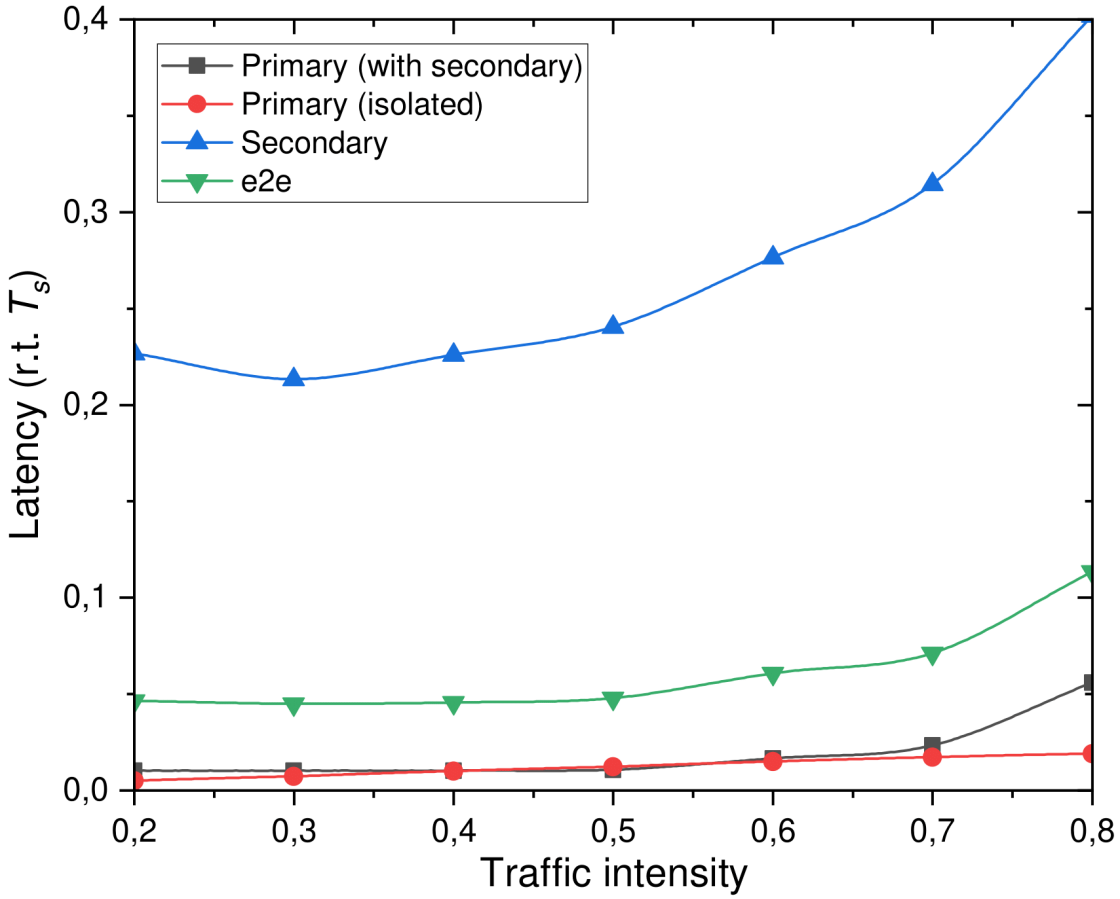
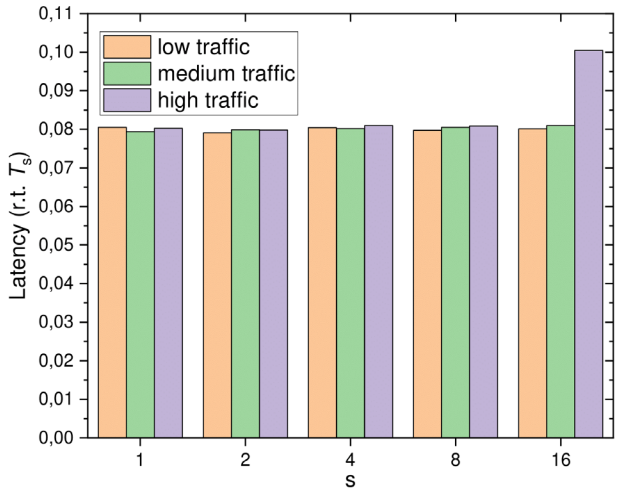


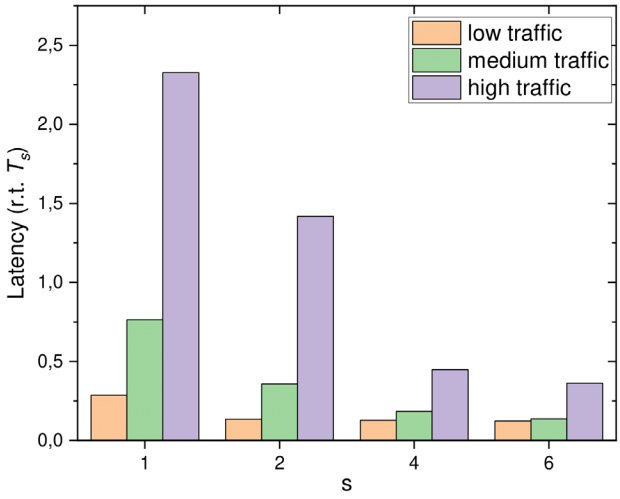
Figure 10. Latency vs traffic intensity of the HB-RAN

A thorough study of traffic intensity impact on latency is presented in Figure 10 which provides interesting insights on the average total delay of the system as well as the interactions of the primary and secondary blockchains. Specifically, the primary blockchain is analysed both in conjunction with the secondary blockchain and on its own; thus, offering a deeper understanding of the performance degradation that is introduced by the hierarchical approach. Moreover, the various traffic intensity values were exclusively applied to the secondary blockchain in order to keep the primary blockchain's characteristics stable. This figure reveals a coherent trajectory, in which both blockchains and the average total delay exhibit a correlated increase in latency as the traffic intensity increases. Nonetheless, some subtle differences that characterize how primary and secondary blockchains behave differently from one another can be extracted. First and foremost, the secondary blockchain is characterized by significantly higher latency compared to the primary. This is a consequence of the fact that the primary chain has more computing capacity as it is the one connected with the BS, responsible for the direct connection between the users and the station, while the secondary has the responsibility to expand the network and connect the users with the primary. In addition, as traffic intensity increases, the secondary blockchain experiences a steeper

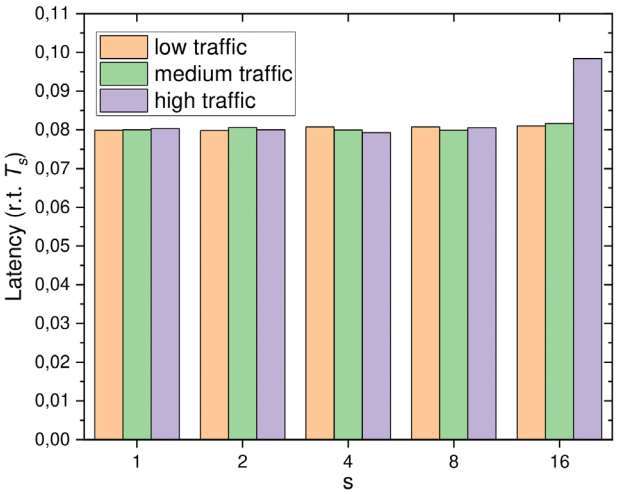
increase in delay, while the latency of the primary blockchain maintains a relatively low latency. Moreover, the average total delay shows a larger dependency on primary blockchain performance, with a trajectory that is closer to the performance of the primary blockchain. Also, the plot highlights the efficiency of an isolated primary blockchain, devoid of secondary blockchain influence. In conclusion, the plot clarifies how variations in network traffic can affect temporal dynamics and the system's overall performance, offering important insights into the complex relationship between the primary and secondary blockchains.



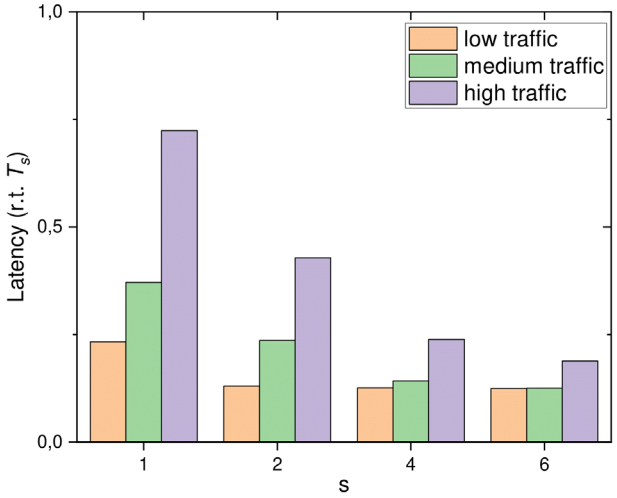
(a) Latency (primary blockchain) for k = 1



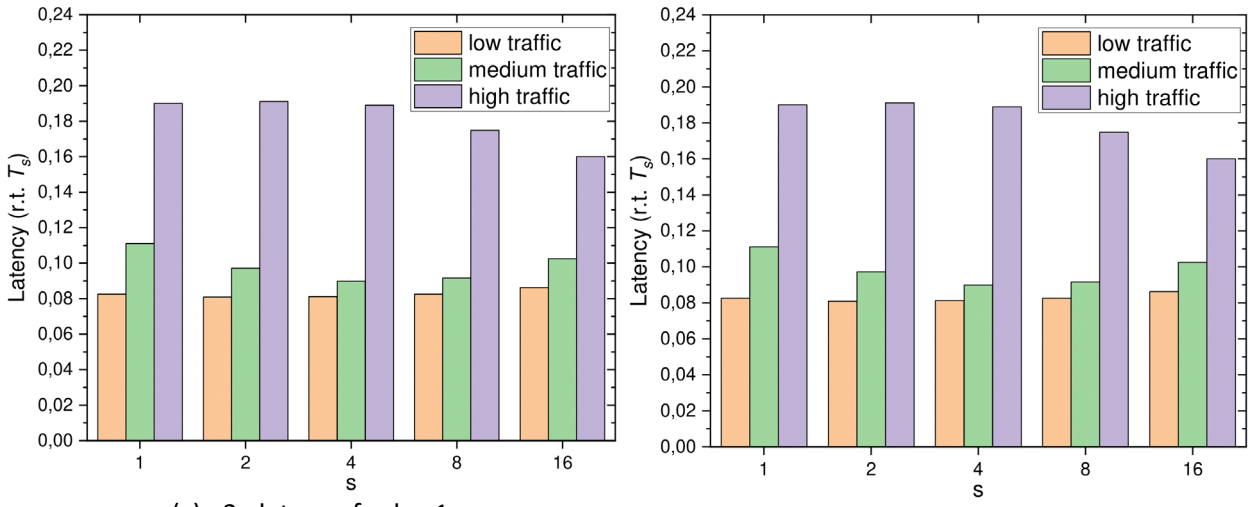
(b) Latency (secondary blockchain) for k = 1



(d) Latency (primary blockchain) for k = 3



(e) Latency (secondary blockchain) for k = 3



(c) e2e latency for k = 1

(f) e2e latency for k = 3

Figure 11. Performance comparison across various traffic scenarios and k values

Figure 11 presents the latency achieved by the primary and the secondary blockchains as well as the e2e system as a function of the number of concurrent users that can be served by a BS, s. In more detail, subfigures (a)-(c) assume that each block of the secondary blockchain can contain a maximum of $k = 1$ transactions, while the rest assume $k = 3$. By comparing the achievable latency of the primary blockchain as depicted in subfigures (a) and (d), it becomes evident that the latency is not affected by variations in the secondary blockchain's maximum capacity or traffic intensity. It is interesting to point out that for s values higher than 16, the latency increases significantly. This phenomenon is caused by the congestion brought on by the high traffic of the secondary blockchain that affects the primary one. Furthermore, the latency performance of the secondary blockchain is illustrated in subfigures (b) and (e), which showcase how greatly can k, ρ, and s affect the performance of the secondary blockchain. Specifically, high traffic values caused increased latency, while low k and s values can restrict the system's performance to a great extent. This highlights the importance of appropriately selecting the various degrees of freedom when designing B-RAN systems. Finally, subfigures (c) and (f) present the e2e latency of the system, which is shown to be significantly influenced by the number of contracts that can be included in a single block of the secondary blockchain. This emphasizes the equilibrium that is forged between traffic intensity and block capacity.

