ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

ROBEN CASTAGNA LUNARDI

MULTI-LEVEL CONSENSUS ALGORITHM FOR APPENDABLE-BLOCK BLOCKCHAINS IN IOT ENVIRONMENTS

Porto Alegre

2021

PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
SCHOOL OF TECHNOLOGY
COMPUTER SCIENCE GRADUATE PROGRAM

MULTI-LEVEL CONSENSUS ALGORITHM FOR APPENDABLE-BLOCK BLOCKCHAINS IN IOT ENVIRONMENTS

ROBEN CASTAGNA LUNARDI

Doctoral Thesis submitted to the Pontifical Catholic University of Rio Grande do Sul in partial fulfillment of the requirements for the degree of Ph. D. in Computer Science.

Advisor: Prof. Avelino Francisco Zorzo

Porto Alegre
2021

Ficha Catalográfica

L961m Lunardi, Roben Castagna

Multi-level consensus algorithm for appendable-block blockchains in IoT Environments / Roben Castagna Lunardi. – 2021.
149 p.
Tese (Doutorado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Avelino Francisco Zorzo.

1. Blockchain. 2. Distributed ledgers. 3. Consensus algorithms. 4. IoT. 5. Internet of Things. I. Zorzo, Avelino Francisco. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).
Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051

Roben Castagna Lunardi

MULTI-LEVEL CONSENSUS ALGORITHM FOR APPENDABLE-BLOCK BLOCKCHAINS IN IOT ENVIRONMENTS

This Doctoral Thesis has been submitted in partial fulfillment of the requirements for the degree of Doctor of Computer Science, of the Graduate Program in Computer Science, School of Technology of the Pontifical Catholic University of Rio Grande do Sul.

Sanctioned on March 3rd, 2021.

COMMITTEE MEMBERS:

Prof. Dr. Otto Carlos Muniz Bandeira Duarte (GTA/ UFRJ)

Prof. Dr. Weverton Luis Da Costa Cordeiro (PPGC/ UFRGS)

Prof. Dr. Fabiano Passuelo Hessel (PPGCC/PUCRS)

Prof. Dr. Avelino Fracisco Zorzo (PPGCC/PUCRS - Advisor)

“You can’t run from the future, you can’t change the past, you’re not that fast.”
(Saul Hudson & Ian Robert Astbury)

ACKNOWLEDGMENTS

I want to thank to my family, in particular to my beloved partner Aline Ibaldo Gonçalves for all the support, care, love and patience during the last 14 years. It would be impossible to handle everything without her support. Also, I want to thank to my parents, Valter Luiz Lunardi and Inês Castagna Lunardi, for the education, care, love and example during my whole life. Additionally, I want to thank to my brothers Delx and Fabricio for the lessons and good (and, sometimes, bad) examples that made me grow.

Moreover, I am really grateful for my advisor Avelino Francisco Zorzo for being kind, helpful and patient during the four years of my PhD. This thesis would have been impossible without the support and mentoring of Avelino. Even working with him for 4 years, I am still surprised by his intelligence and friendliness. I have also received a lot of support and help from my friends in the Dependability and Security Research Group (CONSEG). Special thanks to Régio Antônio Michelin, Charles Varlei Neu, Daniel Dalalana Bertoglio for the partnership, discussions and beers. Also, I want to thank everyone from the CONSEG that helped during the years, such as Henry C. Nunes, Aline Zanin, Eduardo Arruda and Alex Orozco.

Additionaly, I really appreciate the friendship and discussions from many colleagues from PUCRS. In particular, I would to like to thank to Marcelo Paravisi, Laura Angélica Tomaz da Silva, Luisa de Castro Guterres, Ramon Pereira and Ramão Tiburski.

Furthermore, I thank to Salil S. Kanhere for the valuable discussions, friendship and nice experiences during (and after) my research internship at UNSW - Sydney. Salil has been an example of researcher and friend that I will carry for my whole life, thank you. Also, I thank Volkan Dedeoglu for the collaboration in many works with (very) short deadlines.

In addition, I want to thank to all member from Secure & Resilent System group - Newcastle University for the whole support during my 6 months research visit. In particular, I would like to thank to Aad van Moorsel for the inputs in my research, friendship and support, it was really important to me. Also, I have received a lot of feedbacks, inputs and support from Maher Alharby, including long discussions about this work and conversations about many other subjects. Thank you very much Maher for the collaboration and friendship. Also, special thanks to Amjad Aldweesh, Brian Randell, Changyu Dong, Ehsan Toreini, John Mace, Luca Arnaboldi, Magdanele Ng, Maryam Mehrnezhad, Milad Kazemi, Ricardo M.

Czekster, Roberto Metere (sorry if I forgot to cite someone), whose made my time in UK meaningful, easier and fun.

Finally, this study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior – Brasil (CAPES) – Finance Code 001. Additionally, I thank the financial support from IFRS. Also, I thank Australian Academy of Science for the support in the 2018 Australia-Americas PhD Research Internship Program.

ALGORITMO DE CONSENSO MULTINÍVEL PARA BLOCKCHAIN COM BLOCOS EXTENSÍVEIS PARA AMBIENTES IOT

RESUMO

Atualmente, diferentes dispositivos coletam dados e prestam serviços na Internet. Alguns desses dispositivos - ou apenas coisas - colaboram para trocar informações e usá-las para tomar decisões mais inteligentes em um ambiente chamado Internet das Coisas (IoT - *Internet of Things*). A possibilidade de conectar objetos físicos do dia a dia está criando novos modelos de negócios, melhorando processos e reduzindo custos. No entanto, os problemas de segurança em IoT podem ter um alto impacto nos ativos físicos e corporativos. Recentemente, a tecnologia *blockchain* surgiu como uma possível solução para superar problemas de segurança em IoT. Apesar disso, as *blockchains* tradicionais (como o Bitcoin e Ethereum) não são adequadas para a natureza de recursos restritos dos dispositivos de IoT ou para o grande volume de informações produzidos em ambientes de IoT típicos. A adoção de uma estrutura de *blockchain* leve chamada *appendable-block blockchain* foi proposta para ser usada em ambientes IoT. Esta *blockchain* adota uma estrutura de dados diferente, baseada em blocos com dados desacoplados e anexáveis. Embora esta *blockchain* tenha apresentado bons resultados de desempenho (alguns milissegundos para acrescentar um novo bloco), a falta de um algoritmo de consenso o torna vulnerável a muitos problemas de segurança. Outro problema nas implementações atuais de *blockchain* é a falta de discussão sobre o comportamento dos usuários em diferentes contextos e como elas poderiam ser adaptadas para diferentes algoritmos de consenso. Para superar esse problema, esta tese apresenta um conjunto de etapas para criar um mecanismo de consenso multinível para diferentes contextos. A ideia principal é desenvolver uma solução que permita o uso de algoritmos de consenso no nível dos blocos e no nível das transações. Além disso, esta solução pode ajudar a paralelizar a inserção de informações que separando os nós em contextos. Essa abordagem pode ajudar a fornecer uma solução que pode usar diferentes configurações ou consensos simultaneamente, de acordo com os requisitos de cada contexto no ambiente de IoT. Finalmente, os resultados obtidos nos experimentos mostram que um consenso multinível pode produzir um alto rendimento e baixa latência para inserir novas transações em *appendable-block blockchains*.

Palavras-Chave: Blockchain, distributed ledgers, algoritmos de consenso, IoT, Internet das Coisas.

