Manuel António
Valente Couto

Autenticação Baseada em Blockchain para NDNs

A Blockchain-Based Authentication for NDNs

PROPOSTA DE TESE

Universidade de Aveiro
2023

Manuel António Valente Couto

Autenticação Baseada em Blockchain para NDNs

A Blockchain-Based Authentication for NDNs

PROPOSTA DE TESE

"*The greatest challenge to any thinker is stating the problem in a way that will allow a solution*"

— Bertrand Russell

Universidade de Aveiro
2023

Manuel António Valente Couto

Autenticação Baseada em Blockchain para NDNs

A Blockchain-Based Authentication for NDNs

Proposta de Tese apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à conclusão da unidade curricular Proposta de Tese, condição necessária para obtenção do grau de Mestre em Engenharia de Computadores e Telemática , realizada sob a orientação científica do Doutor André Ventura da Cruz Marnôto Zúquete, auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor Carlos Roberto Senna, Investigador no Instituto de Telecomunicações

.

Texto Apoio financeiro do POCTI no âmbito do III Quadro Comunitário de Apoio.

Texto Apoio financeiro da FCT e do FSE no âmbito do III Quadro Comunitário de Apoio.

o júri / the jury

presidente / president

vogais / examiners committee

Prof. Doutor Alexandre Júlio Teixeira dos Santos

Professor Associado com Agregação do Departamento de Informática da Escola de Engenharia da Universidade do Minho (arguente)

Prof. Doutor André Ventura da Cruz Marnôto Zúquete

Professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro (orientador)

agradecimentos / acknowledgements

Agradeço primariamente aos meus pais e irmão por todo o apoio incondicional, desde o início do meu percurso escolar até à conclusão desta etapa. Agradeço também à minha namorada, que me motivou ao longo dos últimos meses e me ajudou a não perder o rumo, bem como a todos os meus colegas com quem partilhei ao longo dos últimos anos as mesmas aventuras, alegrias, tristezas, e com quem experienciei momentos únicos. Finalmente, agradeço também o apoio ao Prof. Dr. Zúquete e ao Dr. Carlos pela orientação e pelo seu estímulo à aprendizagem desta area. Um obrigado.

Keywords Blockchain, Named Data Networking, Cidades Inteligentes, Disseminação de Conteúdo, Internet das Coisas, Otimização de Rede, Desafios de Segurança

Resumo

A Information-Centric Networking (ICN) tem sido considerada a candidata mais promissora para superar as desvantagens das arquiteturas centradas em hosts quando aplicada a redes IoT. Esta dissertação abordou o desafio de garantir a segurança nas comunicações em ambientes de Named Data Networking, com foco principal na sua aplicabilidade no contexto de Cidades Inteligentes. Esta tese propõe um mecanismo de autenticação baseado em blockchain para melhorar a segurança nesse paradigma de comunicação emergente. O estudo, que avalia a eficácia do sistema, a sua escalabilidade e a sua resistência a stress, utiliza conjuntos de dados de mobilidade reais do ambiente de Cidades Inteligentes de Aveiro, conhecido como Aveiro Tech City Living Lab (ATCLL), com ênfase particular no seu desempenho no domínio das Cidades Inteligentes, e utiliza um simulador de NDN para avaliar a proposta de segurança baseada em blockchain, em cenários do mundo real. Com base nos resultados obtidos, demonstrando a viabilidade das soluções de segurança baseadas na tecnologia blockchain, esta pesquisa representa uma contribuição significativa para a segurança NDN em Cidades Inteligentes. A solução de segurança baseada em blockchain é capaz de aprimorar o ecossistema da NDN, impondo um mínimo de sobrecarga de segurança. As medidas de segurança apresentam uma influência decrescente à medida que o tamanho dos arquivos aumenta, proporcionando um equilíbrio entre a integridade dos dados e o desempenho do sistema. O mesmo ocorre ao falar de escalabilidade, pois a solução proposta conseguiu acomodar um número crescente de consumidores.

Keywords

Blockchain, Named Data Networking, Smart Cities, Content Dissemination, Internet of Things, Network Optimization, Security Challenges

Abstract

The Information-Centric Networking (ICN) has been considered the most promising candidate to overcome the drawbacks of host-centric architectures when applied to IoT networks. This dissertation addressed the callenge of securing communication in Named Data Networking environments, with a primary focus on its applicability to the context of Smart Cities. This thesis proposes a blockchain-based authentication mechanism to enhance security in this emmerging communication paradigm. The study, evaluating the system´s effectiveness, is scalability and stress resilience, employs real mobility datasets from Aveiro's Smart City environment, known as the Aveiro Tech City Living Lab (ATCLL), making a particular emphasis on its performance in the Smart City domain and uses an NDN simulator to assess the proposed blockchain-based security approach in real-world scenarios. Based on the obtain results, demonstrating the viability of security solutions based on blockchain technology, this research represents a significant contribution to securing NDN in Smart Cities. The blockchain-based security solution is capable of enhancing the NDN ecosystem while imposing minmal security overhead. Security measures exhibit a decreasing influence as file sizes increase, providing a balance between data integrity and system performance. The same thing happened when talking about scalability, as the proposed solution was able to accommodate an expanding number of consumers.

Contents

List of Figures

List of Tables

List of Algorithms

Glossary

NDN	Named Data Networking
ndnSIM	Named Data Networking Simulator
ARP	Address Resolution Protocol
IP	Internet Protocol
ATCLL	Aveiro Tech City Living Lab
ICN	Information-Centric Networking
DLT	Distributed Ledger Technology
API	Application Programming Interface
VANET	Veicular Ad Hoc Network
FIB	Forwarding Information Base
PIT	Pending Interest Table
CS	Content Store
RSU	Road-Side Unit
Mbps	Megabits per Second
Gbps	Gigabits per Second
NFD	Named Data Networking Forwarding Daemon
PPKD	publisher public key digest
DHT	Distributed Hash Table
ITS	Intelligent Transportation Systems
IoV	Internet of Vehicles
V2V	Vehicle-to-Vehicle
V2I	Vehicle-to-Infrastructure
V2X	Vehicle-to-Everything
DApps	Decentralized Apps
NFT	Non-fungible Token
PoET	Proof of Elapsed Time

CHAPTER 1

Introduction

Since the beginning of the Internet that the main approach when working with communication networks is based on the location of the data, it is embed in the protocols that support the interconnection between computers, such as Address Resolution Protocol (ARP) and Internet Protocol (IP), which rely on peer addresses. However, with the development of the Internet, new perspectives have emerged to complement this location-based approach. This led to the creation of NDN, that focus on obtaining data regardless of its location. However, this has created a challenge in ensuring the authenticity of the information, because the implicit relation of the data with its origin do not exist anymore. To address this, our goal is to guarantee a reliable origin for every piece of content, regardless of its delivery, in an NDN by 1) connecting producers to the correspondent produced data and 2) creating a list of reliable producers. The proposed solution will be based on a blockchain and evaluated using an NDN simulator [1] and real-world testing in Aveiro Tech City Living Lab (ATCLL) infrastructure [2].

For the development of the work foreseen for this thesis, three important aspects must be studied in depth: NDN environments, NDN security and distributed ledger technology. Accordingly, the next section is divided into these three themes.

1.1 Objectives

The main objectives of this work are:

- Overcoming NDN Security Challenges;
- Implement the suggested solution within the context of Smart Cities;
- Analyze the impact on Network Optimization;
- Contribute to NDN Security in Smart Cities.

1.2 Contributions

The main contributions of this work are:

- Adds blockchain technology-based security to NDN environments;
- Adds a simple transport layer for secure file dissemination across NDNs;
- It proposes an alternative for managing names and content in NDN with access control.;

1.3 Dissertation Structure

The remaining document is structured as follows:

- **Chapter 1 | Introduction**
- **Chapter 2 | State of the Art and Related Work** - Contextualization of the addressed themes.
- **Chapter 3 | System Architecture and Implementation** - Presents the system architecture and provides the implementation developed.
- **Chapter 4 | Results** - Describes all the tests and results performed and discusses the resulting values.
- **Chapter 5 | Conclusion and Future Work** - A conclusion of the work proposed in this document and a future work section which discusses possible improvements to the developed system and directions to follow to achieve such improvements.

CHAPTER 2

State of the Art and Related Work

As the proposed authentication method is for NDN environments, recent work in NDN and its architecture will be discussed in the Named Data Networking Section. Following, the NDN security solutions and their weaknesses will be covered. The Distributed Ledger Technology Section will examine how authentication and security can be improved using blockchain technology, as the proposed system uses it for decentralization, tamper-resistance, and transparency. Lastly, there is a section that discusses the tools and frameworks used to complete this project.

2.1 Named Data Networking

NDN is an innovative approach to building computer networks that aims to improve the current Internet Protocol (IP) in a number of ways. It is a part of a larger class of ICN, techniques that put distribution and labeling of data ahead of devices or places [3]. The way NDN and IP networks handle data routing is one of their main distinctions. NDN networks employ data names to direct packets to their destinations as opposed to IP networks, which use IP addresses [4]. NDN networks can also offer superior caching, security, and content delivery features.

In an NDN architecture, the Consumer requests Contents without knowing the providing host, and the communication follows a receiver-driven approach, *i.e.* as seen in (Figure 2.1) [5], the process of retrieving data in NDN starts with the consumer sending an Interest packet, the Interest packet is forwarded by routers until it reaches a producer with a matching data entry, then the data (Data packets) follows the reverse route/path of the request, traveling back to the consumer who requested it. The system is then responsible for mapping the requested data and its location. An NDN architecture is characterized by the adopted approach on each of the following key functionalities: naming and name resolution, routing and forwarding, and caching [6].

NDN nodes use three structures in the Forwarding process (Figure 2.2). The Forwarding Information Base (FIB) contains the names of a given Content, provided by the routing

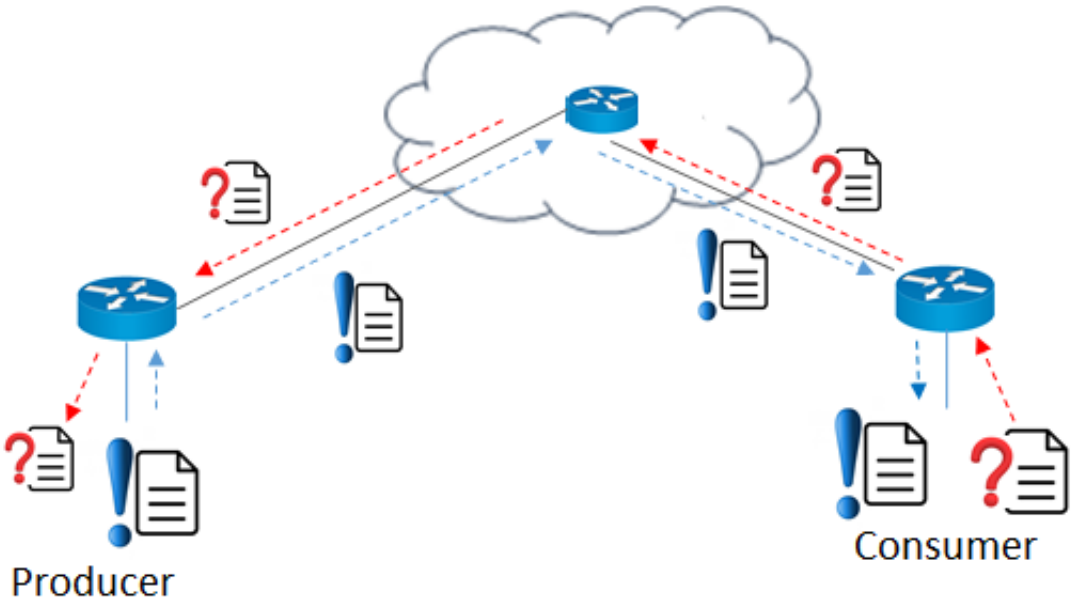


Figure 2.1: Consumer & Producer interaction[5].

protocol, and associates them with the interfaces on which they were received. In addition to the Forwarding Information Base (FIB), NDN nodes use two other tables: Pending Interest Table (PIT) and Content Store (CS)[5]. NDN routers use routing algorithms or a self-learning mechanism to update and announce their FIB entries. The forwarding strategy in NDN routers is stateful, meaning that the routers keep information about received requests until they are satisfied or timed out. The forwarding strategy forwards Interest packets according to the FIB entries, local measurements, or other per-name-space forwarding policies. NDN routers also have the ability to use multi-path forwarding to ensure priority and load balancing and avoid failed links. When a router receives an Interest packet, it first checks if the requested data can be satisfied from the local Content Store (CS). If the data exists in the local CS, the router sends back the Data packet to the source interface. If the CS does not hold the requested data, the router performs a lookup in the PIT to see if there are any similar pending requests, if an entry with a similar name exists, the router aggregates the incoming Interest packets requesting the same data. If no pending Interest with the same name exists, the router creates a new entry for this Interest packet and sends it to its upstream neighbor(s) according to its forwarding strategy.

According to Zhang et al. [4], NDN can be assume a hierarchical structured naming scheme, similar to URLs. For example, a video produced by Aveiro may have the name /aveiro/videos/demo.mpg, where the "/" symbol delineates name components in the text representation. This hierarchical structure allows applications to represent the context and relationships of data elements, as well as provide name aggregation, for instance "/aveiro" could correspond to an autonomous system originating the video. As seen, the delivery of content is made more effective by NDN networks since they use data names to identify content. Distributing content according to its name rather than its location makes it simpler to identify

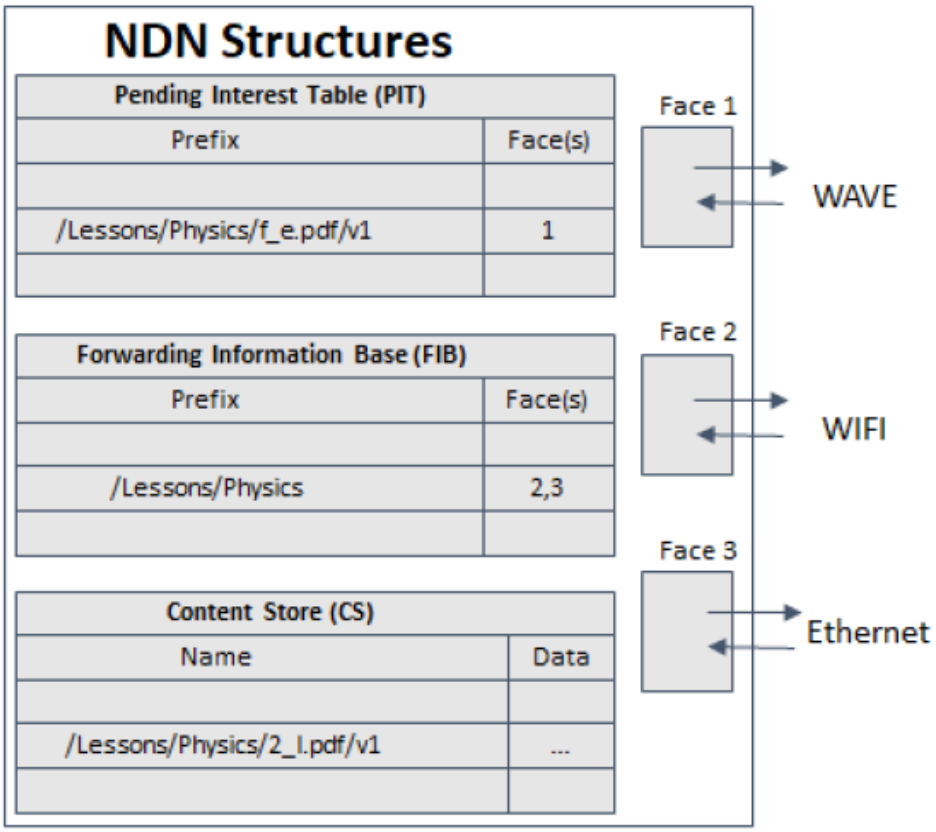


Figure 2.2: Node's control structures [5].

and retrieve, this makes it possible for quicker information distribution and more effective use of network resources. In-network caching is another feature that NDN networks can provide. This feature enables the dissemination of content from many sources, enhancing network resilience.