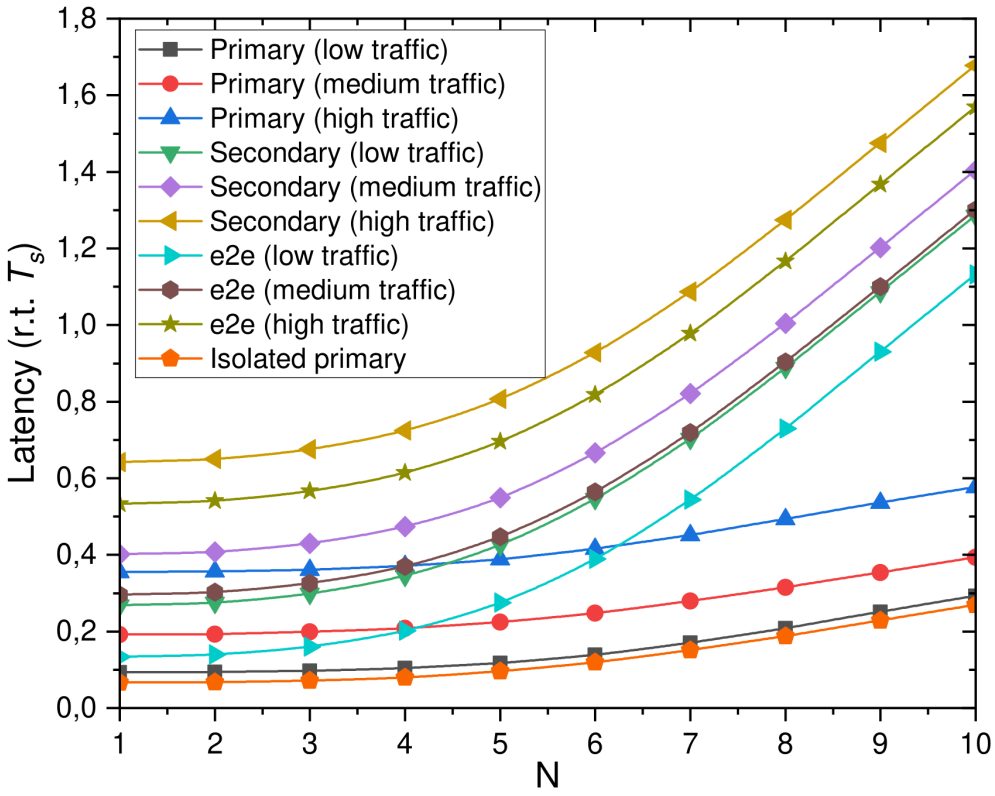


Figure 12. Latency vs N of the hierarchical B-RAN with multiple intensities

An extensive examination of the impact of different N confirmation numbers on latency is presented in Figure 12 for the primary and secondary blockchain. The plot shows the average total delay of the system along with the individual latency of the primary and secondary blockchains. Notably, the primary blockchain's behaviour remains constant throughout the plotted range of N, allowing for a focused examination of the impact of the secondary blockchain on the system. With three distinct traffic intensity values for the secondary blockchain – low, medium, and high – the plot demonstrates how various traffic scenarios influence the latency of the entire system. As anticipated, a higher traffic intensity correlates with increased latency across the board, while lower intensity results in lower latency. The average total delay of the system is particularly interesting since it resembles the behaviour of the secondary blockchain. This discovery implies that secondary blockchain activity has a major impact on the overall performance of the system. Moreover, this figure provides an understanding of the extra latency that the secondary blockchain adds to the major blockchain, particularly when contrasting with a situation in which the primary blockchain is isolated. This comparison demonstrates how much secondary blockchain activity affects the parent blockchain's delay. Overall, the plot emphasizes the complex interplay between primary and secondary blockchains, showing how changes in the characteristics of the secondary blockchain can impact the temporal dynamics of the entire system.

4. Attack Modelling

This section investigates the security landscape of B-RAN. Initially, the most prominent attacks are presented with a focus on the Blockchain and network aspects. Afterwards, the most influential attacks are described in detailed and modelled. The selected attacks are then connected with the B-RAN theoretical model and their impact is evaluated through numerical results.

4.1. Identified Attacks

4.1.1. Attacks on the Blockchain

Since B-RAN uses permissioned blockchains, the risk of external attacks is lower, but insider threats e.g. collusion, and cryptographic material compromises can become critical concerns. As mentioned in D5.2, we must here assume the potential malicious behavior of nodes that are registered and either paying or receiving funds for services, which is a rather ambitious assumption.

51% Attack

A 51% attack occurs when an adversary obtains control of more than half of the total mining or validation power in a blockchain network. With this majority control, the attacker can clandestinely mine an alternative blockchain fork that eventually surpasses the length of the honest network's chain. Once this longer fraudulent chain is released to the network, the attacker leverages the rule that the longest valid chain is accepted as the authoritative ledger to reverse previously confirmed transactions. In doing so, the attacker effectively enables double spending and fundamentally undermines the system's trust and immutability.

Although such an attacker cannot create new coins out of nothing or arbitrarily change the protocol's rules, their ability to unilaterally reorganize the blockchain demonstrates a critical vulnerability in systems lacking sufficient decentralization. Consequently, 51% attacks are regarded as both a theoretical possibility and a practical threat, particularly in networks where mining power is highly concentrated [39].

Selfish Mining

Selfish mining is a protocol deviation in which a miner or a coalition of miners deliberately withholds newly mined blocks instead of broadcasting them immediately to the rest of the network. The attacker's objective is to gain a lead over the public blockchain and cause honest nodes to waste their computational resources on mining blocks that will ultimately not be accepted into the main chain [40]. To execute this attack, the selfish miner maintains a private fork of the blockchain and continues to extend it in secret. The withheld blocks are released to the public network strategically at opportune moments to maximize the attacker's advantage. For example, if an honest miner finds a block, the selfish miner will promptly publish its own previously hidden block (or chain of blocks) in order to invalidate the honest miner's block

and keep the attacker's chain ahead, as shown in Figure 13. Through this tactic, the selfish miner secures block rewards that would have otherwise gone to honest participants.

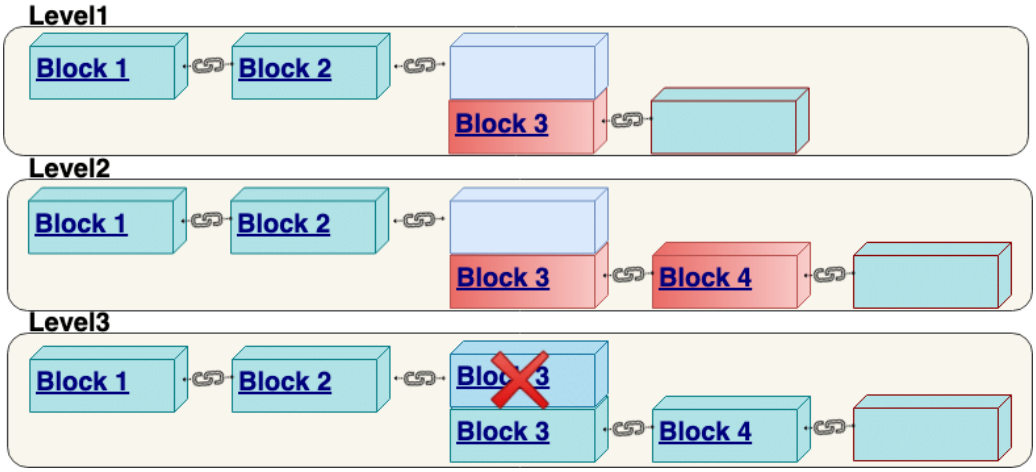


Figure 13. Selfish Mining Attack

This scheme carries inherent risks for the attacker. If the attacker's private chain fails to stay ahead of the public chain—for instance, if the honest miners happen to find a block before the attacker can extend the private chain—then the attacker's withheld blocks will be overtaken and discarded (or orphaned) by the network, yielding no reward. In other words, a selfish miner who cannot consistently outpace the rest of the network will find the benefits of this strategy negated. However, if the attacker commands a sufficiently large portion of the total hash power, they can often maintain a lead over the honest miners and thereby reap rewards greater than their fair share. Early analyses of selfish mining showed that an attacker could obtain a disproportionate share of mining rewards with significantly less than 50% of the network's hash power. In particular, initial studies estimated that selfish mining becomes net profitable when the attacker controls roughly one-quarter to one-third of the total hashing power. Later studies refined this threshold, suggesting that optimized strategies could make selfish mining viable with only around 21–23% of the total hash power [41].

Man-in-the-Middle (MITM) Attack

A MITM attack is a classic network security breach in which an adversary covertly intercepts and potentially alters the communication between two parties who believe they are directly communicating with each other. In this scenario, the attacker surreptitiously monitors the exchange and can inject new messages or modify legitimate messages in real-time [42]. To accomplish this, the attacker typically impersonates each party to the other. For example, the adversary might pose as a legitimate server to the client while simultaneously posing as the client to the server, thereby gaining full control over the information flow between them. This privileged position allows the attacker to harvest sensitive data (such as login

credentials or financial transaction details) and to manipulate the contents of the messages without detection by the legitimate participants.

In the context of blockchain networks, a MITM attacker could intercept and alter data transmissions related to transactions. For example, an attacker could intercept a transaction broadcast (such as an API call or a peer-to-peer network message containing a pending transaction) and change the transaction's destination address or other details before relaying it to its intended recipients [43]. If such communications are not protected by strong encryption and authentication, the participants would remain unaware of the tampering, potentially resulting in fraudulent transactions being accepted into the blockchain. Unlike attacks such as the 51% attack and selfish mining, which compromise the blockchain's consensus layer, a MITM attack targets the network communication layer. The attacker exploits weaknesses in network-level security (for example, insufficient encryption or lack of proper authentication) rather than any flaw in the blockchain's consensus protocol.

Collusion attack

A "collusion attack" is a type of security attack or threat in which a node intentionally makes a secret agreement with an adversary, or the node is somehow made to have such an agreement [44]. In Hyperledger Fabric, due to its particular ordering/endorsing schemes, different collusion attacks can theoretically take place.

- **Endorsing Peer Collusion**: Fabric relies on endorsement policies (e.g., "at least 2 out of 3 peers must approve a transaction"), which render this attack improbable and highly costly in terms of resources; however, if multiple endorsing peers collude, they can fabricate transactions and manipulate the ledger.
- **Orderer Node Collusion**: The ordering service (e.g., Kafka, Raft) determines the sequence of transactions. Here, if orderers collude, they can censor transactions or reorder them to benefit certain parties.
- **Application-Level Collusion**: Here, client applications can collude to submit malicious transactions or manipulate business logic in chaincode.
- **Identity Provider (MSP) Collusion**: Membership Service Providers (MSPs) issue identities for participants in Fabric. If an MSP colludes with certain participants, they can allow unauthorized access or fake endorsements.

Let us stress that the above are far reaching assumptions, especially in the case of NANCY, where a fastBFT consensus protocol is used. Some additional mitigation strategies could, however, include:

- **Strong Governance**: Carefully select and monitor participants.
- **Diverse Orderers & Peers**: Use multiple organizations to reduce single-party control.
- **Audit & Logging**: Regularly audit transactions and endorsements.

- **Zero-Trust Policies**: Assume nodes could be compromised and use cryptographic verification. For this, using a Hardware Security Module (HSM) is a recommended strategy.

Private Key Compromise in a validator node

Compromising a private key in a Hyperledger Fabric validator node (peer or orderer) can lead to serious security breaches, including transaction forgery, unauthorized endorsements, and ledger manipulation. There are several ways to compromise this key in Fabric:

- **Weak File Permissions**: Here, if the private key file `keystore/*_sk` has loose permissions, unauthorized users or processes can access it. A very simplistic example would be that a careless admin might set file permissions to `chmod 777`, allowing any user on the system to read it.
- **Malware or Keylogging Attacks**: Attackers can use malware, rootkits, or keyloggers to extract private keys from memory or disk. For instance, a compromised machine running a validator node may expose the key if an attacker gains remote access.
- **Docker Container Escape**: Hyperledger Fabric often runs inside Docker containers. If an attacker escapes the container, they can access the host file system and steal keys.
- **Man-in-the-Middle (MITM) Attacks**: If keys were to be transferred insecurely (e.g., over an unencrypted channel), an attacker could intercept and steal them.
- **Insider Threats**: A malicious admin or developer with access to the server could copy private keys.
- **Unpatched Vulnerabilities**: If the host OS, Fabric binaries, or cryptographic libraries are outdated, attackers can exploit vulnerabilities to gain access.

Let us again stress that the above are far reaching assumptions, especially in the case of NANCY, where proper file permissions, proper channel encryption and proper containerization engineering are expected. Some mitigation strategies are however listed:

- Store private keys in hardware security modules (HSMs) or secure key vaults instead of the file system. Additionally, restrict SSH access and prevent unauthorized users from logging into the node, and set strict file permissions:
  ```
  chmod 600 keystore/*_sk
  chown fabricuser:fabricgroup keystore/*_sk
  ```
- Apply encryption on the file system where private keys are stored (i.e. keystore/) or use Hardware Security Modules (HSMs) to protect keys from being directly accessed.
- Protect against malware e.g. run security monitoring tools like Falco to detect abnormal container behaviour, use anti-malware tools and endpoint security software on validator nodes, or limit third-party software installed on the node to avoid potential malware
- Harden Docker containers, e.g. using Docker namespaces and user namespaces to restrict permissions, enable SELinux or AppArmor to enforce security policies on Fabric containers, or run containers as non-root users.

- Secure network communication, e.g. using TLS encryption for all communication, and restrict access using firewall rules and VPNs.
- Implement Role-Based Access Control (RBAC) e.g. using Hyperledger Fabric's MSP roles to restrict node access and set up multi-admin approvals before making critical changes.
- Use audit logs to track key access and identify anomalies and periodically rotate private keys and certificates.

Using a hardware security module (HSM)

Several mitigation strategies, beyond proper software engineering, include the use of a hardware security module (HSM) for both protecting the system from insider threats and private key compromises. The cryptographic operations performed by Fabric nodes can thus be delegated to an HSM.

An HSM protects private keys and handles cryptographic operations, allowing peers and orderer nodes to sign and endorse transactions without exposing their private keys [45]. If the system requires compliance with government standards such as FIPS 140-2, there are multiple certified HSMs from which to choose. Fabric currently leverages the PKCS11 standard to communicate with an HSM. This is also what NANCY currently uses to communicate with the PQC token.

To use an HSM with a Fabric node, one needs to update the `bccsp` (Crypto Service Provider) section of the node configuration file such as `core.yaml` or `orderer.yaml`. In the `bccsp` section, we must select PKCS11 as the provider and enter the path to the PKCS11 library that we would like to use. Administrators also need to provide the `Label` and `PIN` of the token that was created for our cryptographic operations. We can use one token to generate and store multiple keys.