MULTI-LEVEL CONSENSUS ALGORITHM FOR APPENDABLE-BLOCK BLOCKCHAINS IN IOT ENVIRONMENTS

ABSTRACT

Currently, there are different devices collecting data and providing services through the Internet. Some of these devices - or just things - collaborate to exchange information and use them to make smarter decisions in an environment called Internet of Things (IoT). Connecting everyday physical objects is creating new business models, improving processes and reducing costs. However, security issues in IoT can have a high impact on both business and physical assets. Recently, the blockchain technology emerged as a possible solution to overcome security issues in IoT. Despite of that, traditional blockchains (such as Bitcoin or Ethereum) are not well suited to the resource-constrained nature of IoT devices or to the large volume of information expected from typical IoT environments. The adoption of a lightweight blockchain framework called appendable-block blockchain has been proposed to be used in IoT environments. This blockchain adopts a different data structure, based on blocks with decoupled and appendable data. While this blockchain presented good performance results (few milliseconds to append a new block), the lack of a consensus algorithm makes it vulnerable to many security issues. Another problem in current blockchain implementations is the lack of discussion on users behavior in different contexts and how it could be adapted for different consensus algorithms. To overcome this problem, this thesis presents a set of steps to create a multi-level consensus mechanism for different contexts. The main idea is to develop a solution that allows the usage of consensus algorithms at the block level and at the transaction level. Moreover, this solution can help to insertion of information in parallel, separating nodes in contexts. This approach can help to provide a solution that can use different configurations or consensus, according to the requirements of each context in the IoT environment. Finally, the results obtained in the experiments shows that a multi-level consensus can produce a high throughput and low latency to insert new transactions in appendable-block blockchains.

Keywords: Blockchain, Distributed ledgers, Consensus algorithms, IoT, Internet of Things.

LIST OF FIGURES

1.1 Schematic overview of the thesis structure. 33

2.1 Blockchain example. 37

2.2 Main blockchain architectures for IoT (extracted from [371].) 41

2.3 Inserting new block using dBFT (extracted from [80]). 44

2.4 Quorum slices used in FBA (extracted from [80]). 44

2.5 Block structure (extracted from [371]). 46

2.6 Merkle tree structure (extraxted from [371]). 47

2.7 Increasing number of papers about blockchain in IoT. 50

2.8 Layer-based model for blockchain categorization (extracted from [371]). 53

4.1 Gateway-based Architecture for IoT (extracted from [216]) 63

4.2 Appendable-block blockchain components (adapted from [215]). 64

5.1 Device connection simplified protocol. 72

5.2 Gateways in three different contexts (extracted from [214]). 74

6.1 Average time to append a transaction in the Gateway connected to the producer Device 84

6.2 Average time to perform the consensus for a block 84

6.3 **T1**: Average time to perform the consensus 87

6.4 **T2**: Average time to perform the consensus and propagate the block 87

6.5 **T3**: Overhead of the transaction insertion procedure 88

6.6 **T4**: Latency to insert a transactions in all gateways 89

6.7 Testbed environment (extracted from [76]). 90

6.8 Test Procedure 92

6.9 Average latency for each transaction in context-based consensus. 97

6.10 Average transactions consensus throughput (transactions per second inserted in the blockchain). 99

6.11 Average latency for each transaction in context-based consensus using PBFT and PoA. 101

6.12 Average transactions consensus throughput (transactions per second) using PBFT and PoA. 102

7.1 Healthcare integration (extracted from [213]) 110

LIST OF TABLES

1.1 Publications during the PhD research. 35

2.1 Overview of consensus algorithms for blockchains in IoT 45

2.2 Blockchain in IoT categorized using the layer-based model 52

3.1 Overview of consensus for blockchain in IoT 60

6.1 Description of 8 scenarios evaluated (from "A" to "I") 81

6.2 Performance Evaluation in 8 different scenarios (all times in milliseconds) . . 83

6.3 Test procedures. 91

6.4 Indicator **T1**: Time to add a new transaction (local gateway) 92

6.5 Indicator **T2**: Time to add a new transaction (remote gateway) 93

6.6 Indicator **T3**: Time to produce a new device block (before consensus) 93

6.7 Indicator **T4**: Time to add a new device block (after consensus) 93

6.8 Indicator **T5**: Time to add a new device block (after consensus) 93

6.9 Evaluated scenarios. 95

6.10 Evaluated configurations. 95

6.11 Evaluated approaches. 95

6.12 Latency (in milliseconds) to insert transaction. 97

6.13 Transactions throughput (transactions per second) using context-based consensus. 98

6.14 Evaluated scenarios using PBFT and PoA. 100

6.15 Latency (in milliseconds) to insert transaction using PBFT and PoA. 101

6.16 Transactions throughput (transactions per second) using context-based consensus. 102

LIST OF ALGORITHMS

4.1 Generic consensus algorithm 68
5.1 Perform transaction consensus - Configuration a 76
5.2 Send transactions pool - Configurations b and c 77
5.3 Perform transaction consensus - Configurations b and c 77

LIST OF ACRONYMS

AES – Advanced Encryption Scheme

CONSEG – Reliability and Security Group

CSIRO – Commonwealth Scientific and Industrial Research Organisation

DAG – Directed Acyclic Graph

dBFT – Delegated Byzantine Fault Tolerance

DVCC – Dual Vote Confirmation based Consensus

DDoS – Distributed Denial of Service

DoS – Denial of Service

DPoS – Distributed Proof-of-Stake

DNS – Domain Name System

ECDSA – Elliptic Curve Digital Signature Algorithm

FBA – Federated Byzantine Agreement

IBFT – Istanbul Byzantine Fault Tolerance

IIoT – Industrial Internet of Things

IoT – Internet of Things

M2M – Machine-to-Machine

MHF – Memory-Hard Functions

PBFT – Practical Byzantine Fault Tolerance

PoA – Proof-of-Authority

PoAh – Proof-of-Authentication

PoB – Proof-of-Burn

PoBT – Proof of Block and Trade

PoEWAL – Proof of Elapsed Work and Luck

PoP – Proof-of-Personhood

PoS – Proof-of-Stake

PoSpace – Proof-of-Space

PoW – Proof-of-Work

PUCRS – Pontifical Catholic University of Rio Grande do Sul

R2AC – Resilient & Robust Access Control

RSA – Rivest–Shamir–Adleman

UNSW – University of New South Wales

UOV – Unbalanced Oil and Vinegar

CONTENTS

1 INTRODUCTION . . . 27
1.1 MOTIVATION . . . 28
1.2 OBJECTIVES . . . 30
1.3 RESEARCH QUESTIONS AND HYPOTHESIS . . . 31
1.4 RESEARCH SCOPE AND ASSUMPTIONS . . . 31
1.5 RESEARCH DESIGN . . . 32
1.6 CONTRIBUTIONS . . . 33
1.7 DOCUMENT ORGANIZATION . . . 34

2 BLOCKCHAIN BACKGROUND . . . 37
2.1 BLOCKCHAIN ACCESS AND CONTROL . . . 38
2.2 BLOCKCHAIN ARCHITECTURES . . . 40
2.3 CONSENSUS ALGORITHMS . . . 41
2.4 DATA MANAGEMENT . . . 46
2.5 BLOCKCHAIN APPLICATIONS . . . 48
2.6 RESEARCH ON BLOCKCHAIN IN IOT . . . 49
2.7 CHAPTER SUMMARY . . . 53

3 RELATED WORK . . . 55
3.1 SEARCHING ENGINE AND PROCESS . . . 55
3.2 RESULTS & DISCUSSION . . . 56
3.3 CHAPTER SUMMARY . . . 61

4 APPENDABLE-BLOCK BLOCKCHAIN . . . 63
4.1 ARCHITECTURE . . . 65
4.2 DATA MODEL . . . 65
4.3 CONSENSUS . . . 67
4.4 SMART CONTRACTS . . . 68
4.5 MAIN OPEN ISSUES . . . 69
4.6 CHAPTER SUMMARY . . . 69