NDN networks can also provide better support for mobility and multi-homing, and can be more easily integrated with other ICN technologies[7]. The effectiveness of NDN is another crucial feature. The Interests enable more effective content caching and routing. By eliminating the need for data to pass via intermediary nodes, NDN's in-network caching also lowers latency and improves network scalability. By enabling direct connection between the data producer and consumer without the need for complicated addressing and routing systems, the data-centric design of NDN also enables more efficient handling of real-time and high-bandwidth applications.

2.1.1 Security Aspects in NDN

Security is a vital issue in NDN environments, especially for applications such as the dissemination of content in vehicular networks. Several studies have been conducted to address security concerns in NDN, covering various aspects of the technology.

NDN proposals often offer a number of security and privacy features, according to the authors, including trust, data origin authentication, peer entity authentication, data integrity, authorization and access control, accountability, availability, data confidentiality, traffic flow

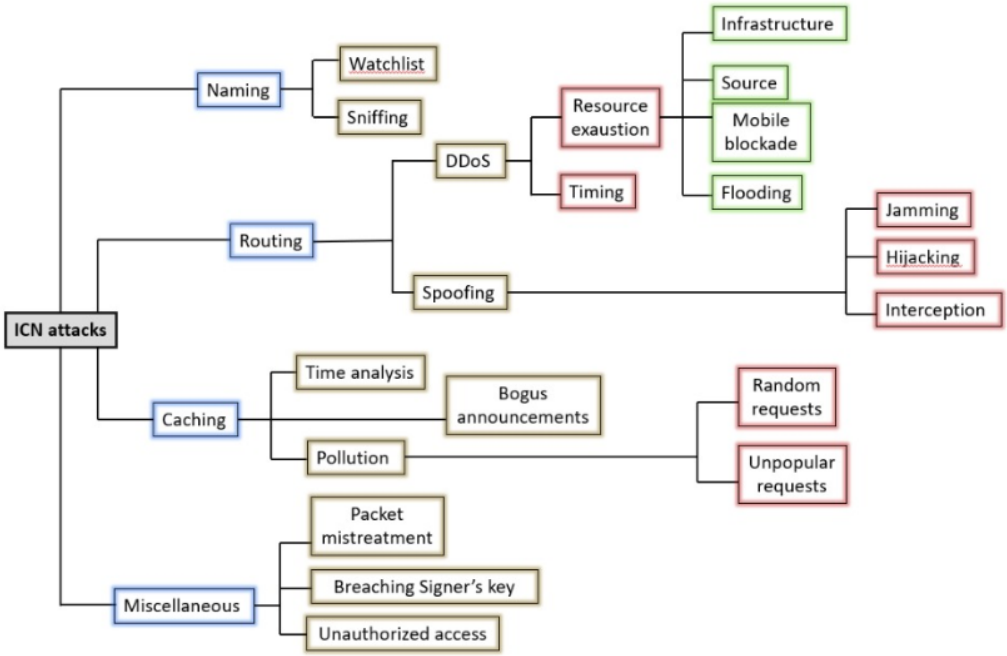


Figure 2.3: ICN attacks referred by [9].

confidentiality, and anonymous communication [8]. However, there is a need for additional research and development to satisfy these objectives because the current NDN architectures fall short of meeting all of these specifications. Hurali et al. [8] also analyzes the security dangers posed by NDN, such as naming, routing, and caching concerns. Watchlist, sniffer, scanning, and protocol attacks which can affect naming, routing, and caching are common threats to NDN. The network security model's Confidentiality, Integrity, and Availability (CIA) objectives may be impacted by these dangers. The authors mention a number of methods, including digital signatures, certificates, cryptographic hashing algorithms, path consent systems, and access control lists, that can be utilized to lessen these risks. These methods are employed to guarantee data accountability, peer entity authentication, data integrity, and data origin authentication. The paper includes a thorough analysis of ICN's current state and future application possibilities, including NDN. With a focus on security and privacy, it draws attention to the difficulties and unanswered research concerns in the field of NDN. In order to reduce the security threats that NDN presents, the authors also stress the necessity of a thorough set of security and privacy features in NDN as well as more research and development.

[9] and [10] provide a comprehensive review of the security threats and countermeasures in NDN, including attacks on the naming and routing layers, content authenticity and integrity, and privacy. They discuss various techniques and mechanisms that have been proposed to address these threats, such as cryptographic signatures, access control, and anonymity. Having this in count, Figure 2.3 illustrates the types of attacks that can occur in this area.

NDN is a viable replacement technology for TCP/IP communication networks that can offer affordable and dependable communication in a highly mobile setting. However, content

poisoning, a vulnerability in NDN that could pose a network hazard, exists. When an attacker injects erroneous information into the network's caches, clients are tricked into accessing outdated or harmful material.

In order to solve this issue, earlier studies presented name-key binding-based network layer techniques, where the producer informs routers of the bindings of names and key values. These methods increase the risk of dynamic content poisoning too, as attackers can edit or create bindings by pretending to be the producer. In order to address this issue, [11] proposes a consumer-oriented, two-phased, lightweight security scheme that includes end-to-end authentication and a packet-level name-key query mechanism. Particularly, the issue of impersonation is resolved by the name-key bindings being validated by an additional consumer verification. The data packet's content is hashed and then encrypted by the producer to create the signature. The producer is identified via the publisher public key digest (PPKD), which is contained in the packet's metadata. The consumer decrypts the signature and matches the decrypted hash with the received content digest after receiving the data packet. The consumer reissues interest with the PPKD in the exclude field to reject all packets with the same PPKD if the comparison is unsuccessful. However, because of the line speed limit and heterogeneous trust architecture, which restrict routers from doing signature verification, this security feature increases the possibility of content poisoning. Researchers came up with a name-key binding network layer approach, where the key stands for PPKD and the content digest, to address this issue. So that routers may receive the key values in the bindings and compare them with the key values in the incoming data packets, the producer specifically tells routers of the bindings of names and key values in advance, i.e., PPKD or content digest. The content of data packets is hashed by routers when they receive them, and the results are then compared to the content digest in previously acquired name-key bindings. Hashing the received content, which has less overhead than decrypting the signature, is the major method used to implement the comparison of the key values. As a result, routers may compare the key values at line speed to effectively reduce content poisoning. The authors also suggest a brand-new trust model to aid routers in identifying and disconnecting from hostile nodes.

To ensure data verification and security, some approaches stand out: signature-based verification, hash-based verification, Bloom filter, authentication information payload, distributed hash table and Merkle trees. In this set of good options, we chose to deepen the studies in two of them, distributed hash table and Merkle trees, whose characteristics that guided the choice are described below.

Distributed Hash Table (DHT) [12] is a distributed data structure used to store key-value pairs across a network of nodes, despite that more advantages were found in Merkle tree. Merkle tree provide a efficient and robust way to verify the integrity of large amounts of data, and is more efficient in terms of space because it only stores the root hash of the data, making them space efficient compared to DHTs, which store all the key-value pairs, they can be easily scaled to handle large amounts of data, as the tree can be constructed by diving data into smaller units and combining them into larger units, for other hand Merkle trees require a complex mathematical structure and they may be more difficult to implement and maintain

compared to DHTs which are simpler and more straightforward to implement.

Merkle trees are a data structure that can be used to verify the integrity of information transmitted over a network. They were first proposed by Ralph Merkle in [13] and have since been used in various applications, including blockchains [14]. In NDN, Merkle trees can be used to ensure the reliability and security of data transmitted over the network. In order to build a Merkle tree, data must first be divided into small units, known as leaf nodes, and then these units must be continuously combined into bigger units using a process known as hashing. The end result is a binary tree that has an exclusive root hash that condenses all of the data in the tree. A different root hash will arise from any changes to the data, hence this root hash can be used to check the accuracy of the data in the tree. In NDN, because of the use of names instead of locations, Merkle trees can be used to summarize the data in a way that can be efficiently verified by a recipient, ensuring that the data has not been tampered with during transmission. This provides a higher level of security for the data transmitted in NDN and ensures that it is reliable and trustworthy. A Merkle Tree enables incremental data validation as opposed to conventional approaches, which demand that the complete data set be communicated before it can be validated. This enables any problems to be found and fixed before the complete data set has been received by allowing the data to be immediately reviewed for mistakes and inconsistencies as it is transferred. This makes the Merkle Tree a useful tool since it allows for real-time data checking without the need to wait for the entire transmission to be finished. The use of Merkle trees in NDN can also improve the scalability of data storage and retrieval. With large amounts of data being transmitted over the network, summarizing the data in a Merkle tree can reduce the amount of data that needs to be transmitted, making the network more efficient and scalable. To verify a piece of data in NDN, a recipient can simply follow the path of hashes up the Merkle tree to the root, comparing each hash along the way to ensure that their copy of the data matches the original[15].

NDN networks use data names for efficient and secure content retrieval. The names are utilized for packet routing and caching, making it easier to search and access content, as well as maximizing network resources. Furthermore, by basing security on data names instead of locations, NDN networks enhance the protection against potential attacks and data tampering. Digital signatures are employed to verify the authenticity of the data, ensuring its integrity during transmission.

2.2 Distributed ledger technology

Distributed Ledger Technology (DLT) is being utilized more frequently in a range of industries for dependable and secure communication. One such field is NDN environments, where DLT can be quite useful in verifying the accuracy and integrity of the data being exchanged. The installation of blockchain technology, as well as its potential applications in NDN contexts, will be the primary focus of this paper's study of DLT.

Blockchain, originally announced as a component of the Bitcoin system [16], keeps track of transactions via a distributed, decentralized ledger. A network of validators, or nodes, that

work together to validate and log transactions on the blockchain, maintains this ledger.

The main benefit of blockchain is that it makes it possible to perform transactions securely and transparently without the need for a centralized authority. Blockchain is a cutting-edge transactional system that eliminates the need for a centralized authority. It makes use of a distributed ledger that is decentralized and records transactions. This indicates that a network of computers, known as nodes, that cooperate to validate and record transactions, rather than a single central organization, such as a bank, maintains the ledger of transactions. Due to the fact that every node on the network has a copy of the same ledger and any changes to the ledger must be consensus-based, this guarantees the security and transparency of the transactions. The blockchain gets its name from the way that its transactions are organized into blocks and then connected in a chain-like structure. Each block consists of several transactions as well as a special code known as a "hash" that connects it to the block before it. As a result, an unbreakable chain of blocks is formed, rendering it impossible to alter or tamper with earlier transactions without the network's consent. Utilizing advanced encryption methods further improves the security of the blockchain. For instance, digital signatures, which are distinctive codes produced by the user's private key, are used to verify each transaction on the blockchain. This ensures that only the person in control of the related private key may approve the transaction. In conclusion, blockchain technology enables secure and transparent transactions without the need for a centralized authority by utilizing a decentralized, distributed ledger system and cutting-edge encryption algorithms. Because of this, it is a desirable technology for several sectors, including finance, logistics, and real estate.

The perspective of the blockchain governance model is a crucial aspect to consider when implementing a blockchain solution. There are three main models: public, consortium, and private. A public blockchain is open to anyone, allowing for maximum decentralization and transparency [17]. A consortium blockchain involves multiple parties from different organizations, balancing decentralization with control [18]. A private blockchain is closed within an organization, providing complete control for the organization but sacrificing decentralization [19].

Each model has its own advantages and disadvantages, and the choice will depend on the specific use case and the goals of the implementation. It is crucial to carefully consider the governance model in order to ensure the success and sustainability of the blockchain solution [20]. The economics of blockchain governance has also been extensively studied, with research exploring the incentives for users to participate in blockchain networks and the role of governance mechanisms in shaping these incentives [21].

Recent research have shown how effective DLT can be in NDN settings. For instance, in [22], a reporting system in a hypothetical smart city is built on the foundation of blockchain technology. Citizens can report on actual events and vote on already-reported events through a mechanism called Dignitas, using their reputation to support another person's claim. By examining the reputation a report has behind it, this approach helps to distinguish between accurate and false information. The system is run by a group of previously reputable authorities in the setting of a city, such as the fire and police departments, and the solution suggested in

this thesis does not call for a centralized authority.

A privacy-aware decentralized and personalized reputation system called PrivBox is presented in [23] with the intention of being used in online marketplaces. In this study, a verified decentralized reputation system is created using blockchain technology as the foundational architecture. PrivBox is a decentralized reputation system for online marketplaces that protects anonymity. It enables customers to voice their opinions about merchants and service providers without violating their privacy. The system ensures that feedback ratings remain private and within a predetermined range by using homomorphic encryption and non-interactive zero-knowledge proof. Additionally, it enables users and service providers to independently confirm the generated information without the need for a reliable third party. The decentralized aspect of the system is made possible by blockchain technology, which enables a distributed network of nodes to collectively store and authenticate the reputation ratings. PrivBox uses blockchain technology to make sure that the reputation scores are reliable and can be trusted by all parties. PrivBox's overall goal is to give customers a safe and confidential way to assess the reliability of service providers in online marketplaces.