The prebuilt Hyperledger Fabric Docker images are not enabled to use PKCS11. For deploying Fabric using docker, one needs to build own images and enable PKCS11 using the commands available in the Hyperledger documentation. We must also ensure that the PKCS11 library is available to be used by the node by installing it or mounting it inside the container.

The numeric improvement [46] of using a Hardware Security Module (HSM) in Hyperledger Fabric depends on several factors, including the HSM model, Fabric configuration, transaction load, and cryptographic operations performed. However, general benchmarks and studies indicate the following performance improvements in Table 1.

Table 1. Performance metrics with and without HSM

Metric	Without HSM	With HSM
Transaction Signing Speed	~200-500 TPS (transactions per second)	2x-10x improvement
Endorsement Latency	10-50 ms per transaction	Reduced by 30-70%
CPU Usage on Validator Nodes	High (CPU-bound operations)	Lower (Offloaded to HSM)

Security	Private key stored in memory/disk	Zero-exposure of private keys: The private keys never leave the HSM, significantly improving security. Tamper resistance: HSMs are resistant to physical and logical attacks. FIPS 140-2 Compliance: Many HSMs comply with high security standards.

Currently, there is a lack of publicly available, detailed performance comparisons between different HSM brands and models within Hyperledger Fabric environments. While some studies have explored hardware acceleration in Fabric, they often focus on custom solutions rather than commercial HSMs [47]. The paper "Blockchain Machine: A Network-Attached Hardware Accelerator for Hyperledger Fabric" [48] discusses a custom hardware accelerator achieving up to a 12x speedup in block validation, resulting in commit throughput of up to 68,900 transactions per second.

Exploring custom solutions for empowering B-RAN with HSM is, in fact, an interesting idea for securing B-RAN architectures such as NANCY in the future.

4.1.2. Attacks on the Network

Due to their open specifications and interoperability requirements, the open radio access network (O-RAN) is vulnerable to various threats focusing on the networking layer. The aforementioned characteristics increase the attack surfaces, making O-RAN susceptible to DoS and DDoS attacks. Particularly vulnerable are centralized control points and virtualized network functions, which can become single points of failure [49].

In more detail, DoS attacks aim to disable services by overwhelming the network resources with excessive traffic or resource-intensive requests, typically originating from a single source. The goal of these attacks is to exhaust bandwidth, computational power, or storage capacity, resulting in the denial of legitimate service requests. Furthermore, DDoS attacks extend this threat by using multiple compromised systems or devices organized into botnets, amplifying their impact and making detection and mitigation much more difficult. Specifically, the distributed nature of DDoS attacks makes it difficult to identify and block malicious traffic, as the attack originates simultaneously from geographically dispersed and independently operated sources [49]. A specialized variant of DDoS attacks is known as the slow or "low-and-slow" DDoS attacks. In contrast with traditional DoS attacks involving large volumes of requests, slow DDoS attacks use minimal bandwidth for large periods of time. Slowloris and HX-DoS are such attacks which initiate multiple connections but do not complete them, holding the host resources through minimal interactions. As a

result, the resources are exhausted without triggering standard defenses, making these attacks difficult to detect and mitigate [50]. Figure 14 illustrates an example of DoS attacks where a malicious user sends massive requests with the aim of exhausting the resources of the O-RAN components.

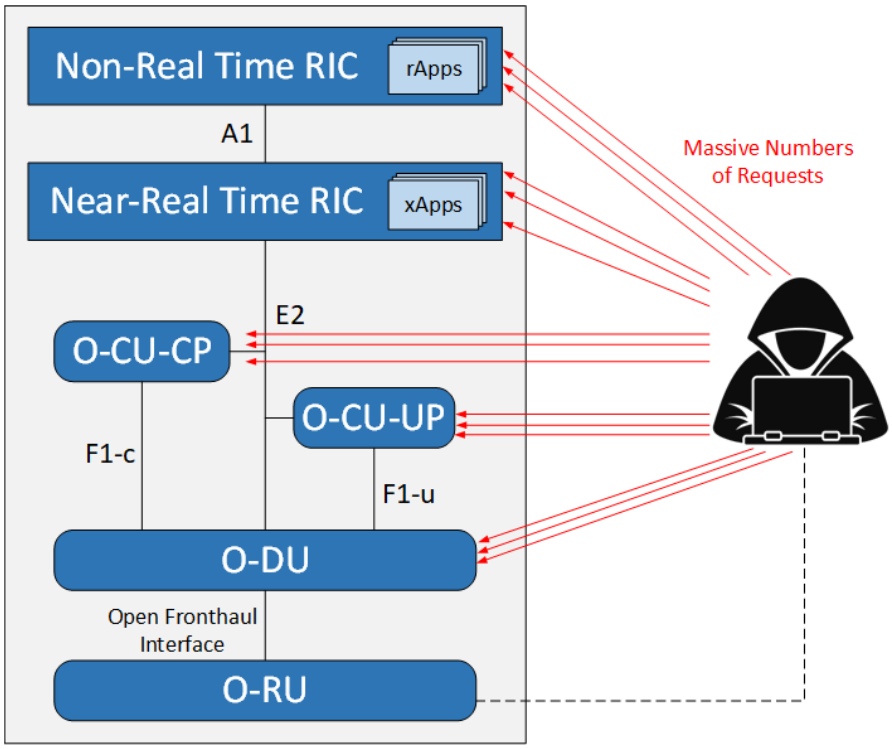


Figure 14. DoS attacks against O-RAN

Reconnaissance attacks are focused on host detection and mapping, as well as service identification, in order to obtain information such as IP addresses, open ports, and other critical details [51]. The information gathered through the reconnaissance attacks serves as a basis for attackers to exploit potential vulnerabilities to compromise the system security [52, 53]. In the context of O-RAN, attackers can target the various hosts of O-RAN components, as well as the user equipment. Also, the disaggregated nature of O-RAN increases the possibility of misconfigurations in particular servers that host the O-RAN components. To this end, reconnaissance attacks, such as port and service scanning and fingerprinting, can expose vulnerabilities that can be exploited by adversaries [54]. An illustration of the reconnaissance attacks is presented in Figure 15.

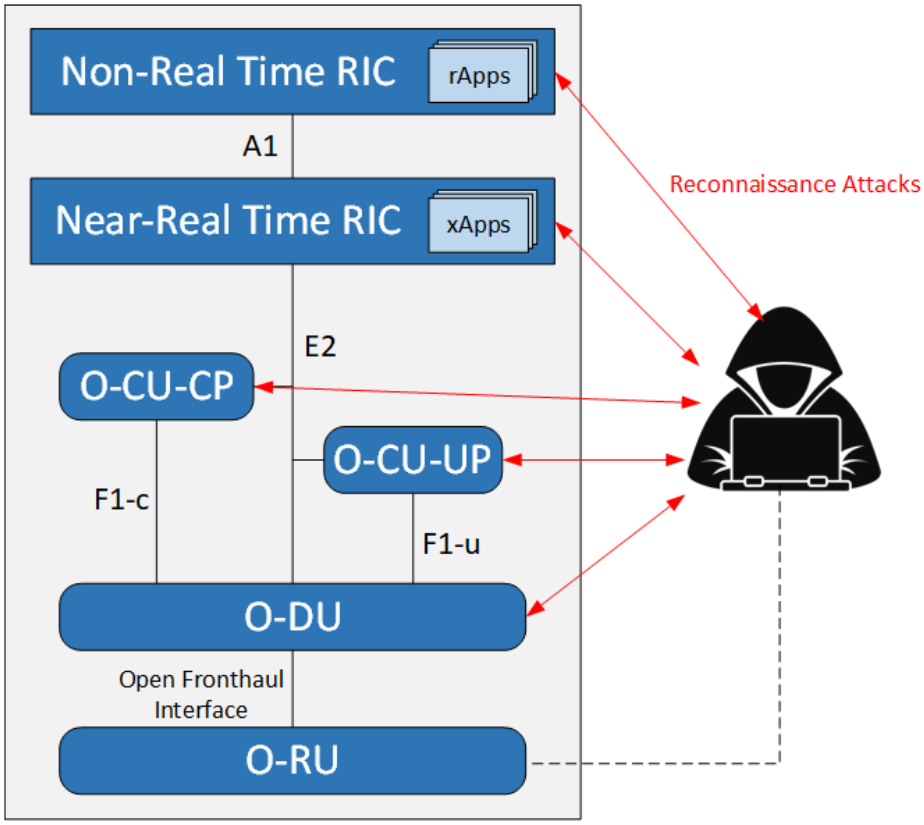


Figure 15. Reconnaissance attacks at O-RAN component

Finally, in MITM attacks, an attacker aims to intercept and manipulate communications between two entities. This allows the attacker to modify messages, steal sensitive data, or disrupt services. MITM attacks often exploit weaknesses in communication protocols or insecure network configurations to gain unauthorized access [55]. Similarly, an eavesdropping attack is a passive attack in which an adversary eavesdrops on network traffic to extract sensitive information. Unlike MITM attacks, which involve active interference, eavesdropping focuses on surveillance, allowing attackers to gather information undetected. This is particularly dangerous in networks that handle personal information, authentication credentials, or control signals [49, 56].

Assuming a post-intrusion scenario in O-RAN context, the attacker being in the same subnetwork as the RICs can launch address resolution protocol (ARP) spoofing attacks to compromise the mapping between MAC and IP addresses [57]. Upon successful implementation of the attack, all data exchanged through the A1 interface will pass through the attacker. As a result, the attacker can tamper with the decision-making processes of RICs. Furthermore, the O-RAN interfaces can expose sensitive user information, such as location and authentication credentials if the security mechanisms are weak or improperly implemented. For instance, an attacker can intercept the communications between the UE and the BS in order to

eavesdrop and/or manipulate the network traffic [58]. Figure 16 illustrates a MiTM attack between the two RICs. In the particular scenario, ARP spoofing is employed to compromise the MAC-to-IP address mapping.

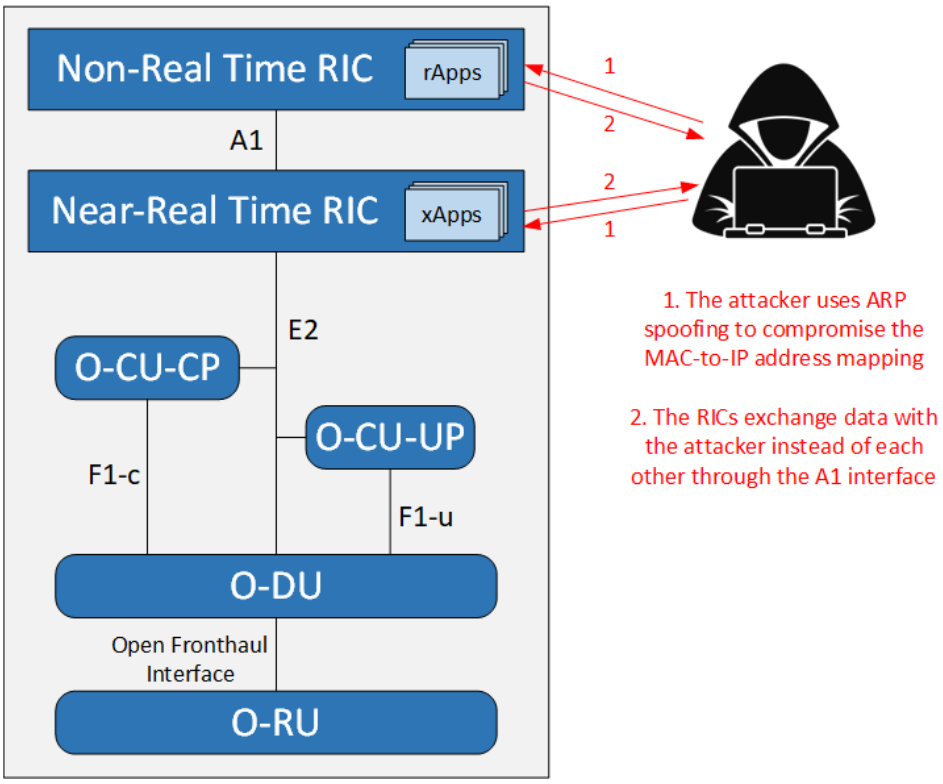


Figure 16. MITM attack between the two RICs

4.2. Simulated Attacks

4.2.1. 51% attack modelling

In this section, we consider the scenario of a 51% attack (specifically double spending) on BRAN and provide closed form expression of the probability of a successful attack. This part demonstrates, by careful inspection and analysis, the robustness and advantages of the B-RAN model across a range of network topologies and with potential security risks.

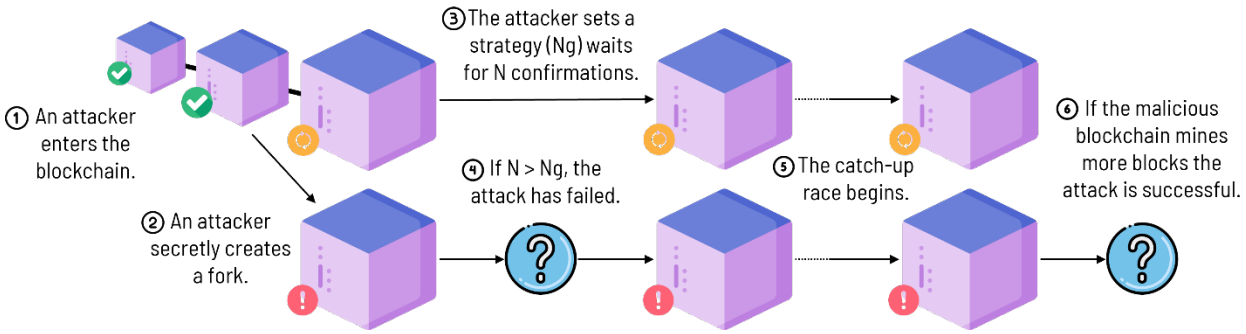


Figure 17. Procedural illustration of an alternate history attack in B-RAN

The incorporation of blockchain technology into RAN systems has the potential to improve security and avoid attacks by malicious users. The decentralized and transparent design of blockchain enhances its resilience against attacks. However, the structure of blockchain raises new security risks that were not present in earlier RAN systems. A typical example is the alternative history attack, which includes malicious attempts to modify the transactions in the blockchain's history. This attack scenario is examined in detail in the following section, along with how it could affect B-RAN's performance or compromise its dependability and security.