5 MULTI-LEVEL CONSENSUS ALGORITHM . . . 71
5.1 CONTEXT-BASED TRANSACTION CONSENSUS . . . 71

5.1.1 CONSENSUS FOR SINGLE TRANSACTIONS (CONFIGURATION A) 76
5.1.2 CONSENSUS FOR UNLIMITED TRANSACTIONS IN A TIME-WINDOW (CONFIGURATION B) 76
5.1.3 CONSENSUS FOR A FIXED MAXIMUM NUMBER OF TRANSACTIONS (CONFIGURATION C) 77
5.2 MULTIPLE CONSENSUS 78
5.3 SECURITY DISCUSSION 78
5.4 CHAPTER SUMMARY 79

6 EXPERIMENTAL EVALUATION 81
6.1 CONSENSUS FOR BLOCKS: INITIAL CONSENSUS EVALUATION 81
6.1.1 METRICS 82
6.1.2 RESULTS 82
6.2 CONSENSUS FOR BLOCKS: CONSENSUS ALGORITHMS COMPARISON . . 85
6.2.1 METRICS 86
6.2.2 RESULTS 86
6.3 CONSENSUS FOR BLOCKS:: GEO-DISTRIBUTED IOT 89
6.3.1 METRICS 91
6.3.2 RESULTS 91
6.4 MULTI-LEVEL CONSENSUS 94
6.4.1 METRICS 96
6.4.2 RESULTS 96
6.5 MULTI-LEVEL CONSENSUS: DIFFERENT CONSENSUS ALGORITHMS 99
6.5.1 METRICS 100
6.5.2 RESULTS 100
6.6 CHAPTER SUMMARY 102

7 DISCUSSIONS 105
7.1 THREATS TO VALIDITY AND LIMITATIONS 105
7.2 SECURITY ISSUES 106
7.3 PROMISING APPLICATIONS 108
7.3.1 SUPPLY CHAIN 108
7.3.2 DATA PRESERVATION 109
7.3.3 HEALTHCARE 109
7.3.4 OTHER APPLICATIONS 111

7.4 OUT OF SCOPE 111

8 FINAL CONSIDERATIONS 113

8.1 THESIS CONTRIBUTIONS 113

8.2 HYPOTHESIS FOUNDATION 114

8.3 DIRECTIONS FOR FUTURE WORK 116

REFERENCES 117

1. INTRODUCTION

Currently, smartbands, robot vacuum cleaners, IP cameras, and other smart devices became part of many people's life. The environment composed by these kind of devices - capable of processing information and communicating with other devices in order to take decisions - is called Internet of Things (IoT). However, these devices are vulnerable to different attacks, *e.g.*, getting access to health information from a personal smartband or using different smart devices to attack a web system. For example, the Mirai botnet [155] was a famous attack that used IoT devices to attack a Dynamic Domain Name System provider. In that attack, millions of devices were exploited (specially using default user and password) to produce this Distributed Denial of Service (DDoS) attack against important service providers, *e.g.*, Netflix, Twitter, and Reddit.

An IoT solution in general is composed of a myriad of devices, both in quantity and diversity. As a consequence, despite the benefits, there are several concerns about performance, safety, and security risks in these heterogeneous networks. Also, the fact that critical infrastructure, such as, energy grids and even human lives in the context of healthcare, can rely upon IoT devices make it even more important to guarantee the correct operation of such devices. Thereby, new challenges arise in this large, ever-increasing, and sensitive domain. Common challenges include overhead in computation, data management, and security [61]. Therefore, several researchers have proposed different ways to handle those challenges. These innovations have been proposed to deal with the challenges offered by IoT, including security challenges such as limitations to the hardware capacity, sensitivity of device information, or the use of devices in botnets [65].

After the popularization of blockchain frameworks, researchers proposed the adoption of blockchain in order to solve some of the security issues in IoT [65][68]. Some important benefits that a blockchain can provide to IoT networks include (although not limited to these):

- **Decentralized architecture**: the decentralized nature of a blockchain ensures that there is no single point of failure/attack in the system. Also, this can help to improve scalability of the solution.

- **Tamper resistance**: information contained in a blockchain is tamper resistant due to the linking through the hash of the blocks and the distributed consensus mechanism. This provides a permanent transaction and communication history for IoT.

- **Trustless communication**: some data exchange in IoT environments has nodes that do not know/trust each other. This kind of applications can have benefits from blockchain-based applications. Consensus algorithms, a mechanism that establishes agreement among untrusted nodes in the network and eliminates the necessity to use trusted intermediaries, can ensure that data are valid without trusting other nodes.

- **Transparency**: Information is inserted in the blockchain with a timestamp and creator's signature, turning this information transparent and traceable. Consequently, other participants can easily verify the information stored in a blockchain and make sure that the transactions are not tampered with or removed.

- **Smart contracts**: Blockchain can work as a distributed system for deploying and executing autonomous contracts. These smart contracts are executed when certain predefined conditions are met without the need for intermediaries. In IoT applications, smart contracts may be used to set the rules of the application, automate processes, and enable seamless communications and transactions between IoT devices and other entities.

To tackle the security issues different proposals investigate the use of the blockchain technology [43, 88, 145, 216, 250, 263]. One of them,the appendable-block blockchain was proposed by the Reliability and Security Group (CONSEG) [216, 232, 215] from the Pontifical Catholic University of Rio Grande do Sul (PUCRS). This blockchain was designed to present a blockchain solution that can solve some of the problems when used in IoT environments.

In this thesis, we propose to expand this blockchain and propose a consensus model to allow the usage of different consensus algorithms in multiple levels, allowing the parallel verification and insertion of the information produced by different nodes. The proposed solution will allow to use different consensus or configuration at the block level and at the transaction level. We intend with to improve availability and integrity of information.

To introduce these concepts, this chapter consists of six sections: Section 1.1 introduces and motivates this thesis. Section 1.2 presents and discusses the objectives of the thesis and, based on that, Section 1.3 describes our research questions and this thesis hypothesis. Section 1.5 presents the research design, which involves background, dynamic approach, and empirical studies. Section 1.6 presents the main contributions of this work. Finally, Section 1.7 presents the organization of the thesis.

1.1 Motivation

Despite the potential benefits of using blockchain technology for IoT, the adoption of this technology depends on a design that suits to IoT applications. High resource consumption, scalability, and slow transaction processing times are persisting problems for the integration of blockchain technologies for IoT. For example, blockchain provided by Bitcoin is not suitable for IoT devices: its size (storage required) and the time required to insert a new information (delay to create a block) is higher than expected in an IoT environment [289].

Different approaches were proposed [111, 85, 43, 250, 216] to use blockchain in IoT scenarios. One of the first commercial solutions for blockchain in IoT, IOTA [111] was proposed to perform Machine-to-Machine (M2M) payments using a different chain structure of blocks called Directed Acyclic Graph (DAG). In a different approach, Dorri *et al.* [85] focused on a lightweight architecture for blockchain, using overlays to control devices and to manage the block insertion in the blockchain. In a different approach, Boudguiga *et al.* [43] proposed a Blockchain-as-a-Service architecture for access control, where devices use protocols to access a blockchain authentication service. Also providing an access control solution, Novo [250] proposes the usage of smart contracts to manage access control for IoT devices. Focusing on a lightweight data management, Lunardi *et al.* [216] proposed an appendable-block structure using a gateway-based architecture. These different approaches focused on different aspects of blockchain (architecture, protocols, data management, and application) that contributed to the adoption of blockchain in IoT scenarios.