Song et al.[24], is one noteworthy example. The authors suggest a method for creating a global distributed storage structure for content and producer information as well as a name searching service for content consumers using blockchain technology, more especially Ethereum smart contracts. The technique builds a reliable knowledge base for content and producers using smart contracts on the Ethereum blockchain. This enables users to quickly inspect the data stored in the smart contracts to confirm the legitimacy and integrity of the data packets they are seeking.To guarantee that the data recorded in the smart contract is legitimate and unaltered, the smart contracts can include digital signatures, hashes, or any other cryptographic techniques. By guaranteeing the validity and integrity of the data packets, this increases the security of the content retrieval procedure. The mechanism also offers content consumers name resolution and content retrieval services, which boosts the effectiveness of content retrieval. A prototype deployment is used by the authors to assess the cost of storage and consumption in smart contracts as well as the security of the mechanism. The outcomes demonstrate how safe and useful the suggested technique is.

Other works that investigate the application of DLT in NDN contexts include [25], that provides a blockchain-based next-generation reputation system, and [26], that suggests a vehicle ad-hoc network reputation system.

In this thesis, a mechanism for blockchain-based authentication for NDN environments will be developed, and we will use Intelligent Transportation Systems (ITS) scenarios to test the developed solution. Part of this scenario is the Internet of Vehicles (IoV) paradigm. IoV paradigm describes how connected vehicles and the Internet may exchange data and communicate with each other, with other connected vehicles, with road infrastructures, and with the Internet. This can be achieved through various types of communications Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Everything (V2X) communications , as the names suggest. It aims to increase customer safety, comfort, and efficiency by tying vehicles and other components of the transportation system together

through communication technologies.

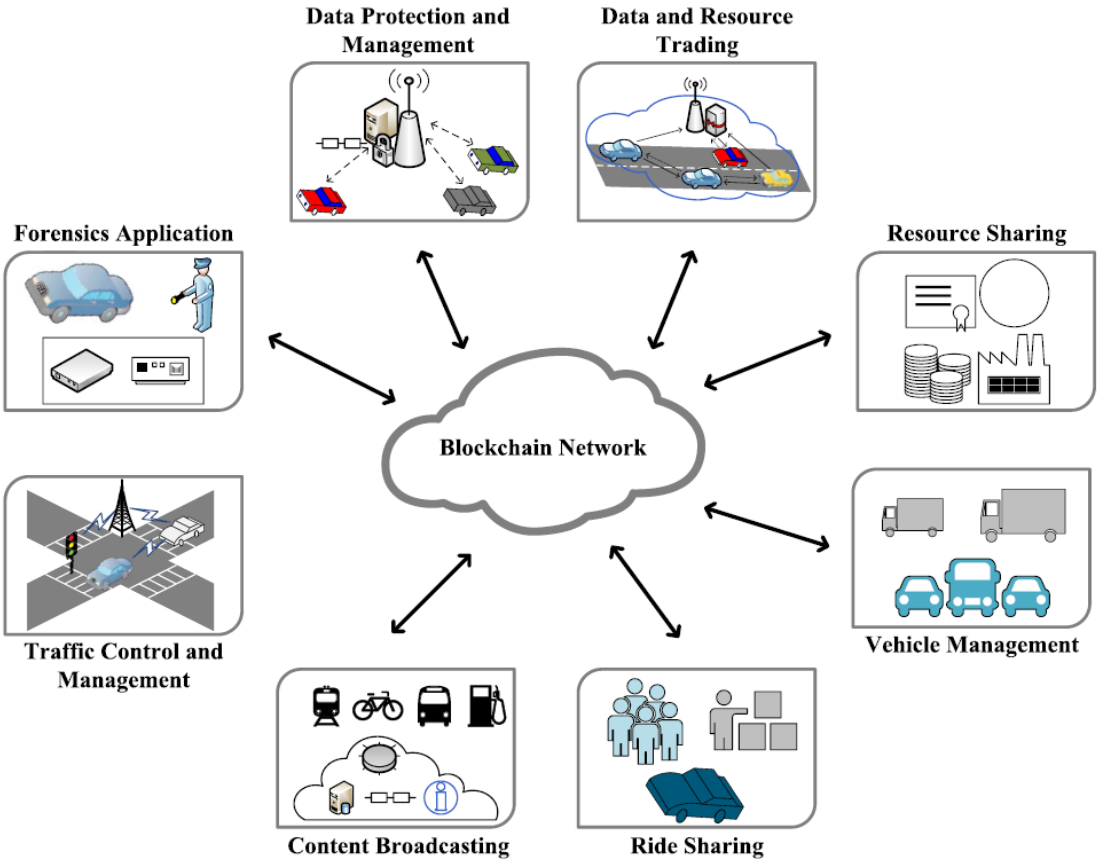


Figure 2.4: Blockchain & IoV [27].

The IoV uses advanced technologies and services to improve transportation systems and reduce emissions, as well as to improve the driving experience and reduce the frequency of traffic accidents. In Figure (2.4) [27] is presented a number of blockchain-assisted IoV application scenarios such as data protection and management, data and resource trading, resource sharing, vehicle management, ride sharing, content broadcasting, traffic control and management and forensics application.

Blockchain technology can assist the mentioned scenarios in various ways. For data protection and management, it provides secure and immutable data storage through its decentralized network and cryptographic algorithms, making it an ideal solution for sensitive data protection. In the case of data and resource trading, blockchain enables the creation of smart contracts to facilitate, verify and enforce trades of data or resources, ensuring transparency and fairness in transactions. The decentralized nature of blockchain allows for secure and trustless resource sharing between parties, such as sharing of bandwidth, storage, and computing power. For vehicle management, blockchain can store vehicle data such as maintenance records, history, and usage patterns, improving transparency and efficiency in management. In ride sharing, blockchain can improve the service by providing secure and transparent payment systems, reducing fraudulent activities, and improving trust between drivers and passengers. Content broadcasting can be facilitated by blockchain by tracking and

distributing digital content, ensuring content creators are paid for their work and consumers can access desired content. Blockchain technology can be used to optimize traffic flow, reduce congestion, and improve road safety through collecting and analyzing traffic data. For forensics application, blockchain can store tamper-proof records of digital evidence, providing a secure chain of custody and improving the integrity of forensic evidence. Some of this cases can be tested in the NDN simulator and the real city laboratory in Aveiro to provide a comprehensive assessment of its applicability in real-world settings the proposed solution will utilize the existing NDN platform for the communications infrastructure of ATCLL Aveiro and will be evaluated using ATCLL real mobility datasets [2].

2.2.1 Tools and Frameworks

A crucial stage in the development process is choosing the best blockchain framework for the project, which requires a thorough analysis of the special qualities and functionalities of the many possibilities.

During the process of determining which framework would be best for developing our blockchain, we looked into a number of well-known options, including as Remix, Embark, MetaMask, and ThirdWeb, all of which had distinctive advantages and characteristics.

Remix

Remix is a popular open-source programming environment and integrated development tool for Ethereum smart contracts and Decentralized Apps (DApps). It is web-based and provides an intuitive user interface for developing, assessing, and putting Ethereum blockchain smart contracts into practice. With features like integrated Solidity compiler, debugging, and code highlighting, Remix is a helpful tool for Ethereum developers.

While Remix is excellent for smart contract development, it is not a comprehensive blockchain framework and may require additional tools and platforms for full-stack **dapp!** (**dapp!**) development.

Embark

Embark is an open-source framework intended to make the creation of DApps on different blockchain networks easier. It provides an extensive collection of tools for **dapp!** frontend integration, testing, and smart contract creation. Additionally, Embark offers an Ethereum local development environment that makes it simple for developers to test and debug their DApps. It is especially well-suited for DApps projects that are Ethereum-focused, as it has features like plugin support, automated deployments, and decentralized storage.

ThirdWeb

ThirdWeb is a blockchain framework designed to transform the process of creating and implementing online apps. It centers on the notion of a "third web" that combines the advantages of the decentralized and conventional webs. ThirdWeb offers the infrastructure and resources needed to develop apps that take use of blockchain technology while preserving a recognizable and easy-to-use online experience, a sort of click and go. It imagines a time

where developers can create safe, decentralized, and privacy-preserving programs, and users have control over their data and online interactions. This framework is primarily used for mid-level Non-fungible Token (NFT)s, which was not the focus of this project.

MetaMask

MetaMask is a browser extension and mobile application that serves as a digital wallet and gateway to blockchain networks, primarily Ethereum. It enables users to manage their cryptocurrency assets, interact with DApps, and securely store private keys.

Developers frequently integrate MetaMask into their DApps to give customers a safe and practical method to communicate with blockchain apps straight from their web browsers. It is essential to the Ethereum ecosystem and acts as a link between users and DApps. However MetaMask is not a complete framework for blockchain programming, so it is out of the choices.

2.2.2 Hyperledger Sawtooth

Hyperledger Sawtooth is a robust and versatile enterprise blockchain platform designed for the development, deployment, and operation of distributed ledgers. It is part of an "umbrella" project called Hyperledger Project, this project is a creation of the Linux Foundation Project, and has members like IBM, Intel and Cisco [28]. Rooted in a philosophy that prioritizes ledger distribution and smart contract security, Hyperledger Sawtooth has gained traction as a reliable solution for various use cases within the enterprise realm. In this section, is approach the key features and aspects that make Hyperledger Sawtooth a compelling choice for this work in terms of blockchain development, while also addressing some of its potential drawbacks.

Pros of Hyperledger Sawtooth

Modularity and Flexibility: Hyperledger Sawtooth's hallmark feature is its modular architecture. This architectural choice allows developers to finely tailor and extend the platform to meet their precise needs. Such flexibility renders it a versatile and adaptable solution across a broad spectrum of use cases, without necessitating extensive, time-consuming customizations.

Distributed Ledger: Hyperledger Sawtooth places a pronounced emphasis on the distribution of ledgers across its network. This approach not only enhances transparency but also fortifies resilience and security. As a result, it becomes a pertinent choice for applications where multiple parties necessitate access to the same ledger, all the while maintaining unwavering trust in the integrity of the system.

Smart Contract Safety: Ensuring the security and reliability of smart contracts is paramount in enterprise blockchain applications. Hyperledger Sawtooth's design underscores this need. It accomplishes this by implementing a secure execution environment, effectively mitigating vulnerabilities that might compromise the overall integrity of the blockchain.

Proof of Elapsed Time (PoET) Consensus Algorithm: Hyperledger Sawtooth's considerable strength is its utilization of the PoET consensus algorithm. PoET is uniquely engineered to address the challenges posed by large, distributed validator populations while concurrently

minimizing resource consumption. What sets it apart is its non-reliance on specialized hardware, simplifying deployment and contributing to reduced operational costs.

Interoperability: Hyperledger Sawtooth, as part of the broader Hyperledger project, benefits from a commitment to open standards and interoperability. This attribute eases integration with other Hyperledger frameworks and third-party systems, ensuring compatibility with an organization's existing enterprise infrastructure.

Permissioned Blockchain: Hyperledger Sawtooth is particularly well-suited for permissioned blockchain networks. In situations where controlled access and privacy are paramount, such as in enterprise environments, Sawtooth excels. It provides organizations with the means to exert control over network participants and access to sensitive data.

Cons of Hyperledger Sawtooth

Learning Curve: The modular nature of Hyperledger Sawtooth, while advantageous, can potentially introduce a steep learning curve for developers unfamiliar with the platform. Understanding the intricate components and their interactions may pose a challenge.

Resource Consumption: Although PoET is designed to be resource-efficient, larger networks may still demand significant computing resources. This can exert a notable impact on the operational costs of maintaining and running a Sawtooth blockchain, particularly for smaller organizations.

Limited Public Network Use: Hyperledger Sawtooth is primarily tailored for enterprise use cases and may not be the most suitable choice for projects that seek to establish fully public and decentralized networks, akin to cryptocurrencies like Bitcoin or Ethereum.

In summary, Hyperledger Sawtooth emerges as a compelling choice for the development of enterprise blockchain applications. Its advantages include modularity, a commitment to ledger distribution, smart contract safety, the PoET consensus algorithm, interoperability, and support for permissioned networks, It was also taking in count the fact that it is Open-Source and has a big community behind, as analyzed in Luis Silva et al. [29]. However, it is essential to recognize that its adoption may necessitate overcoming a learning curve, addressing resource considerations in larger deployments, and understanding that it is best suited for private or consortium blockchain use cases, as opposed to fully public networks. The decision to choose Sawtooth as the foundation for a blockchain project ultimately hinges on specific needs, objectives, and the willingness to invest in mastering its capabilities.

Our goal was to develop a solution that could handle the security concerns in NDN, using our Blockchain approach we can guarantee trust, data origin authentication and data integrity just by the transparency and tamper-proof of the DLT, where the data is public and immutable, once in the blockchain data cant be deleted. In our blockchain we circulate a manifest for each registered file making this solution robust.

2.2.3 SHA256

Hashing is used to verify the integrity of secure messages and files. With sha256 a message is converted in a 256 bits hash, this hash cant be reversed to the original form. The hash code

of a secure file can be posted publicly and users who have the file can confirm they have an authentic version.

SHA-256 was created by the National Security Agency in 2001 as a successor to SHA-1 (https://www.n-able.com/blog/sha-256-encryption ([30])).

2.2.4 Next.js

Next.js is a new emerging technology for building full-stack web applications, developed by Vercel.

As said in Next.js documentation[1] [31] - "Next.js is a React framework that gives you building blocks to create web applications.
By framework, we mean Next.js handles the tooling and configuration needed for React, and provides additional structure, features, and optimizations for your application.
You can use React to build your UI, then incrementally adopt Next.js features to solve common application requirements such as routing, data fetching, integrations - all while improving the developer and end-user experience."

Combining the useful with the pleasant was possible to learn a new emerging technology while developing a fast web application.

[1] https://nextjs.org/learn-pages-router/foundations/about-nextjs/what-is-nextjs

CHAPTER 3

System Architecture and Implementation

In this chapter is pretended to present the proposed system architecture, explain some decisions that were made in order to successfully deploy the Blockchain-Based Authentication for NDN in this scenario. This chapter also explain the reasoning and behaviours some entities and components must have present in any type of implementation.