In the case of the alternative history attack, as seen in Figure 17, an attacker initially gains access to the blockchain as a regular user. At some point, along with the official mining process, the attacker creates an exact duplicate of the official blockchain. Despite the differences in mining rates between the two versions (official and malicious), official blockchain activities are unaffected. The mining rate of the malicious fork, Rm, is determined by the computational capabilities of the attacker. Additionally, the symbol beta represents the ratio between legitimate and malicious blockchains. Once the tampered block gathers N confirmations, the attacker initiates a mining race to catch up with the official blockchain. The attacker evaluates the length of the malicious fork compared to the original chain. If this difference falls below a specified threshold N_g, the attacker persists in mining until the malicious chain surpasses the official one and deems the attack as successful. On the contrary, if the difference exceeds Ng, the attacker ceases the attack deeming it unsuccessful. The likelihood of a successful alternative history attack depends on the attacker's relative mining rate, β, the required number of confirmations, N, and the attacker's strategy, N_g.

4.2.2. Sybil attack modelling

A Sybil-type attack occurs when the attacker creates multiple fake users in order to gain influence over the network. As Blockchain systems rely on the peer-to-peer (P2P) architecture for their decentralization, the above attack is obviously very dangerous. The attacker using multiple false users can control consensus mechanisms, to disrupt communication, or compromise the integrity of transactions. The basic idea of the above method, as presented in [59] exploits the lack of a centralized authority for verifying node identities,

making decentralized systems such as blockchain vulnerable to such attacks. In its basic form, a Sybil attack allows an attacker to control a significant portion of the network, allowing them to alter data propagation, block validation, or even undermine trust in the system.

Modern Blockchain systems rely on a decentralized network of nodes to serialize and verify transactions in a public ledger. Transactions are gathered into blocks, which are then "mined" by solving a cryptographic puzzle. It is impossible for most users to revert previous payments, as successful modification of the chain typically requires majority (or near-majority) mining power. However, when an attacker combines a double-spend attack (spending the same coins in two conflicting payments) with a Sybil attack (computing many artificial or "Sybil" identities), the attacker can exploit communication latency among honest nodes and catch up with or even surpass the legitimate blockchain more readily. By doing so, the attacker can erase a transaction that appeared valid, letting them walk away with both the purchased goods and the supposedly spent coins.

In order to illustrate the synergistic effect of Sybil and double-spend techniques, we now present a detailed, step-by-step breakdown of how a malicious party can orchestrate this combined attack. By injecting numerous Sybil nodes—fake participants that do not genuinely contribute to mining—into the blockchain's peer-to-peer network, the attacker deliberately disrupts the normal flow of information. When legitimate miners are unaware of newly discovered blocks or receive them too late, their efforts can be wasted on outdated forks. Meanwhile, the attacker quietly builds a private chain in secret. If this hidden chain eventually becomes longer than the recognized public chain, the protocol will switch to the attacker's version, rendering earlier "legitimate" transactions invalid. The following stages outline precisely how the attacker leverages this environment to carry out a successful double-spend:

1. **The attacker joins the network**: The malicious party first starts with a typical mining process to participate in the blockchain network. Secondly, he utilizes numerous Sybil (spurious) nodes. The Sybil nodes are essentially mining-incapable but appear to be ordinary participants.
2. **Initiating double-spend**: The attacker sends a first transaction (TX0) to purchase some service. Simultaneously, the attacker secretly prepares a conflicting transaction (TX1) to return the same money back to themselves.
3. **Secret chain and Sybil interference**: While honest miners are building the main (public) chain incorporating TX0, the attacker secretly mines the block in some other unknown chain incorporating TX1. The Sybil nodes, meanwhile, meddle in the block discoveries propagation to the legitimate participants. The fake identities issue "invite" to the legitimate participants, but don't provide them with the blocks when the legitimate participants request them. This artificially decelerates the propagation among legitimate participants.
4. **Gaining an advantage**: Because honest miners are forced to wait longer for real block information, some of their efforts can be wasted on outdated forks or missing blocks. The attacker, meanwhile,

faces no delay in extending their secret chain. If the attacker's hidden chain eventually becomes longer than the main chain, the attacker broadcasts it. By the rules of the blockchain, the network accepts the longer chain, and TX0 effectively disappears (it is replaced by TX1). Thus, the attacker keeps the goods they purchased with TX0 and also retains the original coins via TX1.

5. Unified B-RAN and Attacks Modelling

This section describes the integration of the selected attacks, which were described in Section 4.2., with the theoretical B-RAN modelling that was presented in Section 3. The security of B-RAN is emphasised and is evaluated by means of the successful attack probability, while an equilibrium between performance and security is highlighted.

5.1. Performance Evaluation

5.1.1. 51% attack

To evaluate the probability of a successful attack we assume stable strategy level and mining rates. We consider a scenario where the probability of extending the official chain by one block is $\frac{1}{1+\beta}$, while the likelihood of an attacker to find the next block is $\frac{\beta}{1+\beta}$. This implies that the mining process can be modeled by a series of independent Bernoulli trials with a success probability of $\frac{1}{1+\beta}$. For the attack to be successful, the attacker must deliberately wait for N confirmations. At the same time, the attacker generates n_Y blocks on the malicious fork. Consequently, the stochastic variable denoting the number of failures, Y and follows a negative binomial distribution, $Y \sim NB\big(N, 1/(1+\beta)\big)$, with the probability mass function given by

$$\Pr\left\{Y = n_Y; N, \frac{1}{1+\beta}\right\} = \binom{n_Y + N - 1}{n_Y}\left(\frac{1}{1+\beta}\right)^N \left(\frac{\beta}{1+\beta}\right)^{n_Y}, \tag{29}$$

where $\binom{n}{k}$ denotes the binomial coefficient. Afterwards, both the malicious and the official blockchains start mining with the attacker trying to outperform the official network. If this happens, the attacker can publish the malicious chain and rewrite the confirmed history. However, if the fraudulent chain lags behind by N_g blocks, the attacker abandons the attempt. Let $P_n = \Pr \mathrm{Win}|z = n$ denote the probability of the attacker winning despite starting with a delay of n blocks. Two special cases become evident, specifically $P_{-1} = 1$ and $P_{N_g} = 0$. If the attacker finds the next block, the malicious chain shortens by $n - 1$ blocks compared to the benign chain, and the success probability becomes P_{n-1}. Conversely, if the official blockchain mines a block, the attacker falls further behind to $n + 1$ blocks and the success probability decreases to P_{n+1}. By conditioning on the outcome of the first generated block, the probability of the attacker winning can be written as

$$P_n = \frac{1}{1+\beta}P_{n+1} + \frac{\beta}{1+\beta}P_{n-1}, \qquad 0 \le n < N_g, \tag{30}$$

Which can be further reformulated as

$$P_{n-1} - P_n = \frac{1}{\beta}(P_n - P_{n+1}), \qquad 0 \le n < N_g. \tag{31}$$

For $n = N_g - 1$, the previous equation can be rewritten as

$$P_{N_g-2} - P_{N_g-1} = \frac{1}{\beta}\left(P_{N_g-1} - P_{N_g}\right) = \frac{1}{\beta}P_{N_g-1}, \tag{32}$$

which, through recursion, yields

$$P_{N_g-n-1} - P_{N_g-n} = \frac{1}{\beta^n}P_{N_g-1}, \qquad 0 \le n < N_g, \tag{33}$$

that can be rewritten as

$$P_{N_g-n-1} = P_{N_g-1} + \sum_{m=1}^{n} \frac{1}{\beta^m}P_{N_g-1}, \tag{34}$$

By expanding the sum, the previous equation can be transformed into

$$P_{N_g-n-1} = \begin{cases} P_{N_g-1}\dfrac{1-1/\beta^{n+1}}{1-1/\beta}, & \text{if } \beta \neq 1 \\ P_{N_g-1}(n+1), & \text{if } \beta = 1. \end{cases}, \tag{35}$$

Next, by utilizing the boundary condition P−1 = 1, (35) can be rewritten as

$$P_{N_g-1} = \begin{cases} \dfrac{1-1/\beta}{1-1/\beta^{N+1}} & \text{if } \beta \neq 1 \\ \dfrac{1}{N_g+1} & \text{if } \beta = 1 \end{cases}. \tag{36}$$

Hence, we derive the expression for P_n as

$$P_n = \begin{cases} \dfrac{\beta^{n+1} - \beta^{Ng+1}}{1-\beta^{N+1}} & \text{if } \beta \neq 1 \text{ and } 0 \le n < N_g \\ \dfrac{N_g - n}{N_g + 1} & \text{if } \beta = 1 \text{ and } 0 \le n < N_g \\ 1 & \text{if } n < 0 \\ 0 & \text{if } n \ge N_g. \end{cases}, \tag{37}$$

As a result, assuming the official blockchain extends N blocks and the malicious n_Y, the attacker commences the race trailing by $(N - n_Y)$ blocks. In this case, the probability of a successful alternative history attack can be expressed as

$$S(N,\beta,N_g)=\sum_{n_Y=0}^{\infty}\Pr\{\text{Win}\mid z=N-n_Y\}\Pr\left\{Y=n_Y;N,\frac{1}{1+\beta}\right\}, \tag{38}$$

or equivalently

$$S(N,\beta,N_g)=\sum_{n_Y=0}^{\infty}\binom{n_Y+N-1}{n_Y}\left(\frac{1}{1+\beta}\right)^{N}\left(\frac{\beta}{1+\beta}\right)^{n_Y}P_{N-n_Y} \tag{39}$$

Finally, by exploiting the identity

$$\sum_{n=0}^{\infty}\binom{n+N-1}{n}\left(\frac{1}{1+\beta}\right)^{N}\left(\frac{\beta}{1+\beta}\right)^{n}=1, \tag{40}$$

(39) can be rewritten as (41), shown below. We can conclude that the success of an attack it depends on a plethora of parameters such as the hash power of the attacker, the N_g threshold, the mining rate Rm of the official blockchain, as well as the official's confirmation number N. A higher N value, while can be slowing down the speed of the blockchain and increase the total latency of the model, at the same time it increases the security of it, by challenging the malicious chain to keep the pace with the official. Additionally, by adopting the longest-chain rule in the official, priority is given to the value of proof of work as this approach is widely recognized for maintaining data integrity and making malicious attacks more difficult as they require greater computational power (hash power) by the attackers [60].

$$\mathrm{S}(N,\beta,N_g)=\begin{cases}1-\sum_{n=0}^{N}\binom{n+N-1}{n}\left(\frac{1}{1+\beta}\right)^{N}\left(\frac{\beta}{1+\beta}\right)^{n}\left(\frac{1-\beta^{N-n+1}}{1-\beta^{Ng+1}}\right) & \text{if } \beta\neq 1\\ 1-\sum_{n=0}^{N}\frac{1}{2^{N+n}}\binom{n+N-1}{n}\left(\frac{N-n+1}{N_g+1}\right) & \text{if } \beta=1\end{cases}. \tag{41}$$

5.1.2. Sybil attack

In order to evaluate the successful Sybil attack probability, it is essential to formulate a probabilistic model that captures both the hidden chain mining by the attacker and the deliberate communication delays introduced by Sybil nodes. The analysis presented in this section highlights how the combination of slowing down honest nodes' propagation and privately mining conflicting blocks significantly increases the attacker's chances of carrying out a successful double-spend.

Let Z denote the number of blocks the merchant waits to confirm before releasing goods, while P is the probability that the attacker successfully creates a secret chain that overtakes the main chain. Therefore, the overall success probability can be expressed as in [61]

$$P = 1 - \sum_{k=0}^{z-1} [\Pr[X_z = k](1 - P_{z-k})], \tag{42}$$

where $Pr[Xz = k]$ is the probability the attacker mines k blocks while the main chain mines z and P_{z-k} is the probability that, given the attacker is $z - k$ blocks behind, it can still catch up.

The probability $Pr[Xz = k]$ follows a negative binomial distribution and can be written as

$$\Pr[X_z = k] = \lambda_1^k \lambda_2^z \binom{k + z - 1}{k}, \tag{43}$$

where λ_1 and λ_2 depend on (i) the attacker's fraction of mining power q, (ii) the impact of delays caused by Sybil nodes, and (iii) how quickly the main chain grows despite these delays. The analytical expressions of λ_1 and λ_2 are given by

$$\lambda_1 = \frac{\beta}{\beta + \delta}, \tag{44}$$

and

$$\lambda_2 = \frac{\delta}{\beta + \delta}, \tag{45}$$

in which β is the has rate of the attacker and δ represents the growth rate of the main chain. Moreover, P_{z-k} is given by

$$P_{z-k} = \left(\frac{\alpha}{\beta}\right)^{z-k}, \tag{46}$$

where α is the attacker's effective mining rate, and β is the "effective growth rate" of the legitimate chain, which is slowed down by the communication delays introduced by Sybil nodes.

By substituting (43) and (46) into (42), the latter can be rewritten as

$$P = 1 - \sum_{k=0}^{z-1} \left[\left(\lambda_2^z \lambda_1^k\right) - \left(\lambda_1^z \lambda_2^k\right)\right] \binom{k + z - 1}{k}, \tag{47}$$

with λ_1 and λ_2 are carefully defined expressions involving both the attacker's fraction of mining power and the fraction of Sybil nodes that slow down honest block propagation.