However, there are still open issues related to a lightweight consensus algorithm that can be used in IoT environments that considers devices hardware constraints and low latency requirements. Consequently, there is a lack of solutions that can be used in IoT scenarios composed by devices performing different tasks in different contexts, *e.g.*, sensors that both control the lightening and the access of a room, where different kinds of access and production of information are required. Also, there are problems related to how the information is inserted in the blockchain due to the consensus algorithm, which can lead to forks and inconsistency in the blockchain. Moreover, to the best of our knowledge, there are no discussion about consensus algorithms that can be adapted for different IoT contexts, producing better relation among security and performance (time response, throughput of transactions, reduced amount of processing required, etc.). It is important to note that we adopt context as a scenario or application for what IoT devices are used for.

In order to overcome this problem, this PhD thesis proposes a model for a multi-level consensus algorithms that considers different IoT contexts and provide parallelism to the insertion of transactions in the blockchain. This multi-level consensus is based on the two level of insertions in appendable-block blockchains: block level (or block header insertion) and transaction level (insertions of transactions in the block ledger). Also, the proposal is to allow the insertion using an adaptive mechanism for different contexts. This model is part of and will be evaluated through an appendable-block blockchain framework [215] (former called R2AC [216] and currently called SpeedyChain [232]).

1.2 Objectives

In order to provide an adaptable multi-level consensus that can consider both IoT context and relevant information in different applications, this thesis aims to propose a new consensus model for blockchains in IoT (particularly to appendable-block blockchains). The main goal of this model is to guarantee a better performance and security for different kinds of insertions in the blockchain. . For example, data insertion that can have a higher impact, *e.g.*, temperature of a water tank in industry, in the IoT environment can require a response time different from sensors that monitor the temperature in an office work space. Additionally, a consensus algorithm that can reduce or mitigate the presence of forks and inconsistencies in the blockchain will be provided.

Also, this model should support inter operation of different contexts, *i.e.*, exchanging data about a user/device that shares information in different applications. For example, information about vehicles that transport products between companies can have information in a blockchain (with a supply chain focus) used by both companies and also it can produce information in a blockchain to monitor sensors for vehicular networks (with a smart city focused application). Both blockchains can share specific information that could be used in both scenarios. Therefore, the following statement defines the main goal of this research:

> *"Propose a model for multi-level consensus algorithm that can consider different IoT contexts and applications, providing better relation among security and performance for IoT environments composed by different contexts."*

To achieve the main goal, the research must accomplish several secondary goals. These secondary goals, presented below, support the development of the model and the applications necessary for its validation to:

1. propose a context-based model for consensus algorithms considering the requirements (e.g., latency and throughput) of blockchains for IoT;

2. incorporate the proposal into the appendable-block blockchain framework [215], taking into account the particular data structure present on this kind of blockchain;

3. adapt insertion of blocks and transactions in the appendable-block blockchains in order to allow different consensus algorithms;

4. deploy emulated scenarios in order to evaluate and improve the proposed model.

1.3 Research Questions and Hypothesis

Current research about adoption of blockchain in IoT environments have not presented a solution that proposes an adaptive consensus for different users and applications. Consequently, there are some research questions that should be answered:

1. *How existent consensus algorithms perform in different IoT scenarios?* - Answering this question can help to understand the impact of each consensus algorithm in the workload of IoT devices, the number of messages exchanged, and the throughput of information inserted in the blockchain.

2. *What are the security issues associated with each consensus algorithm that can impact IoT?* - Answering this question can help to understand what is the impact for security and which problems can occur during deployment of an IoT architecture.

3. *How multiple consensus algorithms can be performed to allow parallel insertion in the blockchain?* - Answering this question can help to understand how to improve the parallelism in the insertion of data in the blockchain.

4. *How a consensus algorithm can be defined in order to be adapted for different contexts or to help in the interoperability among different applications?* - Answering this question can provide a model for a different consensus algorithm capable of being adapted in different IoT scenarios and to help to exchange information with different applications or contexts.

These research questions help to support and to evaluate the hypothesis of this thesis, described as follows: **It is possible to have a consensus algorithm that can handle different kinds of block insertion, allowing interoperability among different blockchain applications and a better relation among performance and security for IoT environments**.

1.4 Research Scope and Assumptions

The solution proposed on this thesis was designed to be used in hierarchical IoT environments. In particular, we used the layer-based IoT architectures proposed and adopted by different researchers [157, 66, 175, 17]. Consequently, we assume that the IoT architecture has gateways, which manage and control the communication and information produced by devices. We assume that devices present heterogeneous and constrained hardware.

Additionally, this thesis focus on ensuring availability and integrity of data shared by gateways. We assume that gateways have a hardware that is capable to perform cryptography and to manage a secure channels with devices. A discussion about hardware capabilities and requirements is provided in a previous work [216]. Also, devices tampering is not in the scope of this thesis. As a consequence, attacks on devices are not covered by this thesis.

1.5 Research Design

In order to verify the research hypothesis presented in the previous section, a model for a multi-level consensus algorithm for blockchain will be proposed, which can be adapted for different kinds of insertions of information and blocks in IoT ledgers. Also, a prototypical implementation using appendable-block blockchain will be used to evaluate the performance of the proposed model. The results of this evaluation can help to understand the different blockchain behaviors and how they perform in comparison to existent solutions. Consequently, it will be possible to verify if the hypothesis is valid or not. The main steps to prove the hypothesis are presented in Figure 1.1 and listed next:

1. To present an overview of the basic concepts that guide this thesis, such as blockchain and its main components (Chapter 2).

2. To define in which IoT scenarios the proposed model can be applied to and understand the limitations of these scenarios(Chapter 2 and Chapter 3).

3. To evaluate different consensus algorithms that can be used in blockchain for IoT scenarios (Chapter 2 and Chapter 3).

4. To propose a consensus model that can be adapted for different nodes/tasks in IoT to perform block insertion in blockchain (Chapter 4 and Chapter 5).

5. To implement a prototype for the proposed model using SpeedyChain framework (Chapter 4 and Chapter 5).

6. To perform experiments with the proposed solution in IoT scenarios defined previously (Chapter 6).

7. To compare the results of the proposed solution with previous version of the appendable-block blockchain and to discussion about limitations, security issues and potential applications of the proposed solution (Chapter 7).