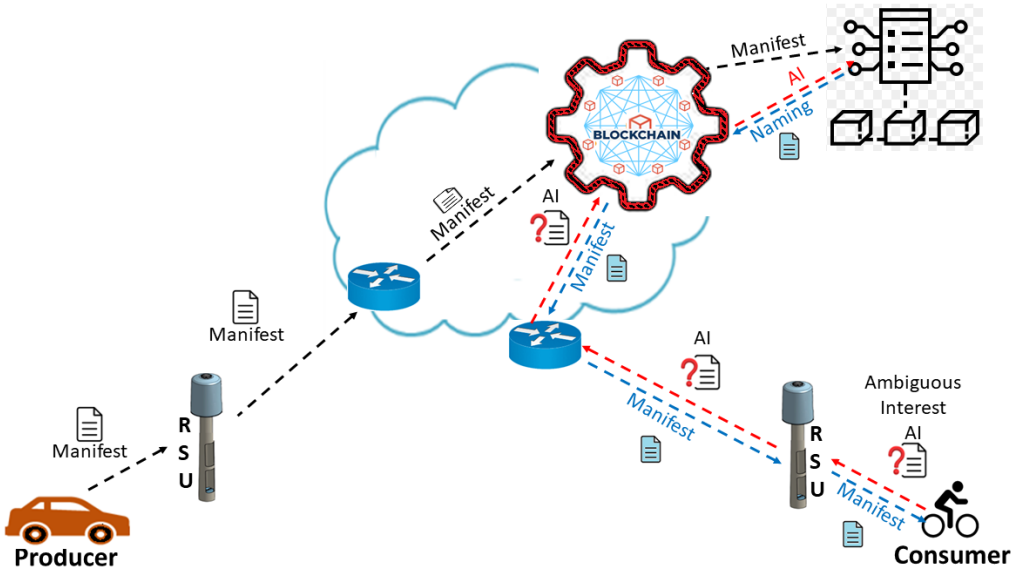


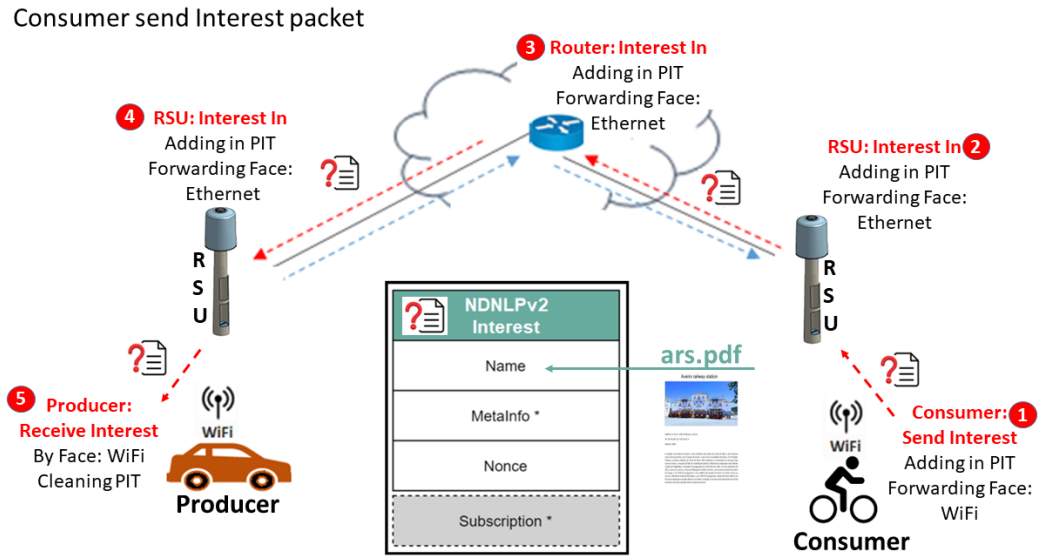
Figure 3.1: Security NDN Architecture

Introducing figure 3.1 that illustrates a architecture for a Veicular Ad Hoc Network (VANET) scenario, using our proposal, the first step falls over a reliable producer and his responsibility to upload the manifest of their available files into the blockchain. This is the first step in our network.

Next, a consumer who wants some sort of data searches for it in the blockchain. The blockchain retrieves the right name to the consumer's interest, and then, with the right name for the interest, the consumer, requests and receives the designated manifest.

The consumer sends the interest to the network, and it is passed through the nodes until it reaches the producer that has the data. After receiving the interest, the producer sends the data in reverse path.

This data is sent by chunks, and the consumer keeps validating every chunk in the process through hashes and ultimately with the Merkle tree.



L. Gameiro, C. Senna, and M. Luís. ndniot-fc: Iot devices as first-class traffic in name data networks. Future Internet, 12, 2020.

Figure 3.2: Send Interest Packet, source: [32]

Shown in Figure 3.2 is the process of sending the interest packet.

The consumer starts by inquiring about the available files in the blockchain. The blockchain responds with options, and if one of the options is appreciated, the consumer specify the request and inquiry the blockchain again, receiving the manifest from the desired file.

Having the manifest, the consumer can proceed asking the file to the producer in the NDN, when a consumer broadcasts an interest packet, the Road-Side Unit (RSU) receives it and checks if the requested data is already in its CS. If it is, the data packet is sent. If it is not, the RSU forwards the interest packet and updates its FIB.

The RSU checks its Pending Interest Table (PIT) for an existing entry with the same content name. If it exists, it aggregates incoming interests. If there is no matching entry, a new entry is created in the PIT.

When a node with the requested data is finally found, the data packet is sent in reverse path, following PIT entries. Each node/RSU along the path stores the data in its CS and checks its PIT entries. If there are pending interests for the data, they are forwarded to satisfy them and their PIT entries are removed.

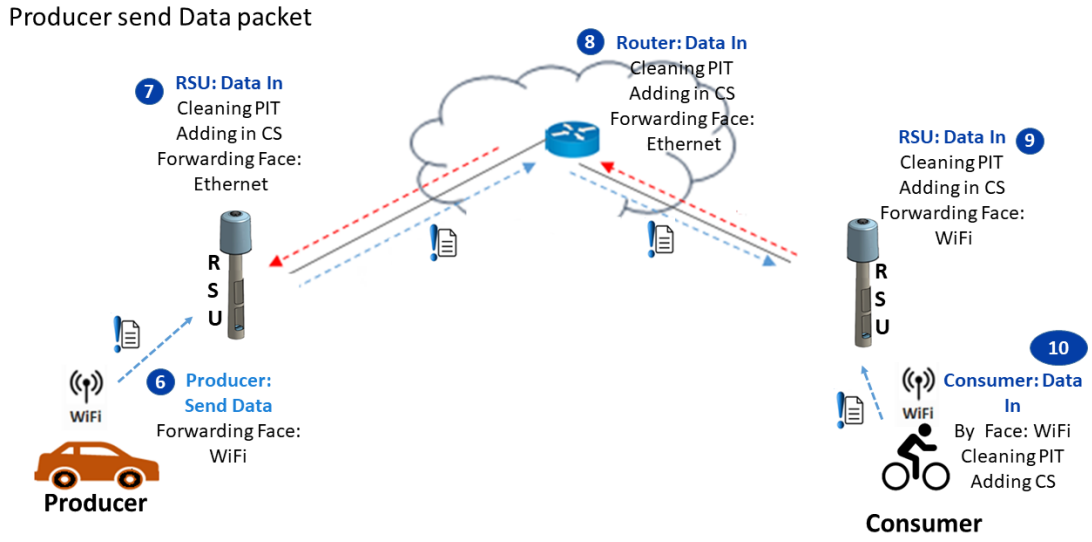


Figure 3.3: Send Data Packet, source: [32]

Once is found the node with the requested data, the producer begins a cycle where the consumer sends packets of interests, each corresponding to a chunk specified in the manifest.

Simultaneously, the producer responds by sending the data packet containing the desired chunk. When this data packet is being sent, it will clean the PIT entries of the nodes that it goes past.

3.1 Named Data Networking Simulator - ndnSIM

The well-known NS-3 network simulator is expanded to accommodate NDN protocols and concepts by the open-source simulation framework ndnSIM.

ndnSIM is an open-source C++ simulator for large-scale experimentation, it is widely used by the research community for the developments on NDN and NDN-based architectures. It has some enthusiastic features that stand out. The packet format is according to the original NDN packet format. It uses basic NDN primitives from *ndn - cxx* library (NDN C++ library with eXperimental eXtensions) [1]. It uses the source code of Named Data Networking Forwarding Daemon (NFD) for the forwarding and management, which is a network forwarder that manages, implements and develops abreast with the NDN protocol, it is considered a core component (as described in [33]). NFD is present in the main structures such as CS, PIT and FIB.

The integration with NFD and *ndn - cxx* library guarantees that simulations can be as closely matched to real-world situations as feasible and that they can be recreated in real-world settings without requiring major modifications. Moreover, real-world implementations can directly employ the forwarding strategies that ndnSIM has implemented.

[1] https://github.com/named-data/ndn-cxx

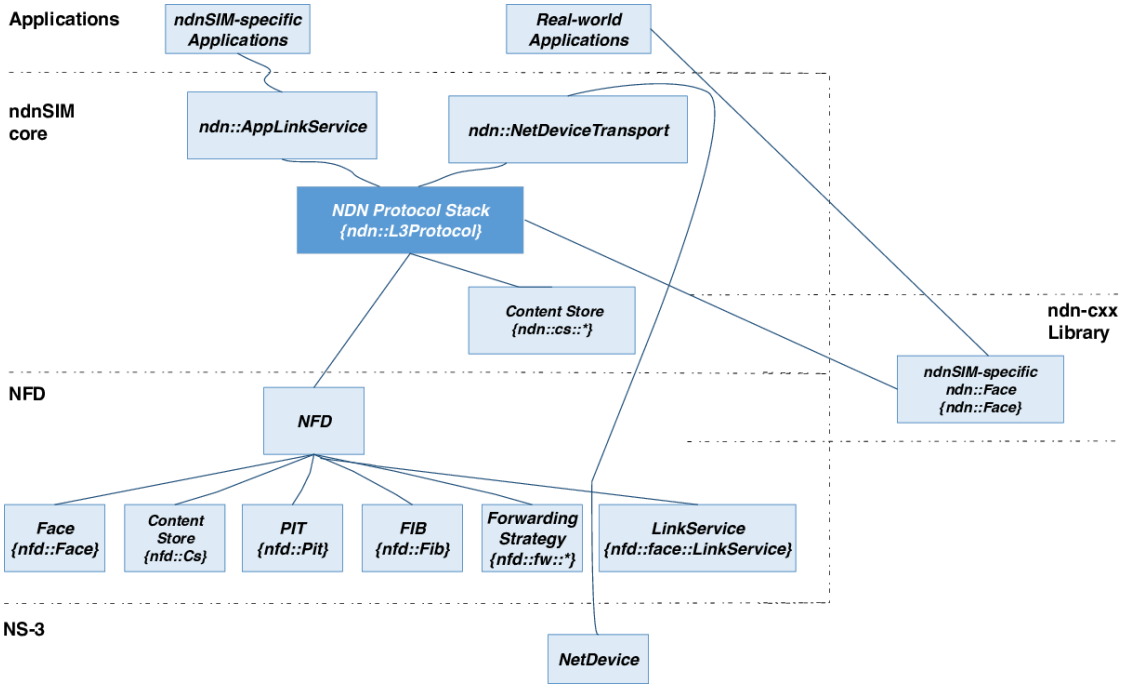


Figure 3.4: ndnSIM components structure. Source: ndnSIM[34]

3.1.1 Components

In the following section, there is a brief explanation of ndnSIM essential elements. This will give light on the inner workings and functionalities of the NDN architecture 3.4 and offer a thorough grasp of the key building pieces behind it.

Table 3.1: ndnSIM Components

Component	Description
NS-3	Since the simulation platform is based on the NS-3 platform, it uses the NS-3 base components. NetDevice provides the network layer to a device interface, enabling communication between devices through an NS-3 channel. The simulator uses it for ndnSIM-specific transport, which implements the NDN paradigm.
NFD	NDN Platform core component that implements base components such as Faces, CS, PIT, FIB, among others. These are also known as internal structures.
Internal Structures of NFD	- Face (nfd::Face): Implements communication abstraction, used by lower-level transport mechanisms. - Content Store (nfd::Cs): A cache of Data packets with limited capacity. - Pending Interest Table (nfd::Pit): Used to track incoming Interest packets. - Forwarding Information Base (nfd::Fib): Keeps information about available paths to each Content. - Forwarding Strategy (nfd::fw::*): Implements the forwarding strategy.

Table 3.1 – continued from previous page

Component	Description
	- LinkService (nfd::face::LinkService): A link service abstraction of a Face. - Forwarder (nfd::Forwarder): Considered the core component of NFD.
ndnSIM Core	- NDN Protocol Stack (ndn::L3Protocol): Responsible for initializing and deploying the NDN stack on each node. - AppLinkService (ndn::AppLinkService): Implements the ndnSIM-specific transport interface. - NetDeviceTransport (ndn::NetDeviceTransport): Implements the ndnSIM-specific transport interface, allowing communication between nodes through this layer.
Content Store	Refers to the module that integrates the old implementation of the Content Store in the simulator (not considered in this work).
Applications	ndnSIM allows the deployment of specific applications on each node, including Consumers and Producers, as well as limited integration of real-world applications written with the *ndn-cxx* library.

3.1.2 Forwarding Strategies

It is possible to modify the forwarding strategy that is being used in simulations. Table 3.2 represent the available options.

Table 3.2: ndnSIM Forwarding Strategies

Strategy	Description
Best-route	Decides to forward Interest packets to the least cost interface.
Multicast	Decides to forward Interest packets through all interfaces.
NCC	Uses a re-implementation of CCNx 0.7.2 default strategy.
Client Control Strategy	Based on a consumer application's local decision as to which interface will send Interest packets.

In this project was opted to use Best-route forwarding strategy.

3.1.3 Developed Consumer

To achieve this projects goals, it was necessary to modify the default ndnSIM Consumer. This new Consumer, initiates a search for the desired data by talking with the blockchain. Once it has the correct filename, the next step involves retrieving the manifest from the blockchain. With all of these set up, the Consumer proceeds to send interests to the network. These interests are subsequently answered by the producer. In 1 there is a pseudo-code representation of the Consumer.

Algorithm 1: Consumer Relevant Code

Data: File Name
Result: Search the desired data in the Blockchain
1 `searchData` ()
Data: filename, data buffer
Result: Save the manifest
2 `handle_save_manifest` ()
Data: File Name
Result: Request the manifest to the Blockchain
3 `handle_manifest_request` ()
Result: Save each chunk, make the Security Verifications, and create the File
4 `PrintReceivedFileContent` ()
Result: Send Interests
5 `SendPacket` ()
Result: Handle the Received Data
6 `OnData` ()

searchData()

This function is the one called to search for the desired data reference in the blockchain.

handle_save_manifest()

This function duty is to complement handle_manifest_request() and save the manifest in the Consumer side.

handle_manifest_request()

This function serves to request the manifest file, from the name obtained in searchData().

PrintReceivedFileContent()

This function aims to verify and save each chunk, so the File can be created. It also creates the file, but only after verifying the Merkle tree.

SendPacket()

This function is used to send Interest Packets to the Producer.

OnData()

This function is called every time new Data arrives.

3.1.4 Developed Producer

It was also necessary to adapt the default ndnSIM Producer. This Producer populates the FIB table and its is responsibility to handle the interests and return the desired data.

Algorithm 2: Producer Relevant Code

Result: Start the Producer, Populates FIB

```
1 StartApplication ()
```

Result: Handle Interests, send the Data

```
2 OnInterest ()
```

StartApplication()

This function is called to start the Producer App, this is here the FIB is populated.