In summary, Sybil nodes artificially slow down the legitimate network's block communication. This helps the attacker more effectively use their own mining power to overtake the honest chain. If they do, the attacker voids their "public" payment transaction and reclaims the same funds, thus completing a double-spend. This attack highlights how network-layer manipulation can lower the mining power threshold needed for a successful double-spend, making it more dangerous than the standard assumption that one needs the majority of the hash power.

5.2. Numerical Results

This section sheds light on the security aspects of the proposed B-RAN model by providing numerical results that were collected based on the modelling of the alternate history attack that was presented in Section 4.2.1. The demonstrated results focus on the interaction among various degrees of freedom of B-RAN and provide interesting discussions on its adaptability to various configurations. This analysis is crucial, as attackers can target either the primary or secondary blockchain of the proposed framework. Moreover, the security achieved is compared to other works. This allows us to capture the dynamic nature of security challenges within B-RAN and provide design guidelines that ensure robust protection against potential attacks.

5.2.1. 51% attack

Figure 18 presents the probability of a successful attack as a function of the rate between the hash power of the official and malicious blockchains. Different attack strategies, Ng, of the attacker and various numbers of confirmations, N, are taken into consideration in the analysis. All of the scenarios include both the proposed and conventional BRAN modelling approaches. It immediately becomes evident that the two modelling approaches provide similar results, which validates the validity of the proposed framework. As expected, the probability of successful attacks increases as the β values increase, while it approaches 100% when the malicious and official blockchains have comparable mining power. This is the case for both $N = 1$ and $N = 3$ configurations, with the latter exhibiting better security performance of 2×10^{-3} for low β values. This observation indicates a consistent behavioral pattern for both configurations across various attack scenarios, regardless of variations in Ng values. The convergence of the two configurations suggests a shared vulnerability for B-RAN systems that is introduced due to the existence of blockchain. Finally, it is important to highlight that the official blockchain is characterized by robust computational capabilities that cannot be easily matched by the malicious one.

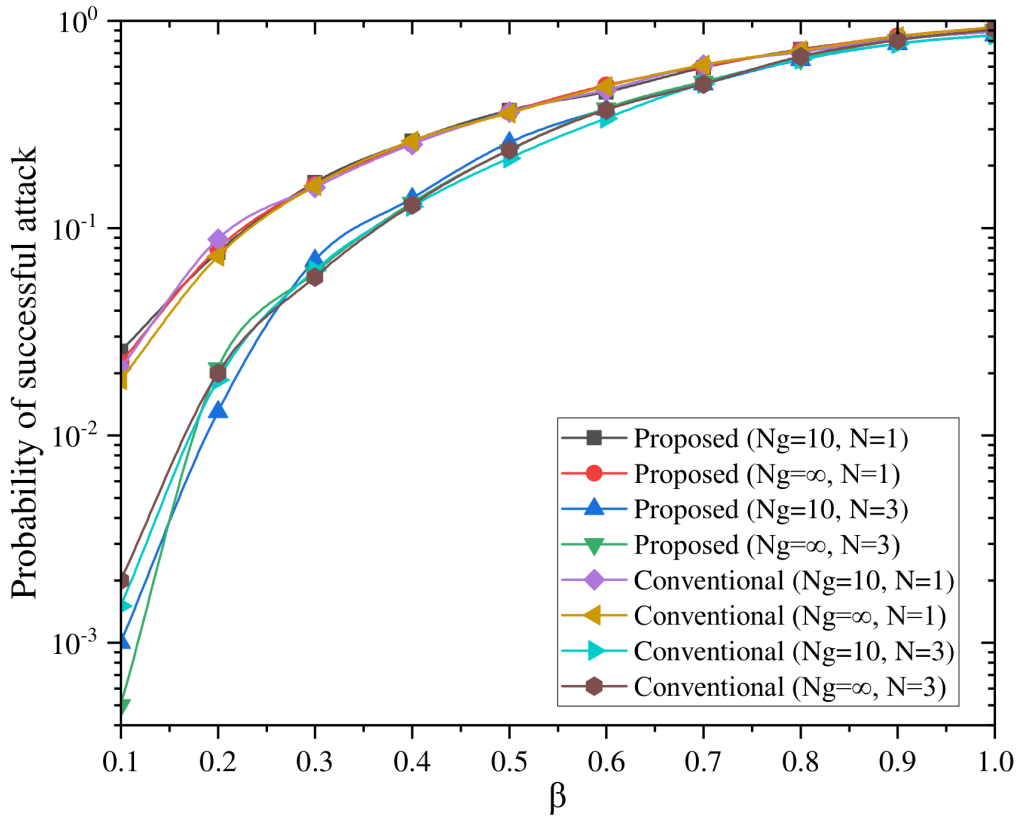


Figure 18. Probability of successful attack vs the attacker's mining power for multiple combinations of Ng and N

Figure 19 illustrates the interactions between latency and security for various different configurations of B-RAN. Specifically, six combinations are drawn for $s = 10$ or 25 and k values equal to 1, 2, or 3. By observing any of the plotted configurations, it is evident that as security increases the latency increases as well. This highlights an equilibrium between security and latency when designing BRAN systems. In systems where security plays a significant role, a trade-off with temporal performance is expected, and vice versa. Although at first glance different configurations appear to have similar behavior, a closer inspection uncovers significant differences. Specifically, the black line that is characterized by $k = 1$ and $s = 10$ indicates the worst performance in both security and latency, whereas the yellow line of $k = 3$ and $s = 25$ is the fastest and the most resilience to attacks. Furthermore, the models with higher s values demonstrate a notable improvement in security when compared to their low-s counterparts, which highlights the important role of s. Moreover, a closer investigation of the $s = 25$ configurations reveal a medium variation between $k = 1$ and $k = 2$, but only a minor difference between $k = 2$ and $k = 3$. This indicates that after a certain point increasing the k value does not result in improved performance. Overall, this figure underlines the significance of appropriately selecting the design parameters of B-RAN to achieve the intended latency without sacrificing security.

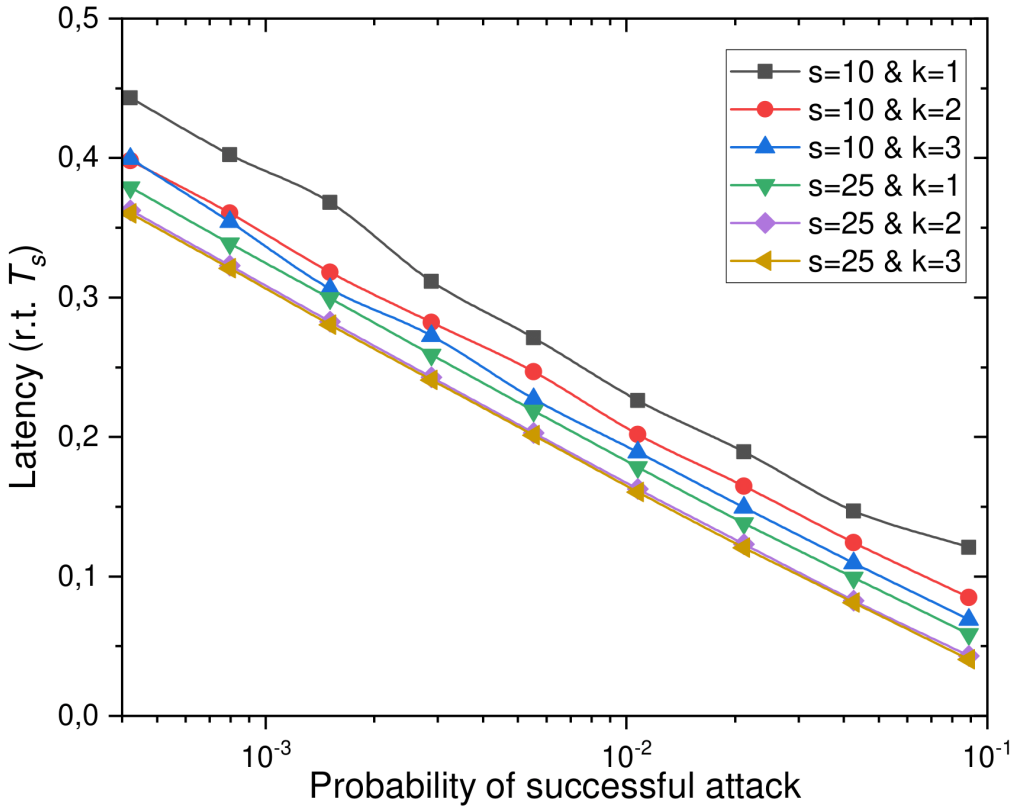


Figure 19. Security vs latency for different s and k configurations

5.2.2. Sybil attack

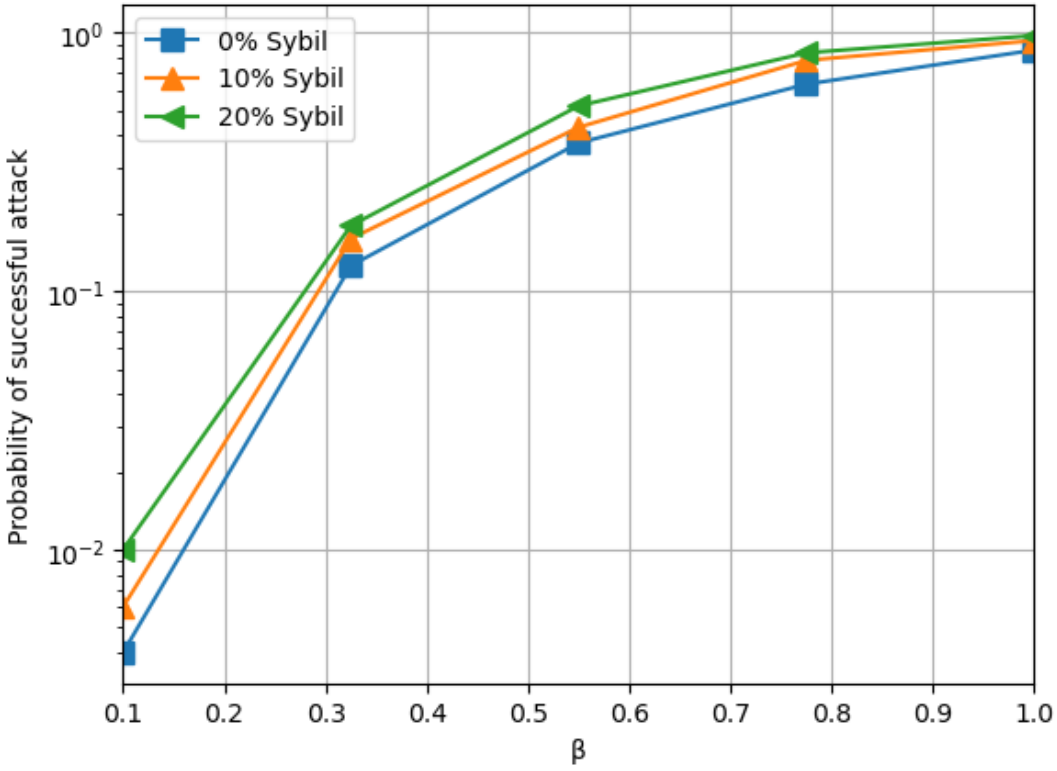


Figure 20. Probability of a successful Sybil attack as a function of β

Figure 20 illustrates the relationship between beta power (β) of the attacker and likelihood of a successful attack in a blockchain simulating combined double spend (51%) and Sybil attacks. There are three cases shown: the baseline case of 0% Sybil nodes shown in blue is a pure double spend attack; an intermediate case of 10% Sybil nodes is shown in orange; and an advanced attack of 20% Sybil nodes is shown in green. All three cases show comparable increasing trajectories as beta power increases, culminating in the predicted convergence towards near-certainty, a probability of approximately 100%, as beta approaches 1.0. The pure form double spend attack always has the minimum success probability for any beta value, whereas the 20% Sybil attack provides the attackers with the maximum success probability throughout the range. The intermediate 10% Sybil case, as anticipated, is somewhere in between these two extremes, indicating the incremental advantage that accrues by adding more fake (Sybil) nodes. It is notable that all scenarios begin with low probabilities of success, i.e., below 1%, at β = 0.1; however, they all show a steep turning point around β = 0.3, after which the probabilities surpass the 10% mark. The most rapid rate of increase is achieved between hash power levels of 0.2 and 0.6, after which the curves begin to flatten out as they approach certainty. This graph clearly illustrates the way in which introducing Sybil nodes

significantly optimizes attack effectiveness to facilitate smooth compromise of blockchain networks with relatively moderate computational power.

6. Intelligent Optimisation

In this section, we explore the intelligent optimizations in blockchains, in which the AI-enhanced algorithms are used to optimize performance, security, and scalability. With more and more networks in blockchains, problems like low transaction processing speed, energy efficiency, security and network congestion arise. Intelligent optimization techniques like artificial intelligence and machine learning help in solving these issues by improving consensus mechanisms, resource utilization, and overall efficiency. Through the integration of such intelligent solutions, blockchain systems can operate more efficiently, and their long-term stability and flexibility to new technology are ensured.