8. To answer the research questions and to verify our hypothesis (Chapter 8).

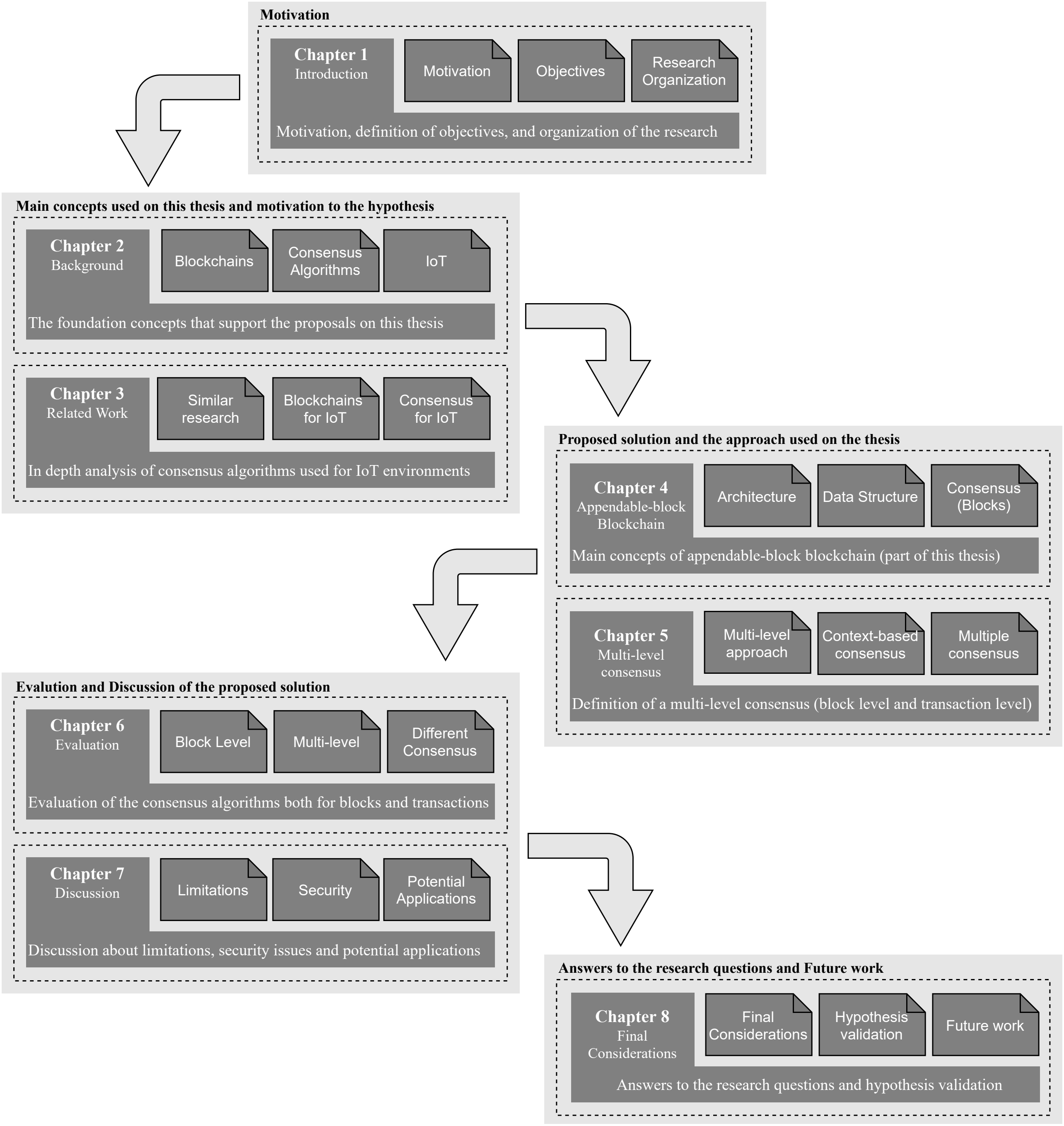

Motivation
Chapter 1
Introduction
Motivation
Objectives
Research Organization
Motivation, definition of objectives, and organization of the research
Main concepts used on this thesis and motivation to the hypothesis
Chapter 2
Background
Blockchains
Consensus Algorithms
IoT
The foundation concepts that support the proposals on this thesis
Chapter 3
Related Work
Similar research
Blockchains for IoT
Consensus for IoT
In depth analysis of consensus algorithms used for IoT environments
Proposed solution and the approach used on the thesis
Chapter 4
Appendable-block Blockchain
Architecture
Data Structure
Consensus (Blocks)
Main concepts of appendable-block blockchain (part of this thesis)
Chapter 5
Multi-level consensus
Multi-level approach
Context-based consensus
Multiple consensus
Definition of a multi-level consensus (block level and transaction level)
Evalution and Discussion of the proposed solution
Chapter 6
Evaluation
Block Level
Multi-level
Different Consensus
Evaluation of the consensus algorithms both for blocks and transactions
Chapter 7
Discussion
Limitations
Security
Potential Applications
Discussion about limitations, security issues and potential applications
Answers to the research questions and Future work
Chapter 8
Final Considerations
Final Considerations
Hypothesis validation
Future work
Answers to the research questions and hypothesis validation

Figure 1.1: Schematic overview of the thesis structure.

1.6 Contributions

We propose a model that can help blockchains to handle information from different contexts. This can help to adapt the blockchain to the application requirements. Thus, we propose a multi-level consensus mechanism that allows to use different consensus algorithms for different contexts and, at same time, to provide parallelism in the consensus procedure (*e.g.*, allowing the execution of a consensus algorithm for each context in parallel). Also, the SpeedyChain framework (an appendable-block blockchain framework designed by the CONSEG research group) was improved considering the advances obtained in the thesis. In accordance with our goals, the main contributions of this work are related to our feature interaction approach and they are listed next:

1. A study to investigate the state of the art about consensus algorithms used for blockchains in IoT;
2. The presentation of appendable-block blockchains and how consensus affects this blockchain;
3. The improvement of appendable-block blockchains to support different consensus algorithms for blockchains in IoT;
4. The proposal of context-based consensus algorithms at the transaction level on appendable-block blockchain;
5. The proposal of multi-level consensus model, allowing the adoption of different consensus algorithm for blocks and transactions;
6. Analysis of different experiments, considering different IoT scenarios to evaluate the impact of consensus algorithms over block insertion in the blockchain;
7. Context-based consensus evaluation and discussion on the adoption of different configurations.

Finally, as sub products of the thesis, we published our main findings in relevant venues presenting the advances of the research. Additionally, we collaborated with renowned researchers to improve the quality of the research in blockchain: Professor Aad van Moorsel (Newcastle University - UK) and Salil S. Kanhere (University of New South Wales - Australia).

These collaborations helped to guide the research of this thesis, as well to improve the definition of the scope of this work. Table 1.1 shows a list of publications performed during the PhD. This can help to present an overview of our contributions so far.

1.7 Document Organization

The remainder of this document is structured as follows. Chapter 2 presents the fundamental concepts about blockchain and its main aspects for IoT environments. Next, Chapter 3 introduces the state-of-the-art blockchain proposals for IoT, focusing on aspects related to consensus algorithms, performance, and how they are related to the thesis. Chapter 4 presents the appendable-block blockchain [215], a blockchain framework proposed by Reliability and Security Group (CONSEG) from Pontifical Catholic University of Rio Grande

Table 1.1: Publications during the PhD research.

Paper Title	**Venue**	**Year**
Publications related to the core of Thesis		
1. Context-based consensus for appendable-block blockchains [214]	**IEEE International Conference on Blockchain**	*2020*
2. Appendable-block Blockchain Evaluation over Geographically-Distributed IoT Networks [76]	**IEEE International Black Sea Conference on Communications and Networking**	*2020*
3. Impact of consensus on appendable-block blockchain for IoT [215]	**EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services**	*2019*
4. Distributed access control on IoT ledger-based architecture [216]	**IEEE/IFIP Network Operations and Management Symposium**	*2018*
Publications related to the Thesis		
5. Context-based smart contracts for appendable-block blockchains [251]	**IEEE International Conference on Blockchain and Cryptocurrency**	*2020*
6. Data-Driven Model-Based Analysis of the Ethereum Verifier's Dilemma [8]	**IEEE/IFIP International Conference on Dependable Systems and Networks**	*2020*
7. Modelo de negócio para saúde colaborativa usando smart contracts: caso Token-Health [44]	**Revista Brasileira de Computação Aplicada**	*2020*
8. A journey in applying blockchain for cyber-physical systems [79]	**International Conference on COMmunication Systems NETworkS**	*2020*
9. Blockchain technologies for IoT [80]	**Advanced Applications of Blockchain Technology**	*2020*
10. Estruturando diferentes aplicações com Blockchain [213]	**SBC Horizontes**	*2020*
11. Performance and cost evaluation of smart contracts in collaborative health care environments [217]	**International Conference for Internet Technology and Secured Transactions**	*2019*
12. Avaliação do uso de Smart Contracts para Sistema de Saúde Colaborativa [45]	**Escola Regional de Redes de Computadores**	*2019*
13. SpeedyChain: A framework for decoupling data from blockchain for smart cities [232]	**EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services**	*2018*
14. Dependable IoT using blockchain-based technology [371]	**Latin-American Symposium on Dependable Computing**	*2018*
Other publications		
15. Gerenciamento de incidentes em SIEM seguindo ITIL[244]	**Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação**	*2020*
16. Pentest on an Internet Mobile App: A Case Study using Tramonto [31]	**International Conference for Internet Technology and Secured Transactions**	*2019*
17. Extração e Gerenciamento de Incidentes em SIEM [243]	**Escola Regional de Redes de Computadores**	*2019*
18. Lightweight IPS for port scan in OpenFlow SDN networks [245]	**IEEE/IFIP Network Operations and Management Symposium Workshops**	*2018*