OnInterest()

This functions is where the Producer handle the received interests, and return the desired data.

3.2 Sawtooth Architecture Overview

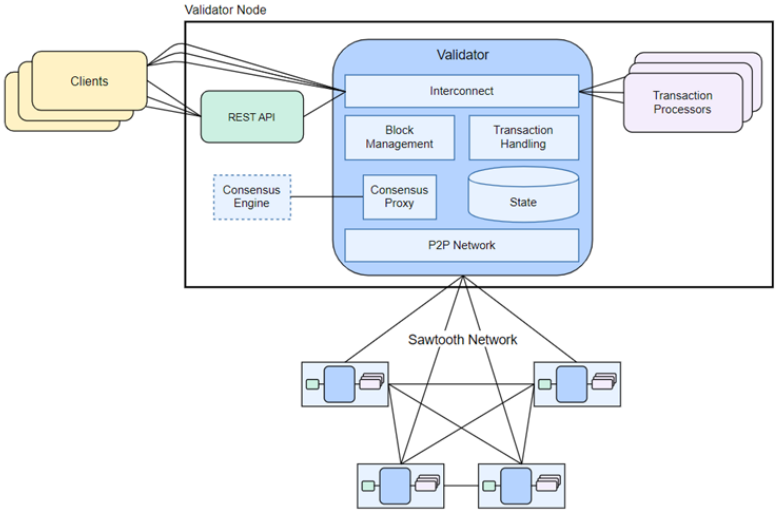


Figure 3.5: Sawtooth Architecture Overview [35]

Figure 3.5 represents a Sawtooth Network Deployment, it has several interconnected nodes, each of these nodes incorporate smaller elements. The validator is the mandatory component, it is present in all the deployments and its responsibilities are:

1. Permission - Check the batch signing key against the allowed transactor permissions
2. Signature - Check for integrity of the data
3. Structure - Check structural composition of batches: duplicate transactions, extra transactions, etc.

The other components function on "plug-and-play" manner, as illustrated in 3.5 the transaction processors, performs the business logic of the system and the consensus engine allows different consensus algorithms to be used. Sawtooth ecosystem features modularity in his components,

this is a vantage that allows a fine design while developing applications because every part of the system is build in a modular way, allowing a much easier control.

3.3 Security Solution

In the context of NDN, one of the primary challenges is ensuring the security of data and communication. The approach that we choose to overcome this obstacle is based on a Blockchain-Based Authentication mechanism for NDN. In this section, we will elaborate on our security solution, which leverages Merkle Trees, hashs, certificates, and a manifest in the blockchain to enhance the security of NDN.

3.3.1 Blockchain

NDN focus on obtaining data regardless of its location. However, this has created a challenge in ensuring the authenticity of the information, because the implicit relation of the data with its origin do not exist anymore. To address this, our goal is to guarantee a reliable origin for every piece of content, regardless of its delivery, in an NDN. To this we need to ensure the reliable of the producers and the received data. To achieve this goal, we start looking for a solution based on blockchain, which is a new emerge technology too.

Our idea is based on the integration of a manifest in the blockchain, this manifest (as exemplified in 3.3.4) contains a defined number of fields addressed in 3.3.4 that can help verifying the integrity and authenticity of the data.

The Blockchain provides secure and immutable data storage, making it an ideal solution for sensitive data protection. In the case of data and resource trading, blockchain enables the creation of smart contracts to facilitate, verify and enforce trades of data or resources, ensuring transparency and fairness in transactions however the perspective of the blockchain governance model is a crucial aspect to consider when implementing this solution because we need reliable producers, to achieve this we set that only authorized producers can upload data to the blockchain but data accessibility remains open to all consumers ensuring transparency in the network.

3.3.2 Producers Register

In order to register a producer on a Sawtooth blockchain, cryptographic key pairs associated to the producer's identity must be created. Its used the command "sawtooth keygen". This command is provided by the Sawtooth framework. It is used to generate cryptographic key pairs for producers. Its supposed to provide the producer's name or identification as an argument within this format sawtooth keygen "Producer_Name". Upon running the "sawtooth keygen" command, two cryptographic keys—a public key and a private key—are generated for the designated producer. Following the keys are generated, the producer may be identified in transactions using the public key, and they can digitally sign transactions using the private key. Through this procedure, transactions are guaranteed to be safely associated with their producer and validated by other users on the blockchain network.

3.3.3 Endpoint Specification

Table 3.3: Blockchain API Endpoints

Endpoint	Description	Arguments
`POST /send`	Manage the file manifest publishing process.	`"customer_name," "data"`
`GET /showdata`	This endpoint is crucial for enabling Consumers to efficiently filter data on the blockchain and access their desired information.	`"filter_string"` (optional)
`GET /getdata`	This endpoint allows producers to retrieve their specific records stored within the blockchain.	None

POST /send

This Application Programming Interface (API)'s main purpose is to manage the file manifest publishing process. This API, is mainly used by the producers, it is designed to receive two arguments: "customer_name," which verify the eligibility of the data producer to transmit information to the blockchain, and "data," which encapsulates the actual manifest data, as implied by its name.

When the API endpoint is called, this two arguments go trough a rigorous sequence of steps that are necessary so the transaction can become valid in the blockchain.

GET /showdata

This endpoint is essential to allowing Consumers to effectively filter data on the blockchain and retrieve the information that they want, in this case the manifest that they want. It has an optional argument called "filter_string," which is used to restrict the data selection when it is provided. So if you don't set a "filter_string", the endpoint will obtain the entire dataset that is kept on the blockchain.

Customers may now more easily identify and retrieve particular data from the blockchain with this simplified endpoint, which also offers flexibility by means of an optional filtering mechanism.

GET /getdata

This endpoint facilitates the retrieval of data linked to the requesting producer, enabling each producer to ascertain their specific records stored within the blockchain. Notably, this endpoint does not require any arguments for its execution.

User Interface

As an additional feature, a Next.js application was developed, aiming to ease the Producer life and ending to be a good alternative when comes to debugging the project, this application

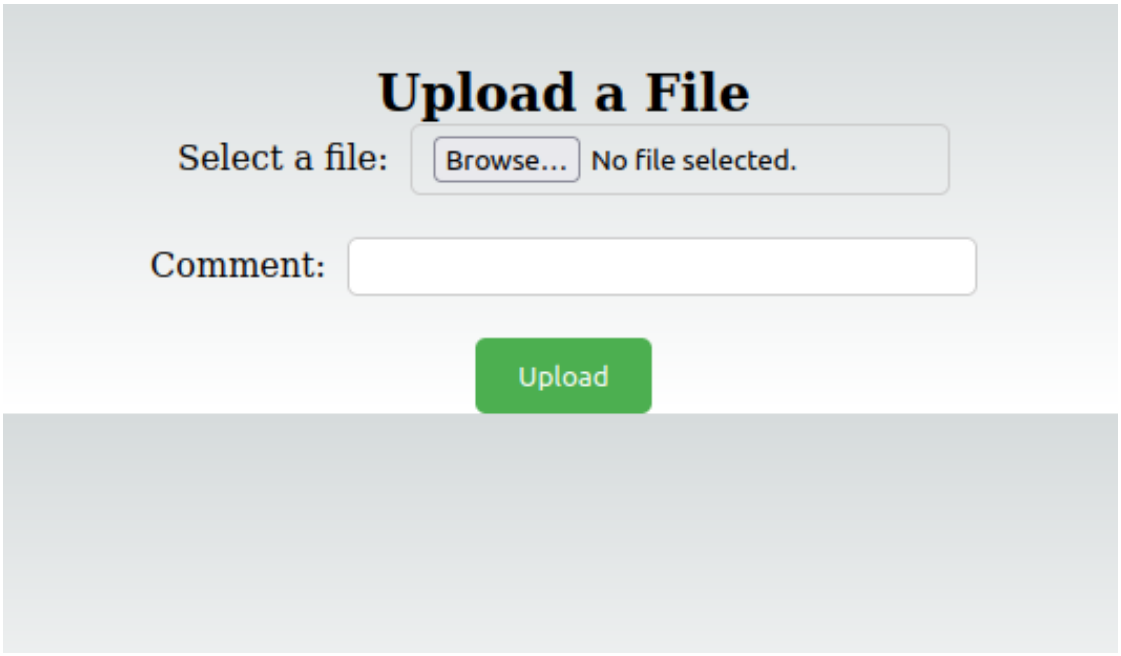


Figure 3.6: Producer Upload Page

```
ted to: c26e26ad96918b937d7574773483d60e558f68829d7d96874c80b417d0216ddf07216cd0
217cfd49794fe427aa78e4bcf8e79932931b0bbb4fbe29bfe8ad7718 (block_num:10, state:b1
c400e1d53ef708759c760ac74c73539915ae185aa6449d3ae4334cd39da2ee, previous_block_i
d:b5f55d81297d7410efa481ec1c94b7fa429abffacdadb09e310df842de74fc9711c62ae78c2bad
a7827ecb50a8f373929d75ebdc1a299bfa11311091d89f34ca)
validator              | [2023-10-29 23:38:41.762 INFO     publisher] Now buildin
g on top of block: c26e26ad96918b937d7574773483d60e558f68829d7d96874c80b417d0216
ddf07216cd0217cfd49794fe427aa78e4bcf8e79932931b0bbb4fbe29bfe8ad7718 (block_num:1
0, state:b1c400e1d53ef708759c760ac74c73539915ae185aa6449d3ae4334cd39da2ee, previ
ous_block_id:b5f55d81297d7410efa481ec1c94b7fa429abffacdadb09e310df842de74fc9711c
62ae78c2bada7827ecb50a8f373929d75ebdc1a299bfa11311091d89f34ca)
validator              | [2023-10-29 23:38:41.767 DEBUG    chain] Verify descenda
nt blocks: c26e26ad96918b937d7574773483d60e558f68829d7d96874c80b417d0216ddf07216
cd0217cfd49794fe427aa78e4bcf8e79932931b0bbb4fbe29bfe8ad7718 (block_num:10, state
:b1c400e1d53ef708759c760ac74c73539915ae185aa6449d3ae4334cd39da2ee, previous_bloc
k_id:b5f55d81297d7410efa481ec1c94b7fa429abffacdadb09e310df842de74fc9711c62ae78c2
bada7827ecb50a8f373929d75ebdc1a299bfa11311091d89f34ca) ([])
validator              | [2023-10-29 23:38:41.775 INFO     block_validator] Finis
hed block validation of: c26e26ad96918b937d7574773483d60e558f68829d7d96874c80b41
7d0216ddf07216cd0217cfd49794fe427aa78e4bcf8e79932931b0bbb4fbe29bfe8ad7718 (block
_num:10, state:b1c400e1d53ef708759c760ac74c73539915ae185aa6449d3ae4334cd39da2ee,
 previous_block_id:b5f55d81297d7410efa481ec1c94b7fa429abffacdadb09e310df842de74f
c9711c62ae78c2bada7827ecb50a8f373929d75ebdc1a299bfa11311091d89f34ca)
```

Figure 3.7: Block Validation after Upload

showcases the functionality of the endpoints mentioned above in 3.3.
In 3.6 and 3.7 we represent the page where a Producer can easily register a new file in the blockchain trough our next.js page. It is also shown the confirmation of a new block being created.

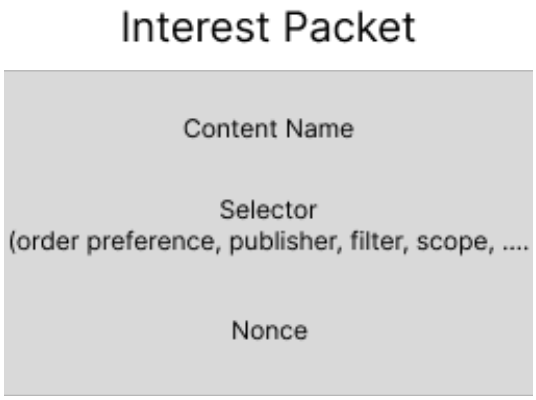


Figure 3.8: ndnSIM Interest Packet Example

Figure 3.9: ndnSIM Data Packet Example

3.3.4 Exchanged Packets in the Network (and Manifest)

In Figure 3.8 is exemplified an ndnSIM interest packet, and in figure 3.9 a data packet. The Interest Packet carries a name that identifies the desired chunk. The Data Packet carries the name of the requested chunk and the data itself. Routers remember the interface from which the request arrived, and then forwards the Interest packet by looking up the name in its Forwarding Information Base (FIB), which is populated by a name-based routing protocol. The Consumer send the Interest Packet to request the desired content and then receives a Data Packet from the Producer containing the data itself. Both have a common Content identifier designated as Name, which is a sequence of name components that follow a hierarchical structure with variable length alongside with any useful network data relevant information to the future intervening nodes.

Manifest

The manifest comes from the blockchain. Prior to requesting a file, the consumer first obtains the corresponding manifest, which allows for comprehensive security and integrity

verifications to be conducted during the subsequent file retrieval process.

This Manifest comes inside the Data field, and includes specific fields such as:

- "nome_ficheiro"
  This attribute specifies the name of the file under consideration.
- "merkle_tree"
  The Merkle tree root hash is a cryptographic hash value generated by processing the individual chunks of the file, with this hash the Consumer can verify that the file is unaltered.
- "assinatura_do_ficheiro"
  This field is the file hash, it is hashed using sha256 (explained in 2.2.3). It serves as a cryptographic proof of the file's authenticity and integrity.
- "numero_de_chunks"
  Where is specified the number of chunks that the file is split on.
- "tamanho_dos_chunks"
  Specify the size of each chunk.
- "comentario"
  This is an optional field for additional remarks or descriptions associated with the file. It serves as a simple commentary, providing context about the file, helps in search purposes.
- "chunks_hashs"
  This section provides a detailed breakdown of the cryptographic hashes for each data chunk, sha256 hashes. These hashes are essential for validating the content of each chunk and ensuring its integrity.
- "timestamp"
  The timestamp records the date and time when the manifest was generated.

3.3.5 Merkle Tree

A Merkle Tree approach was used to ensure the validity of the files in an effort to boost security. Efficiently confirming the legitimacy and consistency of data when it arrives to the Consumer, this is made feasible by these cryptographic data structures. With the use of Merkle Trees, we can ensure that the information obtained from the NDN is valid and unaltered.

In this project, the consumer gradually builds a Merkle tree, calculating is leafs, after receiving each chunk. Having this, the file's integrity and protection against any unwanted changes are verified by comparing the root of this Merkle tree with the root value in the manifest (explained in 3.3.4).

3.3.6 Chunks Hashes

The Consumer actively compute the SHA-256 hash of each chunk to make his verification. For this verification, the SHA-256 hash is carefully compared to the one in the manifest. Each individual data chunk validity and integrity is determined by this authentication method. In this way, the consumer can check chunk-by-chunk, if a chunk is altered or unaltered during the run time.