Important research has been done in AI and blockchain, since AI can process enormous volumes of data, optimize transactions, and automate decision-making, reducing the need for human input. Such an integration not only makes blockchain systems perform better but also enables smarter fraud detection and risk assessment functionalities. According to [62], the application of AI in blockchain has been explored thoroughly and shown to simplify tasks and make systems smarter. AI agents, when implemented alongside smart contracts, oversee performing repetitive tasks within blockchain networks. This automation allows human resources to focus on more sophisticated and creative tasks, optimizing efficiency across industries. AI combined with blockchain opens new doors for innovation, particularly in the finance, healthcare, and supply chain management industries. With the application of AI, blockchain-based systems can dynamically respond to changing circumstances, optimizing security, performance, and decision-making processes [63]. However, blockchain technology involves a set of trade-offs such as security, performance, and decentralization. AI techniques can simplify these complex decisions by automatically optimizing procedures and improving governance processes [64]. Blockchain, although useful, is costly because it involves high computation and storage needs. Additionally, scalability is an impressive concern in the application of AI and blockchain because additional consensus processes and storage needs can slow down processes [62]. But AI, because of its ability to process large data and leverage high computing power, can be employed to make blockchain networks scalable [65] so that they are efficient and dynamic enough to cater to future demands.

6.1. Reinforcement Learning Framework

In this section, we present a RL method in order to optimize the performance of a B-RAN. We built an RL agent learned from simulation B-RAN data, able to adjust blockchain parameters at runtime in an attempt to maintain request latency close to a specific target. The agent is designed to operate in a live environment, continuously adapting to network conditions when the conditions require it. Additionally, it optimizes resource usage, ensuring effective performance while minimizing computational and energy consumption. This approach enhances both the efficiency and scalability of the B-RAN, making it more suitable for real-world applications.

Our framework utilizes a closed-loop control architecture with an RL Proximal Policy Optimization (PPO) to control latency times and optimize blockchain resource allocation networks (BRAN), as can be seen in Figure 21. The architecture tries to keep the network latency close to defined thresholds while reducing resource utilization through intelligent parameter tuning. Our system is composed of three main components. First, the observation module is always gathering network measurements in real-time, such as average latency over sliding time windows, along with the mining power which denotes the blockchain network's capabilities and service rate which represents the network resources assigned to the service provisioning. These values are forwarded to the agent, which is the decision-making component of our system. Afterwards, the agent decides on network conditions and makes optimal parameter setting decisions, prioritizing actions that maintain the latency around a target latency level while at the same time choosing minimum achievable resource allocation. Finally, the actuation module of the agent implements these choices by adapting BRAN parameters like mining capacity and service execution rates. For state representation, we encode the current network conditions as a vector incorporating latency measurements, resource utilization, and proximity to a critical latency threshold. Complete information about the state allows the agent to make knowledgeable decisions via complete network observability. The action space involves discrete parameter adjustments with inherent constraints to avert system overloading, for instance, the agent is unable to reduce mining power to unsafe minimum values or to exceed the available resources. Additionally, our training procedure utilizes a simulated BRAN environment that replicates real traffic patterns and network conditions.

The simulation can create varied scenarios such as abrupt traffic surges to provide robustness to the agents. The reward function weighs two conflicting goals: keeping latency under thresholds effectively and reducing parameter changes to save resources. The two-objective reward promotes resource-conserving utilization. The PPO updates the policy of the agent iteratively during training, weighing the exploration of new actions against the exploitation of old successful strategies. Once deployed, the system is integrated into the live BRAN model to retrieve metrics and push configuration changes. The agent is conservative in its actions, initiating parameter changes only when latency times are above a threshold. This helps achieve system stability through a minimal amount of configuration changes. Key benefits of our framework are resource efficiency, flexibility, and low cost of operation. Minimizing the number and volume of parameter changes, the system reduces operational costs without sacrificing performance levels. The various training scenarios allow the agent to generalize effectively to unseen network conditions. Moreover, the stability of the PPO algorithm guarantees consistent performance without requiring frequent policy updates, again lowering system overhead. The framework is a scalable solution for latency-critical BRAN environments, nicely trading off performance demands with resource conservation through data-driven automation.

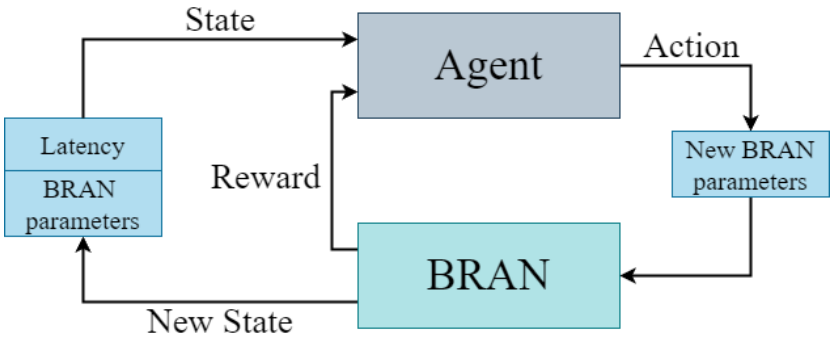
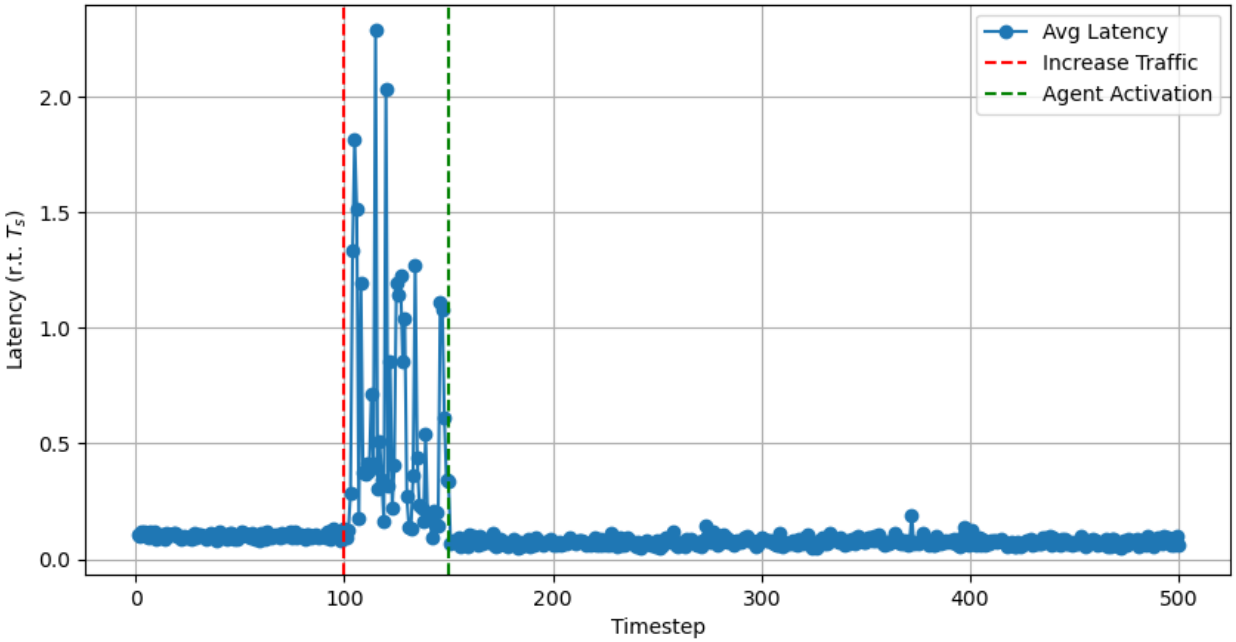
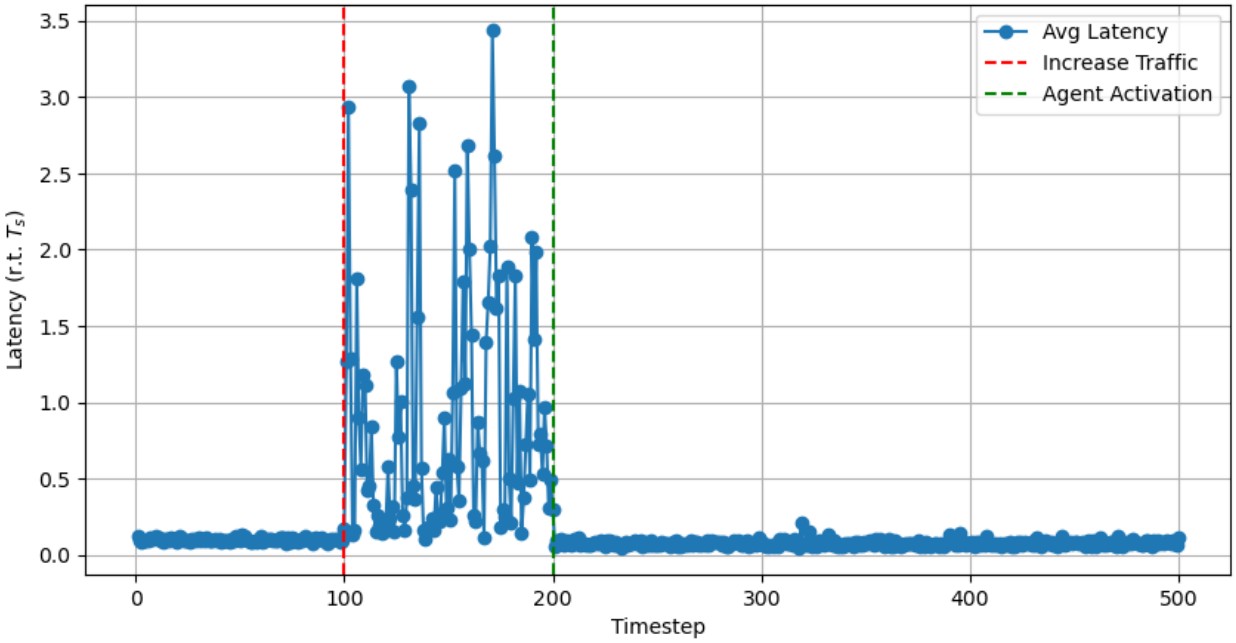