do Sul (PUCRS). This chapter also discusses preliminary results and how the thesis expands and improves the appendabable-block blockchain. Chapter 5 presents the multi-level consensus algorithm, which discusses the different levels where the consensus can be performed. Next, in Chapter 6, different evaluations are performed, showing the results to the performance of consensus in different levels. Chapter 7 presents a discussion about the performed evaluation, as well as the threats to validity, security aspects, topics out of scope of this thesis, and possible applications of appendable-block blockchains. Finally, Chapter 8 presents the final remarks of this thesis proposal, presenting the main research contributions of the thesis.

2. BLOCKCHAIN BACKGROUND

Early in 2008, an “entity” published a paper through the Satoshi Nakamoto pseudonym, describing a cryptocurrency called Bitcoin [242]. The paper presented a system running over a peer-to-peer (P2P) network that allows two different accounts (represented by their public keys) to exchange (crypto)currency directly, without using a third party to mediate the operation.

In Bitcoin, transactions involving two parties are created and stored in a block. Thus, each block contains a set of transactions. On one hand, these transactions are organized in a Merkle tree through a binary hash chain [187]. On the other hand, blocks are ordered and sequentially connected through the previous block hash value (as presented in Figure 2.1). The blockchain concept is based on this block link strategy, which differs from the binary hash chain used in the Merkle tree.

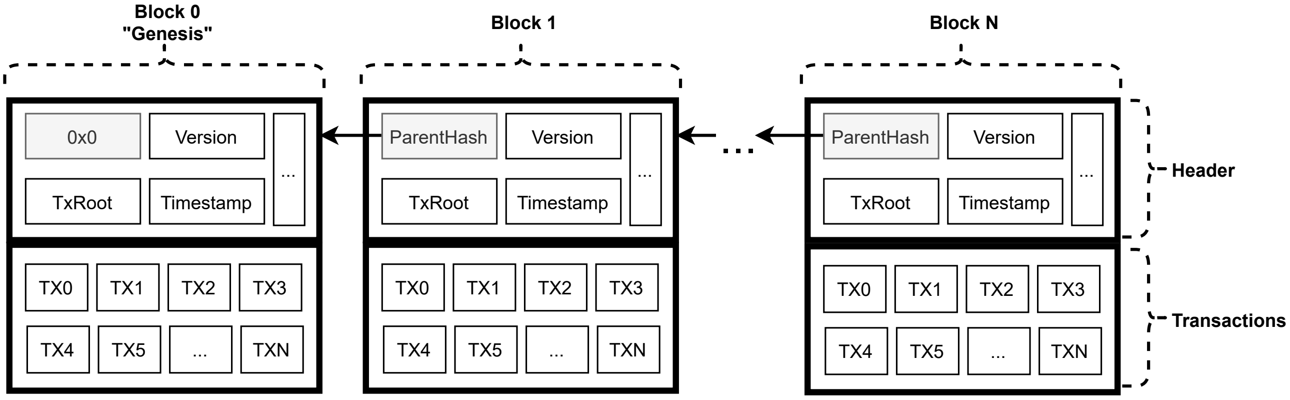


Figure 2.1: Blockchain example.

Before a block is inserted into the blockchain, first a consensus algorithm has to be executed. There are different types of consensus algorithms that might be executed before inserting a transaction in a block, for example, Proof-of-Work (PoW), Proof-of-Stake (PoS) or Practical Byzantine Fault Tolerance (PBFT) [65]. Bitcoin, for example, uses PoW consensus algorithm (sometimes called a mining process), while BlackCoin [339] secures its blockchain through the PoS algorithm. BlackCoin uses a process called minting, instead of mining, to validate transactions based on the amount of coins that the peers own.

Initially, blockchain was used as distributed public ledger for cryptocurrencies, for example, in Bitcoin or Litecoin [208]. Lately, new concepts were added to blockchain making its applicability wider. For example, blockchains, such as Ethereum [109], introduced the smart contract concept, which allows a user to write a piece of code and add it to the blockchain. Furthermore, in terms of application areas, blockchains are also being used to make some operations faster than they were before. For example, the Ripple blockchain [298] consists of a decentralized consortium and permissioned ledger used in the banking system, and through that, changed the way banks are exchanging money.

Hyperledger was presented by Linux Foundation, in December 2015, as an umbrella for different blockchain initiatives. Its main focus is to define a cross-industry open standard platform for distributed ledgers. Different types of blockchains can be implemented under Hyperledger, even to provide blockchain infrastructure as a service, *i.e.* Blockchain-as-a-Service (BaaS), and over this infrastructure developers are encouraged to create new applications. One implementation from Hyperledger, the Hyperledger Fabric [49] proposes a distributed ledger platform for running smart contracts. Fabric architecture is modular and uses PBFT consensus algorithm, however the consensus algorithm is executed only to *validate peers* that are also responsible for maintaining the ledger. Additionally, there are peers called *non-validating* in charge of connecting different clients and validating peer transactions. Thus, Hyperledger Fabric [112] works as a permissioned blockchain and, due to this characteristic, its chain size and length depends on its configuration and purpose [240].

The IOTA blockchain proposal is focused to attend IoT needs [111]. Its main difference from the original blockchain is how data are organized. While in the original blockchain blocks are organized and linked sequentially, IOTA uses a Directed Acyclic Graph (DAG) to link blocks. The IOTA DAG is called Tangle and organizes blocks in a graph structure. Before a new block is inserted into the DAG, the insertion algorithm chooses two random unconfirmed blocks (blocks that are in the DAG but were not confirmed yet), confirms the PoW of these two blocks, and the new block points to these two, now confirmed, blocks. Important to mention that this new block, which has now been inserted into the DAG, will be confirmed when another block is inserted into the IOTA DAG. Transactions, in a block, can contain monetary values (similar to a regular cryptocurrency transaction) or a zero value, in that case the transaction holds, then, any other type of information. Although IOTA has been used for IoT, it cannot be applied to low processing power IoT devices due to the PoW consensus algorithm [107].

Several other blockchain initiatives have been developed in the past years, for example SpeedyChain [232], Waves [346], Stellar [322], and it is very likely that new ones will be designed in the near future. It is important to mention that there is no blockchain standard yet, but several researchers are discussing - in venues such as IEEE Blockchain Summits and conferences - the way blockchain is changing the way applications will be developed and how they will interoperate.

2.1 Blockchain Access and Control

Initially, blockchain was proposed by Bitcoin to be used as a free access environment, *i.e.*, any user could access the blockchain and perform the same operations. However, in the last years, different approaches proposed new types of access to a blockchain, especially to blockchains that contain sensitive data. Consequently, today there are three different categories that can be used to classify the access of a blockchain: public, private, and consortium [321].