CHAPTER 4

Results

A Smart City needs a robust digital infrastructure, this means a well-developed network and, of course, security. Our scope is VANETs, one big side of smart cities, that relies on vehicles to relay information. In this work, we aim to use a multi-technology environment based on ATCLL. To evaluate our proposal and evidence the effectiveness of our solution in addressing the identified challenges, was decided to perform particular tests who assess metrics such as end-to-end time, security overhead and the amount of bytes exchanged.

Our initial approach involved a topology of 9 nodes (as illustrated in 4.1), which was simple and efficient, utilizing a single consumer and a single producer. With this topology, we gathered our first results and deductions. After this early findings, was time to go further and was implemented a topology resembling a real-life scenario (Figure 4.2). Featuring 20 nodes, where one serves as the producer and there is the potential to use up to eleven consumers. This new topology played a important role on testing the scalability of the system and its ability to handle increased network stress.

4.1 Testing

4.1.1 Topology

9 Nodes Test Topology

The first network topology that is emulated using ndnSIM, is shown in Figure 4.1, where we provide a visual representation of the initial phase of our testing process. During this phase, we deployed a topology with nine nodes, featuring a single producer and a single consumer.

In these preliminary stages, we allocated 1 Megabits per Second (Mbps) of bandwidth and introduced a 10ms delay.

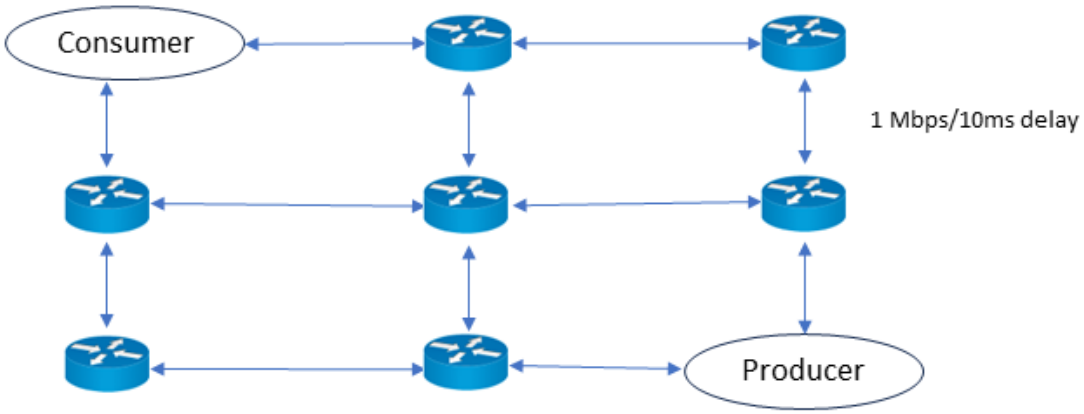


Figure 4.1: NDN 9 nodes Topology

Real-Scenario Test Topology

The second network topology, emulated using ndnSIM, is illustrated in Figure 4.2. The simulated topology consist of one Producer node and eleven Consumer nodes. These nodes are interconnected via both Ethernet and Wi-Fi links, each providing respective bandwidths of 1 Gigabits per Second (Gbps) and 54 Mbps, along with corresponding delay characteristics of 5 ms and 50 ms.

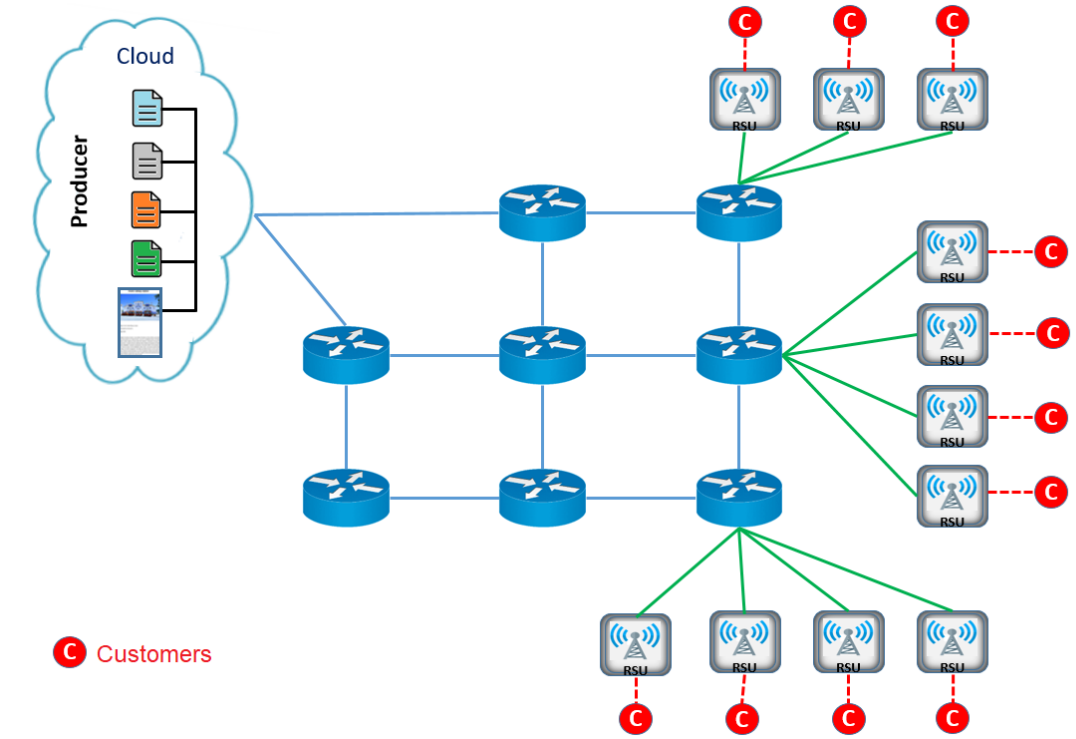


Figure 4.2: NDN 11 Consumers Topology

4.1.2 Influence of the Security Layer

For this section of tests, was used the 9 nodes topoly from 4.1.

End-to-End vs File Size

To perform this test, was used a carefully selected interval of file sizes to assess a specific aspect of our system's performance. The interval for this file size testing is derived from prior research Dinneen et al. [36]. File sizes ranging from 512 bytes to 2 gigabytes (2MB) are included in our investigation. Testing is done on every file size, namely 512 bytes, 2 kilobytes (2KB), 8 kilobytes (8KB), 32 kilobytes (32KB), 128 kilobytes (128KB), 512 kilobytes (512KB), and 2 megabytes (2MB). For each test instance, a single Consumer and a single Producer are deployed to interact with files of predetermined sizes. The file size is systematically incremented according to the specified size range. To ensure robust results, each test is conducted multiple times. Specifically, we run the test 10 times for each file size. Subsequently, the mean is calculated from the obtained data to provide a comprehensive overview of the system's performance and to reduce the margin of error derived from the machine performance.

The data collection process was conducted twice: once with the security layer of the project enabled, and once without it.

Drawing upon the insights derived from the graph presented in Figure 4.3, it can be inferred that the security implementation integrated into this project does not employ a substantial influence on the processing time of the network's consumers, as seen in 4.3, in the worst case scenario there was a 26% discrepancy. Nonetheless, it is worth noting that, as expected, the larger the file size, the greater the disparity in the values.

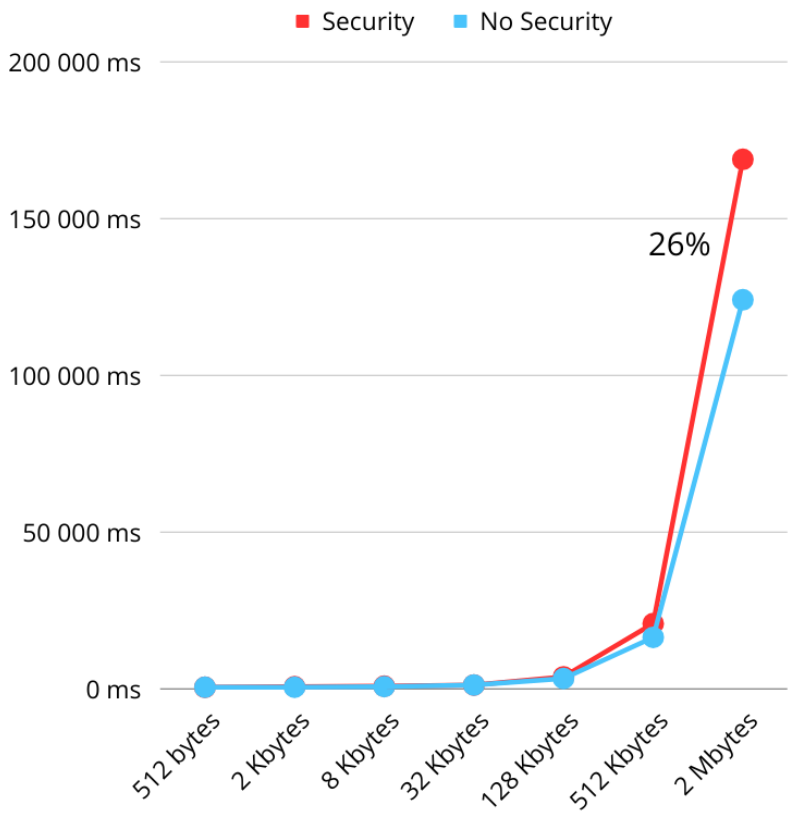


Figure 4.3: End-to-End Time vs File Size

Security Overhead vs File Size

Was maintained the utilization of a single Consumer and Producer throughout this experimental testing phase, with a deliberate focus on progressively increasing the file sizes under investigation. This deliberate approach was implemented to ensure the robustness and consistency of our findings. Each file size was subjected to ten independent test iterations, and the results from these repetitions were subsequently used to calculate the mean values. This meticulous methodology was adopted to guarantee the accuracy and reliability of our results.

The analysis of 4.4 graph yields significant insights into the relationship between file size and the associated security overhead. As the graph clearly illustrates, the overhead decreases as the file size increases. We totaled the time spent on security-related tasks and divided that amount by the overall duration for the end-to-end process to determine this security overhead. This observation is not only encouraging but also rational, as it aligns with our expectations and bears relevance to the nature of the task at hand.

This trend makes sense for several reasons. Firstly, as file size increases, the ratio of security-related computational operations to the total computational load diminishes. Larger files inherently require more processing power, and the security overhead constitutes a decreasing proportion of this total workload. Consequently the impact of security measures on processing time becomes less pronounced.

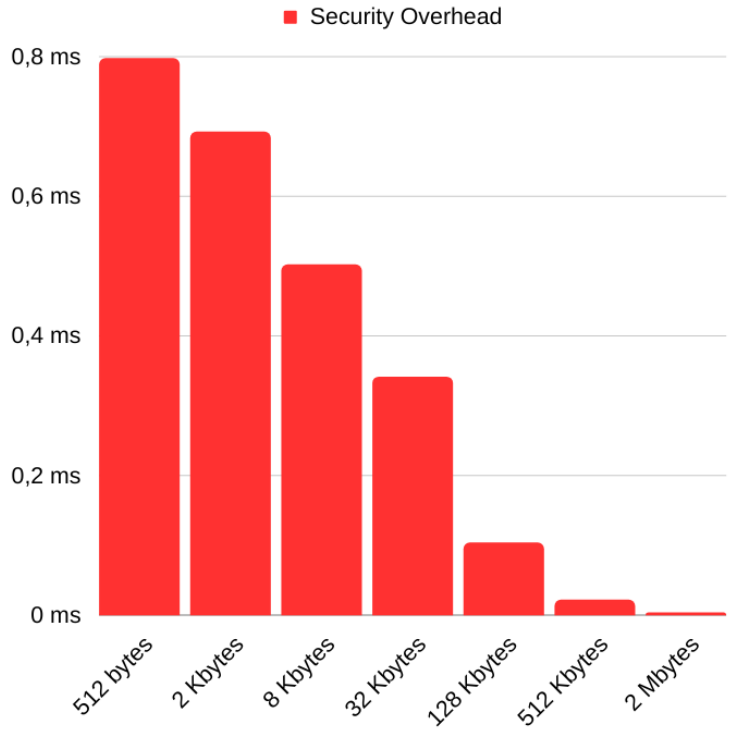


Figure 4.4: Security Overhead vs File Size

Furthermore, larger files often involve a longer time duration for transmission or storage. In this type of scenarios as a network, security measures, despite their computational demands, become a justifiable trade-off for the increased protection they offer.

Bytes Exchanges vs File Size

As anticipated, upon careful examination of the graphical representation 4.5 we can deduce that the volume of bytes exchanged between the consumer and the blockchain exhibits a discernible pattern of growth in line with the increasing file size. This observation is consistent with our initial expectations and provides valuable insights into the dynamics of data transfer within this ecosystem.

It is important to note that these values encompass not only the data manifest transmitted by the blockchain but also encompass the manifest itself and the bytes associated with the messages exchanged throughout the course of the dialogue between the consumer and the blockchain. This holistic consideration of data exchange elucidates the comprehensive nature of the communication process, which extends beyond mere file size and encompasses the intricacies of interaction between the consumer and the blockchain.

In accordance with the methodology employed in the previous tests, a solitary consumer and a singular producer were once again utilized for the present experiment. The data collection process consisted of acquiring ten data samples for each distinct file size under examination, followed by the computation of the arithmetic mean. This rigorous approach

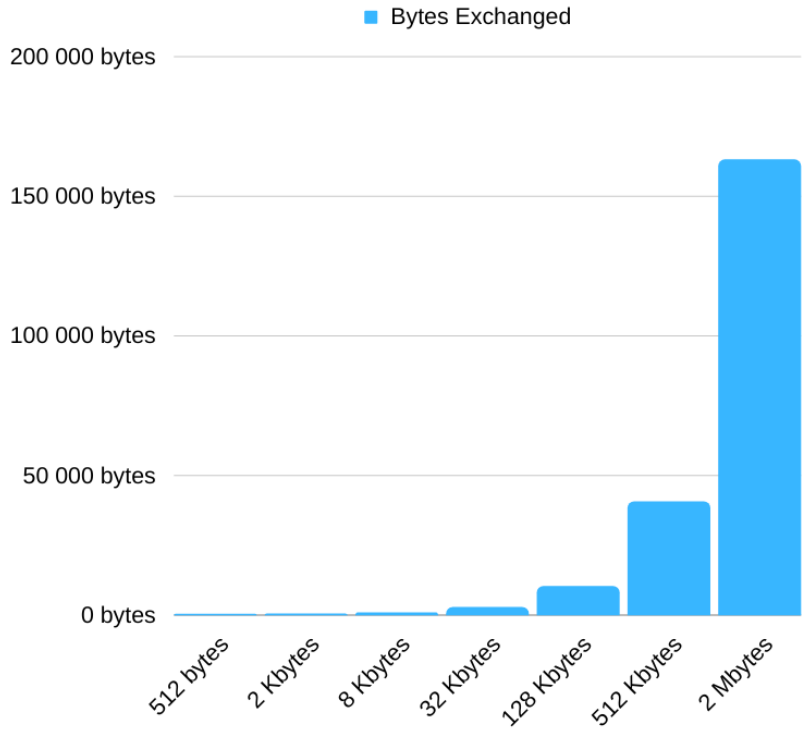


Figure 4.5: Bytes Exchanges vs File Size

was employed to maintain consistency and statistical reliability throughout the study.