Figure 21. RL Framework

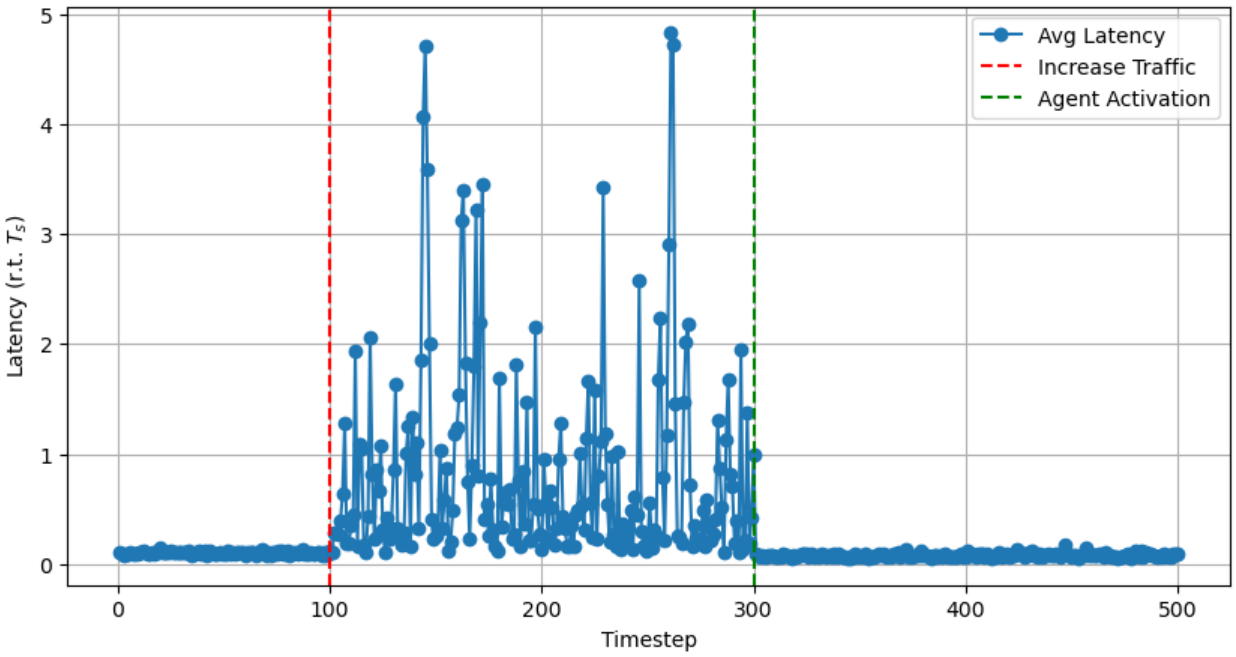
6.2. Performance Evaluation



(a) RL agent activation with a delay of 50 timesteps



(b) RL agent activation with a delay of 100 timesteps

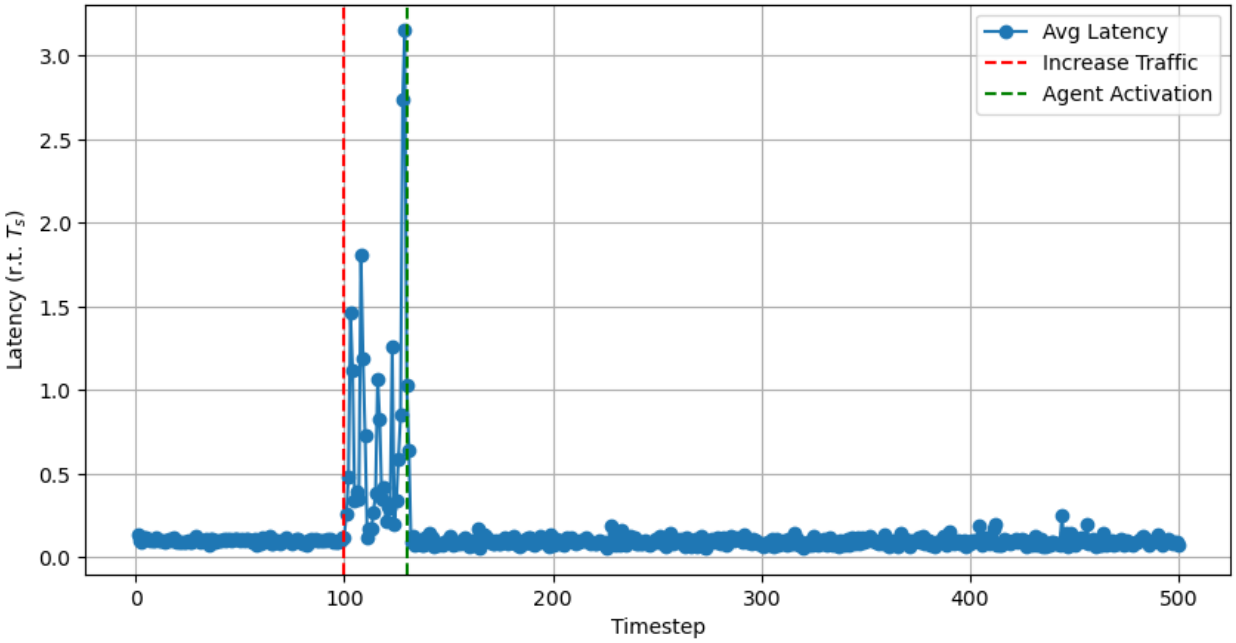


(c) RL agent activation with a delay of 200 timestep

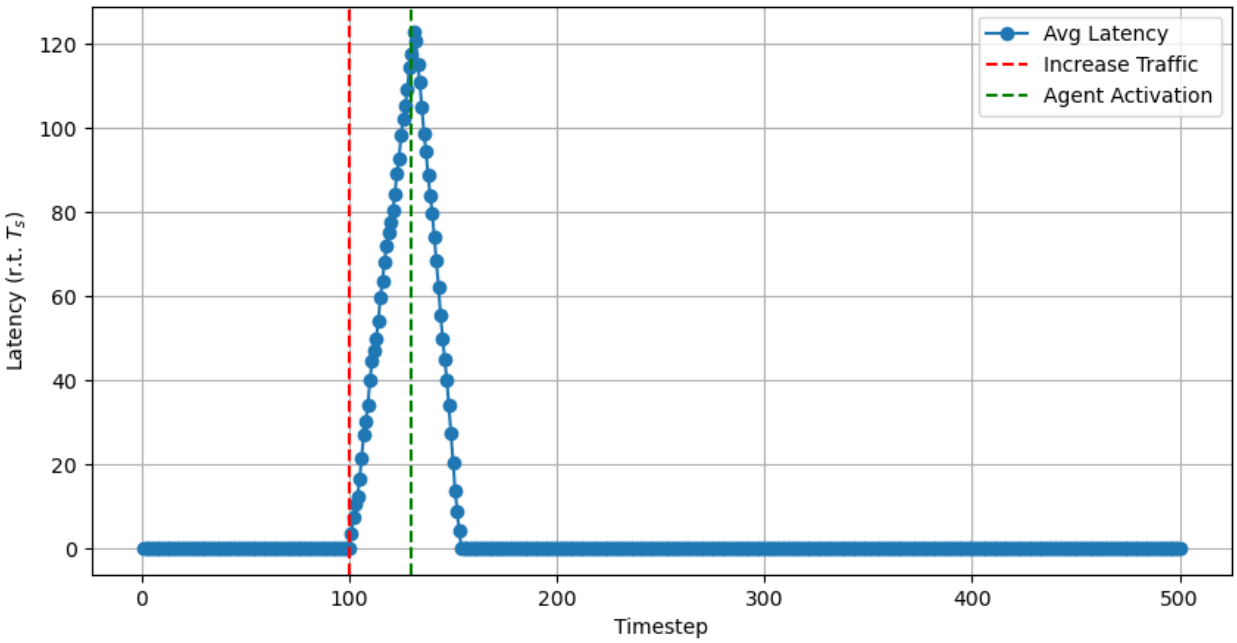
Figure 22. Latency over time for different RL agency activation delay scenarios

Figure 22 (a) depicts the effect of the RL on the average latency of the BRAN simulation. The x-axis represents the time unit in the simulation, and the y-axis represents the corresponding average latency measurements. Under normal operation, which is the time period between 0-100, the system shows a stable low-latency trend with an average. At timestep 100, indicated by a red dashed vertical line, an artificially caused traffic surge leads to significant performance degradation. This event is reflected in latency values that drastically peak at levels above 1 unit and show high variation between timesteps 100 and 150. After the activation of the pre-trained RL agent at time 150, as indicated by the green dashed vertical line, the system demonstrates a quick convergence to the performance levels. It is noteworthy that the reinforcement learning agent effectively optimizes network parameters by identifying the minimum sufficient configuration required to meet target performance metrics, thus conserving computational and network resources while still ensuring quality of service requirements. After the agent is activated, from timesteps 200 to 500, the network resumes stable performance with low latency and little variation. This demonstrates that the reinforcement learning-based adaptive control mechanism works well to mitigate network congestion and restore good performance under dynamic traffic conditions.

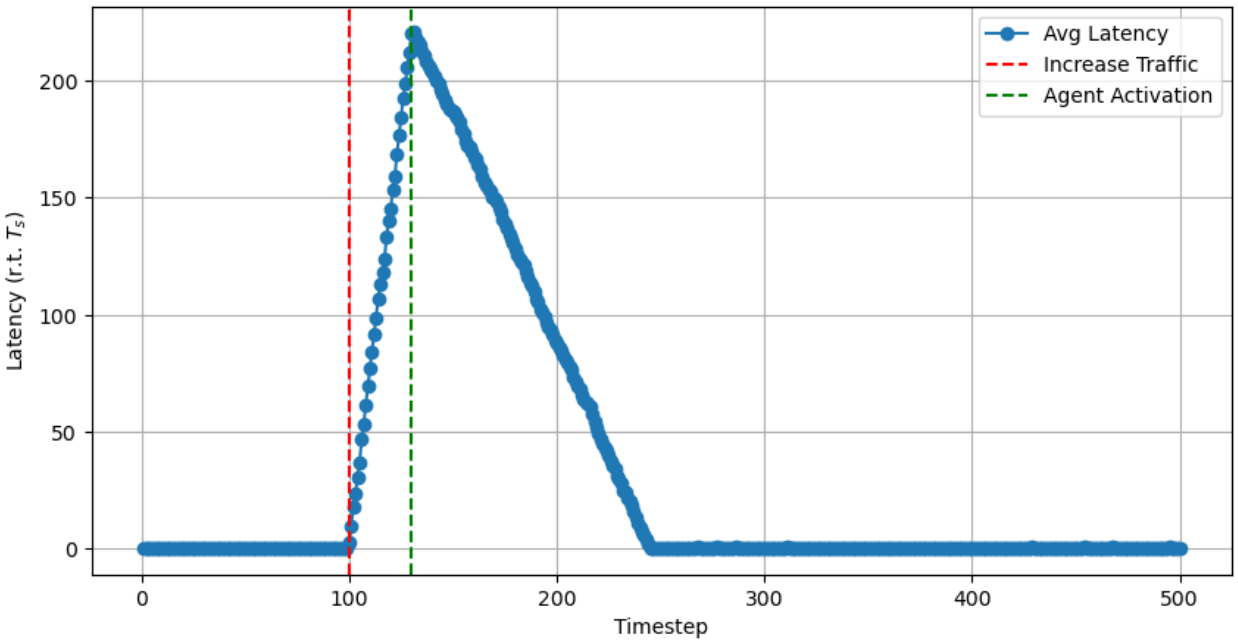
A comparison with the other two cases when the RL agent is called after 100 and 200 timesteps is highly insightful for intervention timing. As the delay in agent activation increases in duration, we observe a corresponding rise in peak latency values. For the 100-unit delay case Figure 22 (b), peak latency values increase to around 3.5 units, far above the 2.25 units observed for the 50-timestep delay case scenario. Even more starkly, in the 200-timestep delay case Figure 22 (c), the maximum latency approaches 4.75 units and experiences sustained high-amplitude oscillations throughout congestion time. This behavior clearly shows that longer pre-intervention delays cause more degradation, which then becomes progressively harder to reverse. The contrast between these three cases demonstrates the presence of a critical threshold beyond which network congestion may be irreversible or take much longer recovery periods. Although our reinforcement learning agent manages to stabilize the system in each of the three cases, the results emphasize the value of early detection and prompt response in sustaining network resilience.



(a) Mediocre traffic increase



(b) High traffic increase



(c) Very high traffic increase

Figure 23. Latency over time for multiple traffic scenarios

The three subplots of Figure 23 present the results of a BRAN simulation of the average latency vs timestep over different traffic rates, with annotations referencing traffic increases and RL agent interventions. Each plot presents average latency, where vertical red lines denote increments of the traffic rate and green lines are activations of the RL agent. In all cases, higher traffic (red line) provokes instantaneous spikes in latency, followed by the activation of the RL agent (green line) and stabilization afterward. At a traffic level of 100 Figure 23 (a), latency reaches approximately 3 latency units and then recovers quickly after intervention. In the 150-traffic case Figure 23 (b), latency increases sharply to approximately 120 units, an indication of moderate system stress; the agent mitigates this spike fairly well, though recovery is slower than in the 100-traffic case. In the maximum load Figure 23 (c), latency peaks at around 200-traffic rate, and remains elevated for a longer period, and with a higher peak value at 200 latency units, an indication of severe stress. Though recovery is very slow, the agent does recover to stability. These results demonstrate the RL agent's robustness in reducing latency across varying traffic loads. It is worth noting that, while recovery durations increase with traffic rates, suggesting operational challenges at high loads, the agent reliably returns to system stabilization, demonstrating its adaptability in dynamic environments. Finally, the prolonged recovery at the 200-traffic rate suggests potential scalability limits, emphasizing the need for complementary strategies in high-stress regimes.

7. Conclusion

The document "D2.2 – NANCY Experimental-Driven Modelling" presents a comprehensive theoretical framework for the B-RAN architecture, which aims to transform wireless networks by enabling service consumers to act as service providers. This deliverable explores three key NANCY usage scenarios: the fixed fronthaul network, advanced coverage expansion, and advanced connectivity of mobile nodes. A central lesson from the B-RAN modelling is that the proposed model, leveraging Markov chain theory, effectively captures the probabilistic transitions between different system states, offering valuable insights into its dynamic behaviour. The introduction of a HB-RAN model represents a novel approach for assessing the performance of coverage expansion scenarios by utilizing nested blockchains to enhance security and reliability.

Performance evaluation through analytical modelling and numerical results underscores the significant impact of parameters like the number of confirmations (N) and block capacity (k) on system latency, with generally higher k values leading to lower latency, particularly under high traffic conditions. Notably, the proposed B-RAN model demonstrates superiority over conventional models in accurately representing the temporal performance of complex B-RAN systems across diverse traffic conditions. Analysis of the HB-RAN model reveals that the secondary blockchain typically experiences higher latency compared to the primary blockchain, and its activity has a substantial influence on the overall system performance. Furthermore, the study identifies an inherent equilibrium between security and latency in B-RAN design, where enhancing security measures often results in increased latency, and vice versa.

In terms of attack modelling, the document concludes that while the risk of external attacks is lower in the permissioned blockchains used by B-RAN, insider threats and compromises of cryptographic material pose critical concerns. The document pinpoints 51% and Sybil attacks as the most impactful threats to B-RAN systems, integrating them into the theoretical modelling framework. The analysis shows that the probability of a successful 51% attack escalates with the attacker's mining power (β) and diminishes with a higher number of confirmations (N). Moreover, Sybil attacks, especially when combined with double-spending attempts, can significantly amplify an attacker's likelihood of success by disrupting communication among honest nodes.

The exploration of intelligent optimization using RL highlights its potential for dynamically adjusting blockchain parameters at runtime to maintain target latency levels while optimizing resource utilization. RL also demonstrates effectiveness in mitigating network congestion and restoring performance under dynamic traffic conditions. This approach facilitates a crucial trade-off between performance demands and resource conservation through data-driven automation.

Looking ahead, the document indicates several future directions, including further investigation into custom solutions for empowering B-RAN with HSMs to bolster security. The application of AI in blockchain

is identified as a key area for optimizing performance, security, and scalability, encompassing improvements in consensus mechanisms, resource utilization, and overall efficiency. The potential of AI for smarter fraud detection and risk assessment functionalities within B-RAN systems is also highlighted. Addressing the inherent trade-offs between security, performance, and decentralization in blockchain through AI-driven automatic optimization and enhanced governance processes is another crucial direction. Finally, the document suggests further examination of the scalability limits of the proposed B-RAN model under high traffic loads and the exploration of complementary strategies for high-stress regimes.

Bibliography

[1] A.-A. A. Boulogeorgos et al., “Artificial intelligence empowered multiple access for ultra reliable and low latency THz wireless networks,” 2022, arXiv:2208.08039.

[2] S. E. Trevlakis, A. A. A. Boulogeorgos, D. Pliatsios, J. Querol, K. Ntontin, and P. Sarigiannidis, “Localization as a key enabler of 6G wireless systems: A comprehensive survey and an outlook,” IEEE Open J. Commun. Soc., vol. 4, pp. 2733–2801, 2023..

[3] M. A. Rahman and M. S. Hossain, “A deep learning assisted software defined security architecture for 6G wireless networks: IIoT perspective,” IEEE Wireless Commun., vol. 29, no. 2, pp. 52–59, Apr. 2022.

[4] H. Cao, L. Yang, S. Garg, M. Alrashoud, and M. Guizani, “Softwarized resource allocation of tailored services with zero security trust in 6G networks,” IEEE Wireless Commun., vol. 31, no. 2, pp. 58–65, Apr. 2024.