In a public blockchain, anyone can join the network and access transaction history recorded on the blockchain. Every node in the network has a copy of the distributed ledger, which is generated by a distributed consensus mechanism. Usually, public blockchains are resilient against attacks and failures due to the redundancy in the network and the consensus mechanism. However, the distributed consensus mechanism causes latency, lower network throughput and inefficiency. Network participants may earn economic incentives for contributing to the consensus mechanism such as Proof-of-Work (PoW) or Proof-of-Stake (PoS). Examples of public blockchains include Bitcoin, Ethereum, and Litecoin.

In a private blockchain, a single organization controls the blockchain by determining the rules of the network and access permissions. Also, the trust is centralized at the owner (that can be composed by a single or multiple nodes). However, some private blockchain proposals can make incentives to the participants to perform the consensus in order to avoid infrastructure costs. For example, a company can run a private instance of Ethereum, controlling the initial difficulty of PoW (number of bits zero in the beginning of the target hash), but not centralizing the consensus (any participant can perform the consensus and receive coins for the mining process). Additionally, the company can define access permissions for every node, leading to improvements on privacy of the transactions. The private blockchain architecture is more suitable for companies or government applications.

Consortium blockchains are used when a group of participants/companies interact and both the consensus mechanism and maintenance of the blockchain are governed by a predetermined group of network participants. The access mechanism of the consortium blockchain defines the rules of access to the blockchain information. Similar to private blockchains, consortium blockchains are more efficient and provide higher transaction privacy than public blockchains. Consortium blockchains are suitable for applications that involve multiple companies or agencies.

In relation to control and permissions to perform actions in the blockchain, there are two different important categories that are used: permissionless and permissioned blockchains [105]. Some works make confusions about public and permissionless or private and permissioned. However, they are not synonymous. For example, a company can use a private instance of Ethereum in a permissionless fashion. Public, consortium and private are related to access to the blockchain, while permissioned and permissionless are related to the control in the blockchain.

In permissionless blockchains any node can join the blockchain and participate in creating and verifying transactions, contributing to the consensus mechanism. Usually, permissionless blockchains use incentives for establishing consensus and network participants. It leads to blockchains with no centralized control leading to a higher resilience against attacks and censorship. The network operation is transparent so that network participants know how the blockchain works and how consensus is achieved.

Differently, in permissioned blockchains authorized nodes are predefined and they have permissions to participate in the blockchain. Permissioned blockchains allow an organization or a group of organizations to record communications, events, and transactions in an immutable manner. Also, the blockchain is controlled by an organization or a group of organizations, and the level of decentralization depends on the structure of the network interactions. Consequently, in this kind of blockchain consensus mechanisms can be less computationally expensive. This can improve scalability and throughput of insertion of information when compared to the permissionless blockchains.

2.2 Blockchain Architectures

The most common adopted architecture in a blockchain is a completely decentralized architecture, where each node can be a full node, *i.e.*, every device communicate directly to other devices in the network to update the blockchain. This kind of architecture is adopted by most of the public blockchains, such as Bitcoin and Ethereum. However, it requires that all devices have enough computing power, battery, memory, and storage for maintaining the blockchain [303] [354]. In IoT scenarios with heterogeneous devices - with different hardware capabilities and limitations - this kind of architecture is hard to be used without compromising the security of the devices, especially against Denial-of-Service (DoS) attacks [155].

In order to mitigate the hardware strict requirements, some proposals adopted a hierarchical P2P architecture (Gateway-based architecture in Figure 2.2). In that environment, fullnodes (also called gateways or overlays) are used to maintain a blockchain with devices information[85][216]. This kind of solution leads to a reduction on the traffic generated through Smart Environments networks and decreases the vulnerability of the devices. Moreover, some proposals execute smart contracts on the blockchain fullnodes, thus reducing processing on limited devices. However, these architectures are more susceptible to the Eclipse attack [308], *i.e.*, when a malicious gateway monopolizes device incoming and outcoming connections.

Another architecture, called Blockchain-as-a-Service [43][289], separates the nodes that control the blockchain from the IoT network (see Figure 2.2). Consequently, all the processing of the blockchain can be performed by a third-party infrastructure, reducing the computing cost for IoT devices. For example, in the work presented by Boudguiga *et al.* [43], IoT devices availability is updated in the blockchain through encrypted messages. However, the trust is delegated to a third-party authentication authority, *i.e.*, the IoT devices are susceptible to security issues if the third-party authority or its API is compromised.

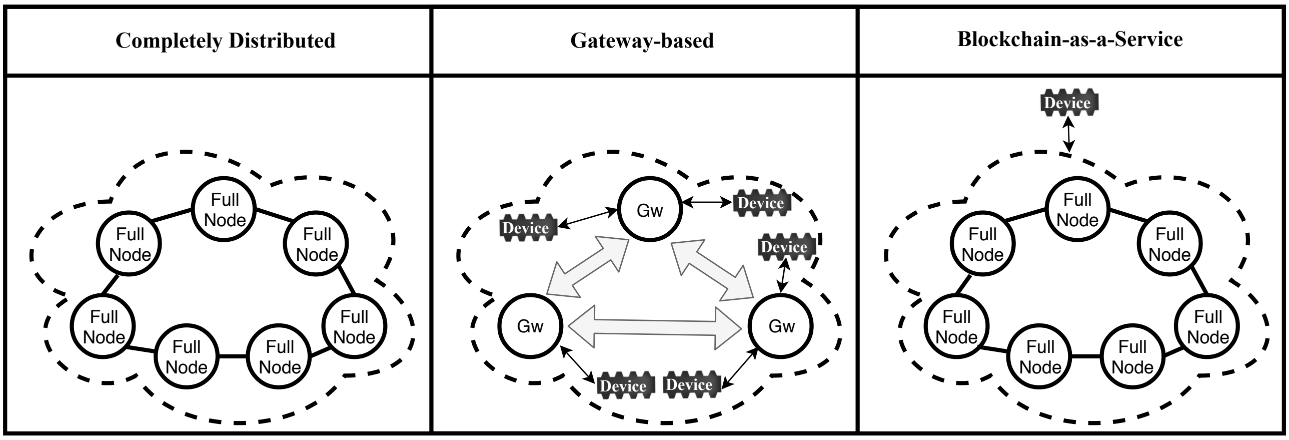


Figure 2.2: Main blockchain architectures for IoT (extracted from [371].)

Each architecture has its own advantages and disadvantages. The selected consensus depends on the purpose and the requirements of each IoT environment. In recent research, Gateway-based and Blockchain-as-a-Service architectures were presented as solution for limited hardware (which are more susceptible to DoS/DDoS attacks). However, these kinds of scenarios present problems related to centralization and a need of trust in a node or a set of nodes. Many discussions were made in the last few years - both in academia and industry - for a standard or pattern for IoT architecture through protocols and other definitions.

2.3 Consensus Algorithms

Consensus mechanisms are required to achieve agreement on the state of the distributed ledger shared by the nodes and to ensure security when there is no central authority to control the state of the ledger. A blockchain guarantees that the information stored in the ledger is unaltered by linking it to previously stored information on the blockchain and validating the authenticity of the information based on digital signatures.

In order to achieve this, distributed consensus algorithms can be performed by the nodes, which do not necessarily trust each other. The consensus algorithm prevents malicious nodes of producing fake transactions and blocks and ensures randomness among the nodes that perform the consensus. Most of the existing consensus algorithms demand the participating nodes to spend computational resources to solve a puzzle to be able to insert a new block. To prevent malicious miners from flooding the network with fake blocks, the consensus algorithms limit the number of blocks that can be generated in specific time periods by adjusting the difficulty of solving the puzzle. The unreliable nature of the peer environment where a blockchain is executed should be considered in choosing the appropriate consensus algorithm to be performed.