4.1.3 Three Consumers and different File Sizes

In this section we advanced to the topology with 20 nodes referenced in 4.2.

In order to validate the robustness of this project and of the previous findings, was opted to redo the tests, which featured a single consumer and a single producer. However, for this phase of the study, the examination was extended with the increase of consumers to three.

This strategic shift allowed us to investigate the performance of the system under the influence of multiple consumers, a scenario that is a little more close to the real-world usage.

This expansion of the consumer base was motivated by the desire to assess the system's ability to accommodate and sustain consistent results when subjected to increased demand. By introducing this element, we aimed to confirm that the principles and standards established in the initial experiments persist in a multi-consumer environment. The systematic repetition of these tests with a larger consumer group provided valuable insights into the system's scalability, ensuring that our work not only maintains its validity but also addresses multi-consumer scenarios with precision and confidence.

End-to-End vs Three Consumers

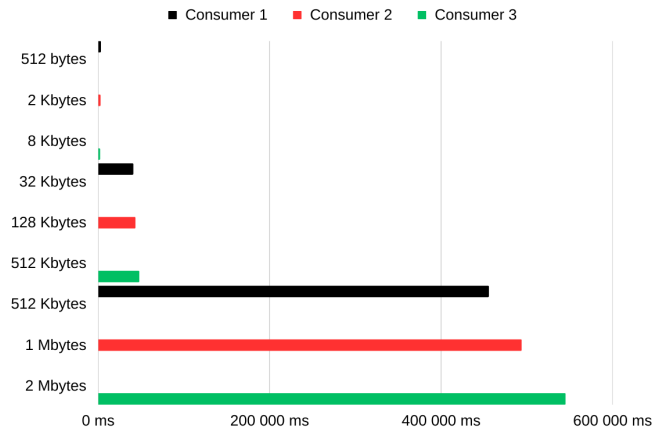


Figure 4.6: End-to-End vs Three Consumers

For these testing three sample groups were selected, each with three different file sizes, so that we could see how the system behaves in different scenarios.

The first group is compound of 512 bytes, 2 kilobytes, and 8 kilobytes, the second group consists of 32 kilobytes, 128 kilobytes, and 512 kilobytes. The last and larger group, in terms of size, had files of 512 kilobytes, 1 megabyte, and 2 megabytes.

Throughout these tests, all three consumer entities and a single producer operated concurrently, emulating a more realistic usage scenario where multiple consumers interact with the

system simultaneously. To ensure statistical robustness and reliability, each test iteration was executed ten times for every three file group. Subsequently, the results from these repetitions were used to calculate the mean values for each respective group.

In the tests involving three consumers, a striking similarity to the behavior observed in the single consumer test became apparent. The relationship between the size of the file group and the end-to-end time remained consistent. Specifically, as the size of the file group expanded, a proportional increase in the end-to-end time was consistently observed. This trend remained consistent for each individual consumer within the group, wherein the end-to-end time increased correspondingly with the size of the file they requested. In essence, larger files led to longer end-to-end processing times for each consumer, reaffirming the direct correlation between file size and processing duration. This consistency in results across both single and multiple consumer scenarios underscores the robustness of the system's performance characteristics.

Our observations suggest that the system maintains its response patterns regardless of the number of consumers in operation, indicating a high level of predictability and scalability. These findings not only validate the system's capacity to manage increased workloads but also emphasize the system's dependability and uniformity in processing times across diverse file sizes and consumer interactions.

Security Overhead vs Three Consumers

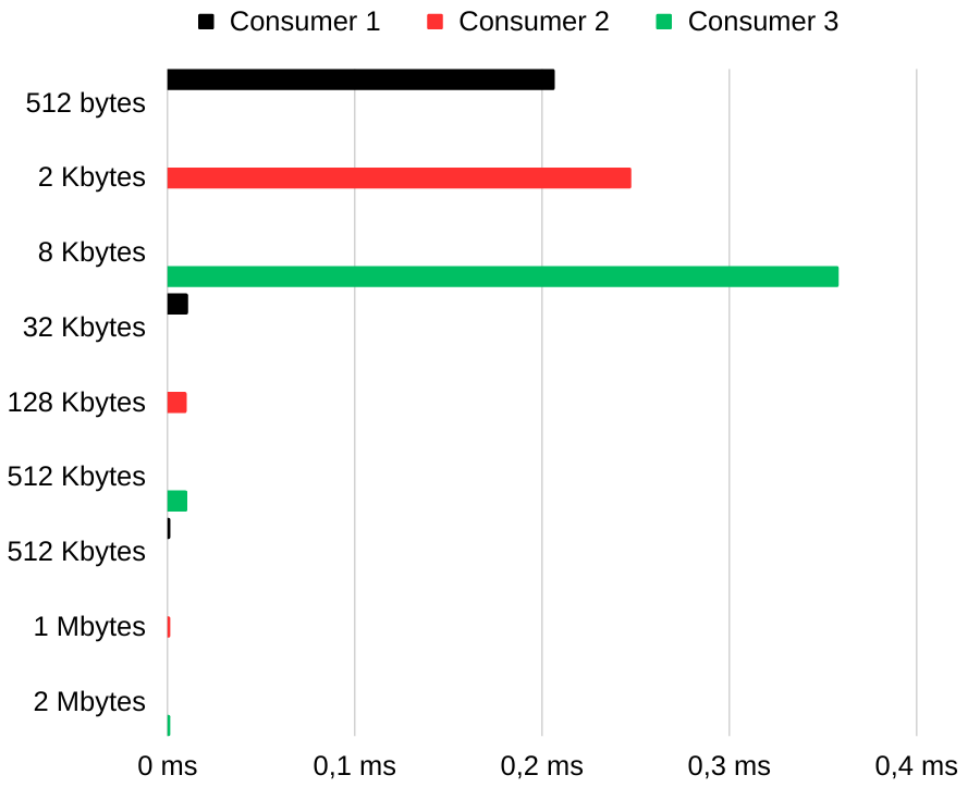


Figure 4.7: Security Overhead vs Three Consumers

In continuation of the established methodology used in the previous test, the current study maintains consistency by employing the same three predefined groups and the identical set of nine files, three for each group. However, our focus has now shifted to a nuanced aspect of the system - the evaluation of Security Overhead.

In line with our prior observations and findings, our investigation into Security Overhead has yielded consistent outcomes. As anticipated, we find a pronounced concordance with the results obtained from the single consumer tests. The pattern that emerges suggests that the Security Overhead is inversely proportional to the size of the files or the overall size of the file group. In simpler terms, bigger files and bigger groups, in terms of size, result in less Security Overhead.

Maintaining the perspective that the trade-off for enhanced protection lies within justifiable bounds, our security layer becomes a reasonable compromise.

Exchanged bytes vs Three Consumers

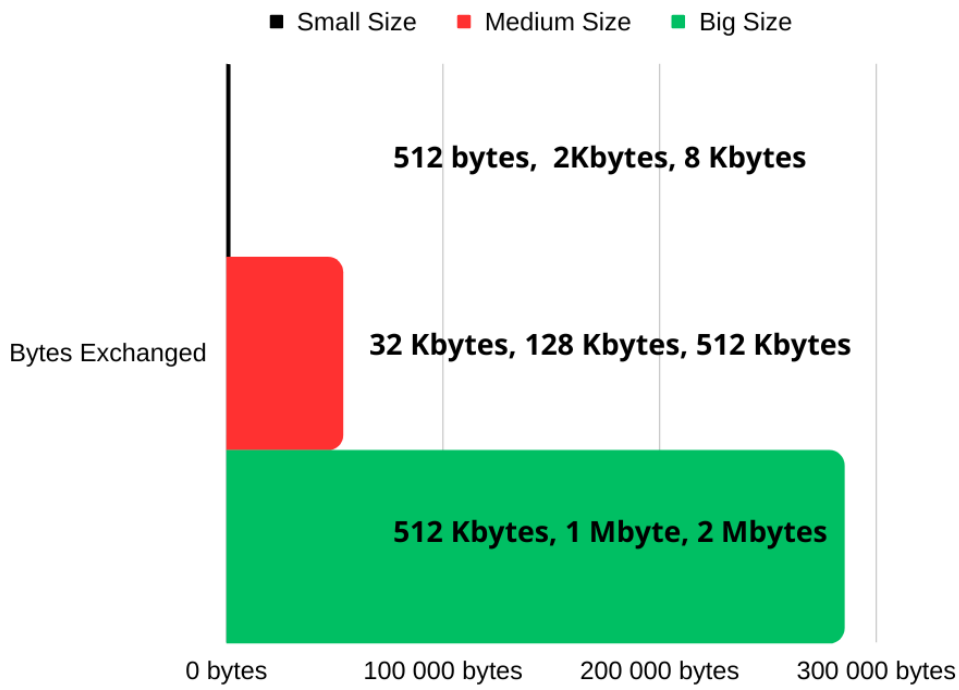


Figure 4.8: Exchanged bytes vs Three Consumers

For the purpose of demonstration, nomenclature was assigned to the groups based on their respective sizes. These three designated groups nomenclature became small size (512 bytes, 2Kbytes and 8 Kbytes), medium size (32 Kbytes, 128 Kbytes, 512 Kbytes), and large size (512 Kbytes, 1 Mbytes, 2 Mbytes).

As anticipated, it was observed that the size of the group directly influences the quantity of data exchanged between the consumer and the blockchain. This data encompasses both

the manifest and conversations between the consumer and the blockchain network. Notice that the influence on the volume of bytes consumed by discussions was relatively insignificant compared to that of the manifest. It is evident that larger groups result in a correspondingly larger volume of bytes exchanged, thereby underscoring the impact of group size on data transfer.

4.1.4 Consumer Scalability Stress Tests

The primary objective of these tests is to scrutinize the network's performance under varying degrees of stress, with a focus on assessing its scalability.

To achieve this objective we maintained the topology with 20 nodes referenced in 4.2.

The stress tests involve a systematic increase in the number of consumers, ranging from an initial count of three and gradually scaling up to eleven, which is the desired topology in is worst case. This incremental augmentation in consumer load is employed to derive profound insights into the network's resilience and capacity to handle increased demand.

The overarching purpose of these stress tests is to investigate the network's response to stress-induced scenarios and to draw conclusive findings regarding its performance and robustness under challenging conditions. By meticulously examining the network's behavior as consumer numbers surge, valuable insights will be gained, allowing for a comprehensive evaluation of its scalability and stress resilience. The results from these Scalability Stress Tests play a pivotal role in elucidating the network's overall capacity and its ability to adapt to ever-evolving demands, thereby contributing significantly to the broader discourse on network optimization and performance enhancement.

End-to-End time vs Increase of Consumers

The graph depicting 4.9 who compares End-to-End time vs. the increase of the number of consumers exhibits a discernible trend. As anticipated, it is evident that, with an increasing number of consumers, the average End-to-End time also increases. This observation aligns with the intuitive expectation that as the network accommodates more consumers, the overall workload intensifies, leading to a longer end-to-end time.

However, it is noteworthy that the increase in average End-to-End time was not disproportionate to the incremental addition of consumers. The fact that the increase was not overly pronounced is indicative of a network that exhibits positive scalability. This favorable outcome underscores the system's ability to efficiently accommodate additional consumers without an overwhelmingly negative impact on End-to-End time.

The observed increase in average End-to-End time is primarily attributable to the stress encountered as the network handles a growing number of consumers. It reflects the underlying dynamics of resource allocation and data processing. Nevertheless, the moderate rise in End-to-End time is a promising result, as it suggests that the network can be scaled effectively to meet the requirements of a larger user base.

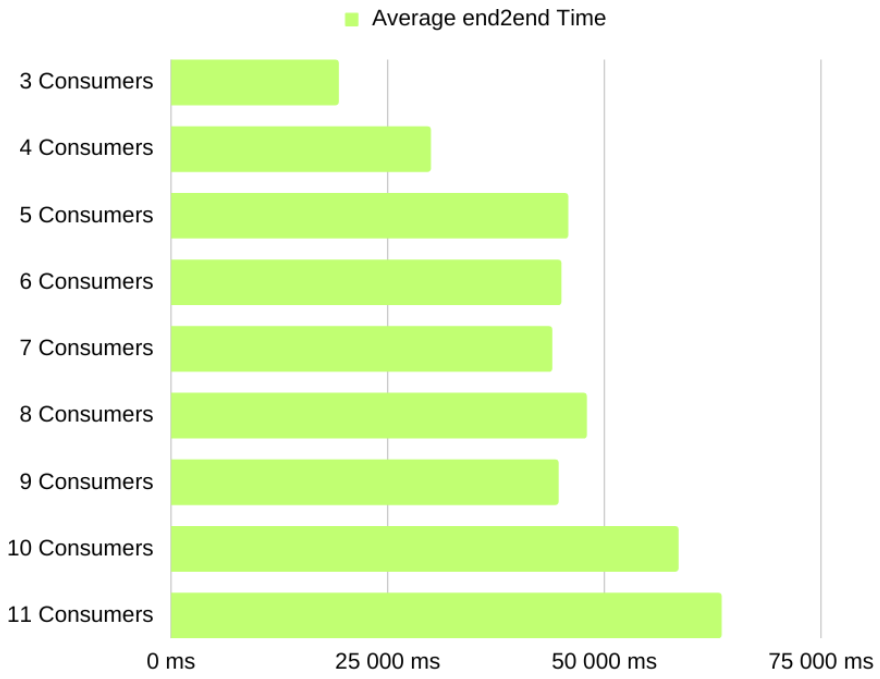


Figure 4.9: End-to-End time vs Increase of Consumers

Security Overhead vs Increase of Consumers

In this test was examined the relationship between the number of consumers and the security overhead, with the objective of elucidating the extent to which security measures affect network performance and scalability.

The graph illustrating Security Overhead vs. the increase of consumers 4.10 revealed an intriguing and positive trend. As the number of consumers progressively expanded from the initial three to a final count of eleven, the security overhead, instead of escalating, exhibited a notable reduction. This counterintuitive observation is a testament to the effectiveness of the security layer implemented within the network.