[5] A. Al-Dulaimi, O. A. Dobre, and C.-L. I, Blockchains: Empowering Technologies and Industrial Applications (IEEE Series on Digital & Mobile Communication). Hoboken, NJ, USA: Wiley-IEEE Press, 2023.

[6] F. Tschorsch and B. Scheuermann, “Bitcoin and beyond: A technical survey on decentralized digital currencies,” IEEE Commun. Surveys Tuts., vol. 18, no. 3, pp. 2084–2123, 3rd Quart., 2016.

[7] H. Liu, P. Zhang, G. Pu, T. Yang, S. Maharjan, and Y. Zhang, “Blockchain empowered cooperative authentication with data traceability in vehicular edge computing,” IEEE Trans. Veh. Technol., vol. 69, no. 4, pp. 4221–4232, Apr. 2020.

[8] Q. Wang, D. Gao, C. H. Foh, H. Zhang, and V. C. M. Leung, “Decentralized CRL management for vehicular networks with permissioned blockchain,” IEEE Trans. Veh. Technol., vol. 71, no. 11, pp. 11408–11420, Nov. 2022.

[9] P. Wang, N. Xu, H. Zhang, W. Sun, and A. Benslimane, “Dynamic access control and trust management for blockchain-empowered IoT,” IEEE Internet Things J., vol. 9, no. 15, pp. 12997–13009, Aug. 2022.

[10] T. Salman, M. Zolanvari, A. Erbad, R. Jain, and M. Samaka, “Security services using blockchains: A state-of-the-art survey,” IEEE Commun. Surveys Tuts., vol. 21, no. 1, pp. 858–880, 1st Quart., 2019.

[11] H. Chen, H. Chen, W. Zhang, C. Yang, and H. Cui, “Research on marketing prediction model based on Markov prediction,” Wireless Commun. Mobile Comput., vol. 2021, no. 1, Jan. 2021, Art. no. 4535181.

[12] X. Yutong, “Applications of Markov chain in forecast,” J. Phys. Conf., vol. 1848, no. 1, Apr. 2021, Art. no. 12061, doi: 10.1088/1742-6596/1848/1/012061.

[13] H. Berthiaux and V. Mizonov, “Applications of Markov chains in particulate process engineering: A review,” Can. J. Chem. Eng., vol. 82, no. 6, pp. 1143–1168, 2004.

[14] L. Kiffer, R. Rajaraman, and A. Shelat, “A better method to Analyze blockchain consistency,” in Proc. ACM SIGSAC Conf. Comput. Commun. Security (CCS), 2018, pp. 729–744.

[15] R. Srivastava, “Mathematical assessment of blocks acceptance in blockchain using Markov model,” Int. J. Blockchains Cryptocurrencies, vol. 1, no. 1, pp. 42–53, 2019.

[16] V. Kontorovich, “Some aspects of blockchain-enabled radio access networks (B-RAN) modeling: Review and theoretical study,” J. Adv. Math. Comput. Sci., vol. 37, no. 7, pp. 44–60, Aug. 2022.

[17] Y. Li, B. Cao, L. Liang, D. Mao, and L. Zhang, “Block access control in wireless blockchain network: Design, modeling and analysis,” IEEE Trans. Veh. Technol., vol. 70, no. 9, pp. 9258–9272, Sep. 2021.

[18] J. Wang, X. Ling, Y. Le, Y. Huang, and X. You, “Blockchain-enabled wireless communications: a new paradigm towards 6G,” National Science Review, vol. 8, no. 9, Apr. 2021.

[19] X. Ling, J. Wang, T. Bouchoucha, B. Levy, and Z. Ding, “Blockchain radio access network (B-RAN): Towards decentralized secure radio access paradigm,” IEEE Access, vol. 7, pp. 9714–9723, 2019.

[20] X. Ling, Y. Le, J. Wang, Z. Ding, and X. Gao, “Practical modeling and analysis of blockchain radio access network,” IEEE Transactions on Communications, vol. 69, no. 2, pp. 1021–1037, Feb. 2021.

[21] T. Sachinidis, A.-A. A. Boulogeorgos, and P. Sarigiannidis, “Dual-hop blockchain radio access networks for advanced coverage expansion,” 10th Int. Conf. Modern Circuits Syst. Technol. (MOCAST), 2021, pp. 1–5.

[22] Q.-L. Li, J.-Y. Ma, and Y.-X. Chang, “Blockchain queue theory,” 7th Int. Conf. Comput. Data Soc. Netw. (CSoNet), 2018, pp. 25–40.

[23] F. Wilhelmi, S. Barrachina-Muñoz, and P. Dini, “End-to-end latency analysis and optimal block size of proof-of-work blockchain applications,” IEEE Commun. Lett., vol. 26, no. 10, pp. 2332–2335, Oct. 2022.

[24] Y. Li, B. Cao, L. Liang, L. Zhang, M. Peng and M. A. Imran, "A Block Access Control in Wireless Blockchain Networks," International Conference on UK-China Emerging Technologies (UCET), 2020, pp. 1-4.

[25] F. Wilhelmi and L. Giupponi, “Discrete-time analysis of wireless blockchain networks,” in Proc. IEEE 32nd Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC), 2021, pp. 1011–1017.

[26] P. Yang, L. Chen, H. Zhang, J. Yang, R. Wang, and Z. Li, “Joint optical and wireless resource allocation for cooperative transmission in C-RAN,” Sensors, vol. 21, no. 1, p. 217, 2021.

[27] P. Jiang, J. Fu, P. Zhu, J. Li, and X. You. “Joint precoding design and resource allocation for C-RAN wireless fronthaul systems.” 2023. [Online]. Available: https://arxiv.org/abs/2305.15926.

[28] Ö. Bulakci, A. B. Saleh, S. Redana, B. Raaf, and J. Hämäläinen. “Enhancing LTE-advanced relay deployments via relay cell extension.” 2011. [Online]. Available: https://arxiv.org/abs/1111.5810.

[29] I. Vilà, O. Sallent, and J. Pérez-Romero, “Relay-empowered beyond 5G radio access networks with edge computing capabilities,” Comput. Netw., vol. 243, Apr. 2024, Art. no. 110287.

[30] K. Tokarz, “A review on the vehicle to vehicle and vehicle to infrastructure communication,” in Proc. Man–Mach. Interact., 2020, pp. 44–52.

[31] R. Q. Malik, K. N. Ramli, Z. H. Kareem, M. I. Habelalmatee, and H. Abbas, “A review on vehicle-to-infrastructure communication system: Requirement and applications,” in Proc. 3rd Int. Conf. Eng. Technol. Appl. (IICETA), 2020, pp. 159–163.

[32] Q. Hua, K. Yu, Z. Wen, and T. Sato, “A novel base-station selection strategy for cellular vehicle-to-everything (C-V2X) communications,” Appl. Sci., vol. 9, no. 3, p. 556, 2019.

[33] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, and H. Wang, “Blockchain challenges and opportunities: A survey,” Int. J. Web Grid Services, vol. 14, p. 352, Oct. 2018.

[34] J. Ahn, E. Yi, and M. Kim, “Blockchain consensus mechanisms: A bibliometric analysis (2014–2024) using VOSviewer and R Bibliometrix,” Information, vol. 15, no. 10, p. 644, 2024.

[35] S. Zhou, K. Li, L. Xiao, J. Cai, W. Liang, and A. Castiglione, “A systematic review of consensus mechanisms in blockchain,” Mathematics, vol. 11, no. 10, p. 2248, 2023.

[36] R. B. Cooper, “Queueing theory,” Proc. ACM Conf., 1981, pp. 119–122.

[37] W. D. Kelton and A. M. Law, “The transient behavior of the M/M/s queue, with implications for steady-state simulation,” Oper. Res., vol. 33, no. 2, pp. 378–396, 1985.

[38] L. Kleinrock. “Queueing systems volume 1: Theory.” 2024. [Online]. Available: https://www.academia.edu/99469679/Queueing_ Systems_Volume_1_Theory.

[39] J. Wang, X. Ling, Y. Le, Y. Huang, and X. You, “Blockchain-enabled wireless communications: a new paradigm towards 6G,” National Science Review, vol. 8, no. 9, Apr. 2021.

[40] X. Ling, Y. Le, J. Wang, Z. Ding and X. Gao, "Practical Modeling and Analysis of Blockchain Radio Access Network," IEEE Trans. Commun. , vol. 69, no. 2, pp. 1021-1037, Feb. 2021.

[41] J. Wang, S. Wu, H. Liang, Y. Ding, and Y. Zhai, “Adaptive mining difficulty for blockchain to resist selfish mining attack,” Journal of Surveillance, Security and Safety, vol. 3, no. 4. OAE Publishing Inc., pp. 14–34, 2023.

[42] B. Pingle, A. Mairaj and A. Y. Javaid, "Real-World Man-in-the-Middle (MITM) Attack Implementation Using Open Source Tools for Instructional Use," 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 2018, pp. 019.

[43] J. Choi, B. Ahn, G. Bere, S. Ahmad, H. A. Mantooth and T. Kim, "Blockchain-Based Man-in-the-Middle (MITM) Attack Detection for Photovoltaic Systems," 2021 IEEE Design Methodologies Conference (DMC), Bath, United Kingdom, 2021, pp. 1-6.

[44] M. Z. A. Bhuiyan and J. Wu, "Collusion Attack Detection in Networked Systems," IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), 2016, pp. 286-293.

[45] “Hyperledger - Using a Hardware Security Module (HSM).” [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-2.5/hsm.html

[46] Cabrera Gutierrez, Antonio Javier & Castillo, E. & Escobar-Molero, Antonio & Álvarez-Bermejo, J. & Morales, Diego & Parrilla, Luis. (2022). Integration of Hardware Security Modules and Permissioned Blockchain in Industrial IoT Networks.

[47] Thales Group, "Hyperledger Fabric." [Online]. Available: https://www.thalesdocs.com/gphsm/integrations/guides/hyperledger_fabric/index.html

[48] H. Javaid et al., "Blockchain Machine: A Network-Attached Hardware Accelerator for Hyperledger Fabric," 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS), Bologna, Italy, 2022, pp. 258-268.

[49] P. Baguer, G. M. Yilma, E. Municio, G. Garcia-Aviles, A. Garcia-Saavedra, M. Liebsch, X.Costa-Pérez, “Attacking O-RAN Interfaces: Threat Modeling, Analysis and Practical Experimentation,” IEEE Open Journal of the Communications Society, vol. 5. Institute.

[50] P. Oktivasari, A. R. Zain, M. Agustin, A. Kurniawan, F. arbi Murad, and M. fabian Anshor, “Analysis of Effectiveness of Iptables on Web Server from Slowloris Attack,” 5th International Conference of Computer and Informatics Engineering (IC2IE), 2022, pp. 215-219.

[51] M. M. Alani, “Detection of Reconnaissance Attacks on IoT Devices Using Deep Neural Networks,” EAI/Springer Innovations in Communication and Computing. Springer International Publishing, pp. 9–27, Oct. 20, 2021.

[52] M. Uma and G. Padmavathi, “A survey on various cyber attacks and their classification,” International Journal of Network Security, vol.15, no.5, pp. 390-396, Sep. 2013.

[53] S. Roy, N. Sharmin, J. C. Acosta, C. Kiekintveld, and A. Laszka, “Survey and taxonomy of adversarial reconnaissance techniques,” ACM Computing Surveys, vol. 55, no. 6, pp. 1–38, 2022.

[54] M. F. Hyder and M. A. Ismail, “Securing control and data planes from reconnaissance attacks using distributed shadow controllers, reactive and proactive approaches,” IEEE Access, vol. 9, pp. 21 881–21 894, 2021.

[55] M. Conti, N. Dragoni, and V. Lesyk, “A Survey of Man In The Middle Attacks,” IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2027–2051, 2016.

[56] E. N. Amachaghi, M. Shojafar, C. H. Foh and K. Moessner, "A Survey for Intrusion Detection Systems in Open RAN," IEEE Access, vol. 12, pp. 88146-88173, 2024.

[57] W. Tiberti, E. Di Fina, A. Marotta and D. Cassioli, "Impact of Man-in-the-Middle Attacks to the O-RAN Inter-Controllers Interface," IEEE Future Networks World Forum (FNWF), 2022, pp. 367-372.

[58] O-RAN Work Group 11 (Security Working Group), "O-RAN Security Threat Modeling and Risk Assessment", 2023. [Online]. Available: https://orandownloadsweb.azurewebsites.net/download?id=554.

[59] J. R. Douceur, ”The sybil attack,” IPTPS 2002: Peer-to-Peer Systems, 2002, pp. 251–260.

[60] S. Nakamoto. “Bitcoin: A peer-to-peer electronic cash system.” Mar. 2009. [Online]. Available: https://metzdowd.com.

[61] S. Zhang and J. -H. Lee, "Double-Spending With a Sybil Attack in the Bitcoin Decentralized Network," in IEEE Transactions on Industrial Informatics, vol. 15, no. 10, pp. 5715-5722, Oct. 2019.

[62] Bhumichai, D.; Smiliotopoulos, C.; Benton, R.; Kambourakis, G.; Damopoulos, D. The Convergence of Artificial Intelligence and Blockchain: The State of Play and the Road Ahead. Information 2024, 15, 268. https://doi.org/10.3390/info15050268.

[63] Cao, L. Decentralized AI: Edge Intelligence and Smart Blockchain, Metaverse, Web3, and DeSci. IEEE Intell. Syst. 2022, 37, 6–19.

[64] T. N. Dinh and M. T. Thai, "AI and Blockchain: A Disruptive Integration," in Computer, vol. 51, no. 9, pp. 48-53, Sep. 2018.

[65] J. D. Harris and B. Waggoner, "Decentralized and Collaborative AI on Blockchain," IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 2019, pp. 368-375.