The diminishing security overhead with the increasing number of consumers underscores the practical worth of the security measures in place, while also highlighting the network's scalability. It demonstrates that the security layer is not only efficient but also gradually less significant to the network's computational performance even as it scales to accommodate a larger user base.

The noted decrease in security overhead as the number of consumers increases provides a compelling endorsement of the security layer's significance within the network infrastructure. This observation conveys that the network not only accommodates the requirements of scalability effectively but also does so without compromising its core responsibilities, which include upholding data integrity and user protection, while also maintaining a reasonable

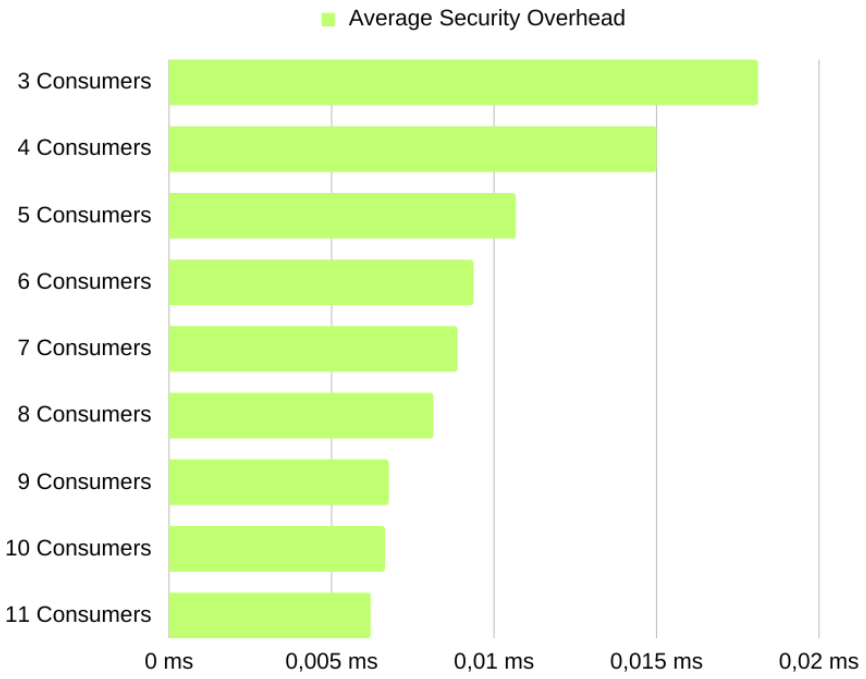


Figure 4.10: Security Overhead vs Increase of Consumers

computational load on the system.

Exchanged bytes vs Increase of Consumers

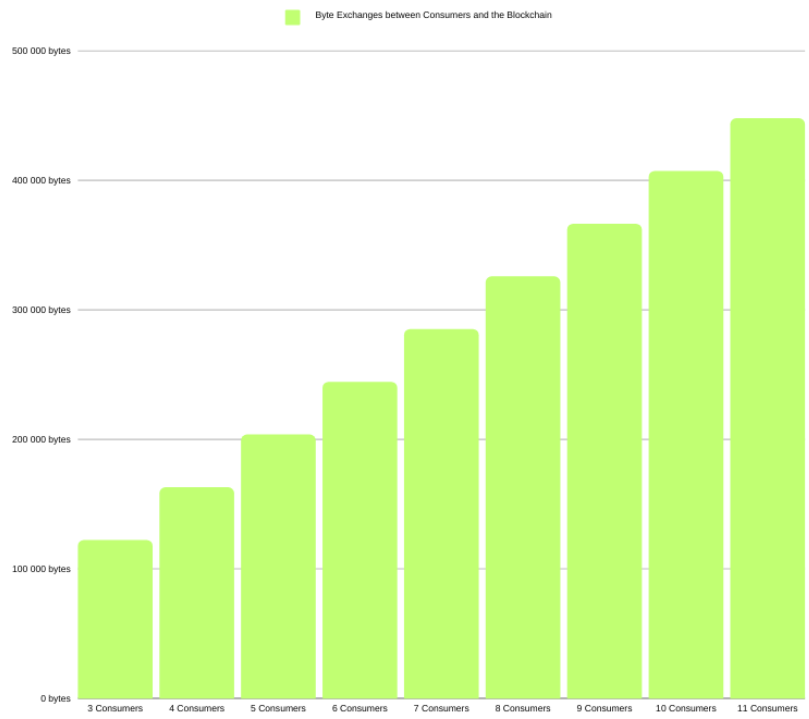


Figure 4.11: Exchanged bytes vs Increase of Consumers

One of the fundamental aspects in assessing the performance and scalability of our network is the analysis of data exchange. To this end, was conducted a series of experiments aimed at understanding how the number of consumers impacts the volume of data exchanged between the consumer and the blockchain. The graph, 4.11, serves as a visual representation of this investigation.

As the graph vividly illustrates, with the increase in the number of consumers, there is a direct and proportionate rise in the volume of exchanged bytes. This phenomenon reflects the growing data load encountered as more consumers engage with the blockchain network.

It is important to note that the exchanged bytes, in this context, encompass both the manifest and the conversations between each consumer and the blockchain. Notably, the manifest and the conversations play distinct roles in the data exchange process. The manifest, which encapsulates the essential information, tends to be the dominant component of exchanged bytes. It serves as a comprehensive guide outlining the data to be transferred and is essential for the secure and accurate exchange of information. Conversely, the conversation data, representing the actual dialogues and interactions between consumers and the blockchain, constitutes a smaller portion of the exchanged bytes. This observation underscores that, in the context of data transfer, the conversations are considerably less influential in terms of data volume compared to the manifest.

The findings gleaned from this analysis are significant for several reasons. First and foremost, the graph reaffirms the intuitive expectation that as more consumers engage with the network, the volume of data exchanged proportionally increases. This insight is crucial for capacity planning and network optimization, enabling the anticipation of resource demands as user numbers grow.

These findings are instrumental in fostering a deeper understanding of the network's performance and its potential for further refinement in the context of scalability.

CHAPTER 5

Conclusion and Future Work

In order to address the unresolved security issues in NDN, this thesis proposes a blockchain-based authentication mechanism for NDN for smart city environments. The system use a decentralized strategy based on a consensus mechanism and a adaptive routing, producing a transparent, distributed, and tamper-proof system. In NDN contexts, the usage of DLT, in particular blockchain technology, offers a solution for reliable and secure communication.

ATCLL real mobility datasets and an NDN simulator will be used to evaluate the proposed method in the context of a vehicular system in Aveiro in order to determine its applicability in real-world settings. The first step in the design of the solution was to conduct a state-of-the-art research on NDNs and Distributed Ledger Technology to evaluate existing solutions and see the challenges still open. This step was completed and the results are presented in Chapter 2.

In Chapter 3 is shown that the solution proposed was designed having in mind real scenarios, such as Smart Cities including VANETs, we accomplished that by using a multi-technology environment based on ATCLL. In this chapter is addressed the proposed security solution, that is blockchain-based.

Finally, in Chapter 4 the system is fully tested. As aimed, with the help of the manifest, trough the blockchain and with hashes technology and the use of a merkle tree, each Consumer was able to authenticate their files. As evident from the results, the Security Overhead was minimal. When comparing the resources allocated to exchange data in the network, the resources dedicated to security aspects are notably low. The scalability of the system under stress was achieved too as illustrated in 4.1.3 and 4.1.4. It was able to display good results as the NDN Security was accomplished. Representing a capable solution to address the security problems in a NDN architecture.

5.1 Future Work

One obstacle that was found trough the process of developing this project was the fact that the blockchain framework was heavy and slow. In response to this issue, we thought about an optimisation strategy based on hash tables. This approach can reduce the amount

of requests made to the blockchain, only to occasions when a new file is uploaded, rather than accesing it each time the Consumer needs a new file.

References

[1] S. Mastorakis, A. Afanasyev, and L. Zhang, «On the evolution of ndnSIM: An open-source simulator for NDN experimentation», *ACM Computer Communication Review*, Jul. 2017.

[2] P. Rito, A. Almeida, A. Figueiredo, *et al.*, *Aveiro tech city living lab: A communication, sensing and computing platform for city environments*, 2022. DOI: 10.48550/ARXIV.2207.12200. [Online]. Available: https://arxiv.org/abs/2207.12200.

[3] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, «A survey of information-centric networking», *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, Jul. 2012. DOI: 10.1109/MCOM.2012.6231276.

[4] L. Zhang, A. Afanasyev, J. Burke, *et al.*, «Named data networking», *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, Jul. 2014, ISSN: 0146-4833. DOI: 10.1145/2656877.2656887. [Online]. Available: http://doi.acm.org/10.1145/2656877.2656887.

[5] L. Gameiro, C. Senna, and M. Luís, «Insights from the experimentation of named data networks in mobile wireless environments», *Future Internet*, vol. 14, no. 7, 2022, ISSN: 1999-5903. DOI: 10.3390/fi14070196. [Online]. Available: https://www.mdpi.com/1999-5903/14/7/196.

[6] B. Nour, K. Sharif, F. Li, *et al.*, «A survey of internet of things communication using icn: A use case perspective», *Computer Communications*, vol. 142-143, pp. 95–123, 2019, ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2019.05.010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366418309228.

[7] L. Zhang and B. Zhang, «Ndn vs ip: A comparison of two networking paradigms», *IEEE Communications Magazine*, vol. 55, no. 8, pp. 84–89, 2017.

[8] L. C. M. Hurali and A. P. PATIL, «Application areas of information-centric networking: State-of-the-art and challenges», *IEEE ACCESS*, vol. 10, pp. 1669–1678, 2022.

[9] L. Zhou, J. Liu, and C.-X. Wang, «A survey of security threats and countermeasures in named data networking», *IEEE Communications Surveys and Tutorials*, vol. 19, no. 3, pp. 1589–1624, 2017.

[10] E. G. AbdAllah, H. S. Hassanein, and M. Zulkernine, «A survey of security attacks in information-centric networking», *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1441–1454, 2015. DOI: 10.1109/COMST.2015.2392629.

[11] J. Zhang, S. Li, and C. Wang, «A secure dynamic content delivery scheme in named data networking», *Security and Communication Networks*, vol. 2022, no. 6304927, 2022.

[12] X. Jiang, L. Zhang, and K. Fall, «Named data networking with dhts», in *Proceedings of the ACM Conference on Information-Centric Networking*, ACM, 2017, pp. 51–60.

[13] R. Merkle, «A digital signature based on a conventional encryption function», *Advances in Cryptology—CRYPTO'87*, pp. 369–378, 1987.

[14] H. Liu, X. Luo, H. Liu, and X. Xia, «Merkle tree: A fundamental component of blockchains», in *2021 International Conference on Electronic Information Engineering and Computer Science (EIECS)*, 2021, pp. 556–561. DOI: 10.1109/EIECS53707.2021.9588047.

[15] L. Zhang, A. Afanasyev, J. Jain, C. Yang, B. Wang, and L. Zhang, «Named data networking», in *ACM SIGCOMM Computer Communication Review*, ACM, vol. 44, 2014, pp. 66–73.

[16] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, [Online]. Available: https://bitcoin.org/bitcoin.pdf, 2008.

[17] R. Martínez-Maldonado, N. Guzman-Ramirez, and M. Zuniga, «Blockchain governance: A research framework and survey», *Information Systems Frontiers*, vol. 23, pp. 769–792, 2021.

[18] J. Mattoni, S. Gesell, and D. Tapscott, «Blockchain governance in the age of cryptocurrency», *Journal of Business Economics*, vol. 88, pp. 547–567, 2018.

[19] K. Christidis and M. Devetsikiotis, «A framework for blockchain governance», *IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2016.

[20] N. Kshetri, «Blockchain governance: A study of blockchain-based decentralized autonomous organizations (daos)», *Information Systems Management*, vol. 37, pp. 107–118, 2020.

[21] L. Giudici, M. Al-Rifaie, and K. Pant, «The economics of blockchain governance», *Information Systems Frontiers*, vol. 22, pp. 1457–1472, 2020.

[22] L. Silva, C. Senna, and A. Zúquete, «Using reputation as a coin to bet on information items distributed in a smart city», in *2019 Fifth Conference on Mobile and Secure Services (MobiSecServ)*, (Miami, FL, USA), Mar. 2019, pp. 1–2. DOI: 10.1109/MOBISECSERV.2019.8686570.

[23] M. A. Azad, S. Bag, and F. Hao, «Privbox: Verifiable decentralized reputation system for online marketplaces», *Journal of Computer Science*, vol. 89, pp. 44–57, 2018.

[24] T. Song, X. Zhang, Y. Wang, and Y. Liu, «Smart contract-based trusted content retrieval mechanism for ndn», *IEEE Access*, vol. 8, pp. 85 813–85 824, 2020.

[25] R. Dennis and G. Owen, «Rep on the block: A next generation reputation system based on the blockchain», *International Journal of Blockchain Technology*, 2018.

[26] F. Dotzer, L. Fischer, and P. Magiera, «Vars: A vehicle ad-hoc network reputation system», *IEEE Xplore*, 2005.

[27] M. B. Mollah, J. Zhao, D. Niyato, *et al.*, «Blockchain for the internet of vehicles towards intelligent transportation systems: A survey», *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4157–4185, 2021. DOI: 10.1109/JIOT.2020.3028368.

[28] *Hyperledger – open source blockchain technologies*, https://www.hyperledger.org/, Accessed on 04/07/2023.

[29] L. F. S. Silva, «Dignitas: Using reputation as a coin to evaluate human sensing in smart cities», Master thesis, Universidade de Aveiro, Dec. 2019.

[30] N-able, *N-able - sha-256 algorithm overview*, https://www.n-able.com/blog/sha-256-encryption, Accessed on 26/09/2023, 2019.

[31] I. Vercel, *What is next.js?*, https://nextjs.org/learn-pages-router/foundations/about-nextjs/what-is-nextjs, Accessed on 20/10/2023.

[32] L. Gameiro, C. Senna, and M. Luís, «Ndniot-fc: Iot devices as first-class traffic in name data networks», *Future Internet*, vol. 12, 11 2020. DOI: 10.3390/fi12110207. [Online]. Available: https://doi.org/10.3390/fi12110207.

[33] N. D. Networking, *Named data networking forwarding daemon (nfd) 22.12 documentation*, https://docs.named-data.net/NFD/current/, Accessed on 24/09/2023.

[34] I. M. Alexander Afanasyev Spyridon Mastorakis and L. Zhang, *Introduction*, https://ndnsim.net/current/intro.html?highlight=architecture, Accessed on 25/09/2023.

[35] Hyperledger, *Hyperledger sawtooth*, https://sawtooth.hyperledger.org/, Accessed on 24/09/2023.

[36] J. Dinneen and B. Nguyen, «How big are peoples' computer files? file size distributions among user-managed collections», Jul. 2021.