A blockchain should guarantee that the information stored is trusted and linked with other information in the blockchain, and that a set of peers certify the authenticity of the information based on digital signatures. In order to achieve this, consensus algorithms can be performed through the nodes - that might not be trusted ones. The unreliable peer environment, where a blockchain is executed, should be considered to choose the consensus algorithm to be performed. Another key aspect to be considered is related to the hardware constraints in IoT devices, such as computing power, memory, and storage.

As mentioned before, there are different consensus algorithms available to be used in blockchains to guarantee that a new block inserted in the blockchain is valid. The main algorithms (although not limited to these) are Proof-of-Work (PoW), Proof-of-Stake (PoS), Proof-of-Space (PoSpace), Practical Byzantine Fault Tolerance (PBFT), delegated Byzantine Fault Tolerance (dBFT), Federated Byzantine Agreement (FBA), Istanbul Byzantine Fault Tolerance (IBFT), and Raft.

Proof-of-Work (PoW) consists in solving a resourcing consuming puzzle to avoid an overload of information to be created [93]. Usually, the task is composed by the generation of a hash for the data contained in the block varying a nonce value in order to obtain at least a predefined number of bits equal to zero at the beginning of the generated hash value. After the block is created, it is broadcast to other peers, and it can be easily verified (compare the received hash with the block hash). Also, the difficulty of the work (*e.g.*, higher number of zeros in beginning of the target hash) can be adjusted over the time. There are different implementations of PoW, both varying the algorithm to perform the "work" - Bitcoin uses SHA-256 to perform the hash, while Litecoin uses Scrypt and IOTA uses Tangle. Although IOTA [111] is the most prominent adopter of PoW consensus for blockchain in IoT, these algorithm tends to have a high impact on battery and processing of limited devices. Additionally, PoW is mostly used in rewarded-based consensus - miners receive coins to perform the "work".

Proof-of-Stake (PoS) is an alternative to the PoW algorithm. In order to reduce the difficulty of the block generation task, PoS uses a random selection of nodes based on wealth or aging of coins [171]. PoS preserves a single branch, as only a single node is responsible for producing a block. Although PoS has the objective to reduce the processing needed to create a block, to the best of our knowledge, there is not a blockchain for IoT using PoS consensus. One problem with PoS in IoT is that it can lead to a centralization of the consensus in few nodes, which creates a single point of attack, partially centralizes trust, and limits scalability.

Proof-of-Space (PoSpace) was proposed to ensure a more energy-efficient solution than PoW. PoSpace can focus both on transient or persistent space. Usually, PoSpace uses Memory-Hard Functions (MHF) or proof of secure erasure functions, which require memory/space intensive computations. One advantage of this method is that a verifier only needs a small amount of space and computation to check the results produced by the node that produced the block [94]. Although it has a lower energy consumption, it requires a higher memory or storage space in the nodes that contribute to the consensus mechanism. Consequently, PoSpace is not suitable for IoT applications with resource-constrained devices, where memory and storage are limited.

Practical Byzantine Fault Tolerance (PBFT) is the ability of a distributed network to correctly reach consensus when a subset of the nodes are faulty or malicious. When there are f faulty nodes in the network, PBFT requires 3f+1 nodes to correctly reach consensus. When a new block is created, a leader node is selected. Then the leader node starts the consensus mechanism by sending the block to the active validation nodes in the network for validation. If more than 2/3 of the active validation nodes vote to validate the new block, the block is appended to the blockchain [53]. PBFT has been used by many blockchain proposals for IoT for the last few years. However, PBFT mechanism suffers from poor scalability. In a large network, the number of messages and the waiting time for node responses can be high. Additionally, in a dynamic P2P scenario, where nodes frequently leave and rejoin the network, achieving consensus becomes difficult as active nodes can change their status during the consensus.

Delegated Byzantine Fault Tolerance (dBFT), similar to PBFT, achieves consensus on new information based on votes. However, in dBFT, validators (nodes that validate and vote) are elected by the requester for each consensus. If the requester does not trust in a chosen validator, the first one can elect another node as validator for the next consensus procedure. Then, the validators choose a node to be the leader (Elected Validator “A” in Figure 2.3) that will create the block and start the consensus procedure (step 2 in Figure 2.3). Consequently, just a small subset of the nodes are used to perform the consensus in dBFT [71]. When more than 2/3 of elected nodes validate the information, it is considered valid (in step 3, nodes A, B and C vote positively, so the new block is considered valid). Neo is one of first blockchains that adopted dBFT-based consensus algorithm. This solution can solve the scaling issue of PBFT, reducing the number of nodes that will perform the consensus. However, in dynamic IoT scenarios it can still present a problem when elected nodes are not reliable.

Federated Byzantine Agreement (FBA), similar to PBFT, achieves consensus based on a set of positive votes. Each node knows a set of other “important” nodes (also called Quorum slice) that are predefined by each node based on arbitrary criteria such as financial arrangements. When performing the consensus, a requester node initiates (*e.g.*, node A in Figure 2.4) the consensus algorithm and waits for the important nodes to validate (nodes B and C in Figure 2.4) the new information. Those important nodes will validate the information when their Quorum slices validate as well, *e.g.*, B will validate when D and E Quorum slices validate as well. Eventually, a enough number of the nodes in the network (also called as Quorum) validate the information and it can be inserted in the blockchain [226]. In this algorithm, only a subset of the network is used to perform the consensus,

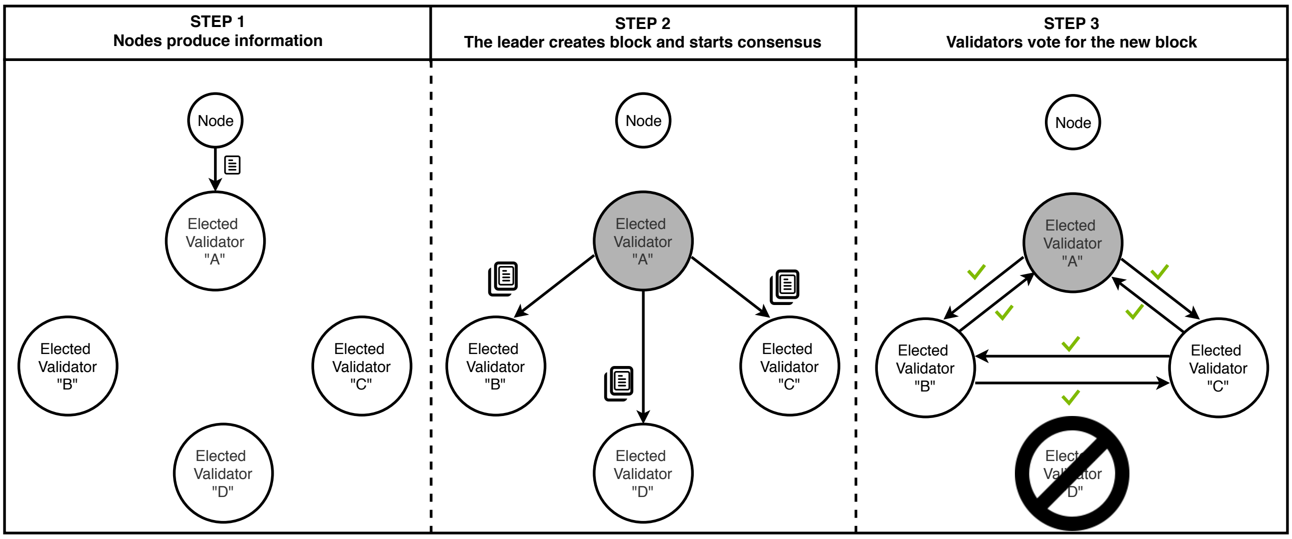


Figure 2.3: Inserting new block using dBFT (extracted from [80]).

i.e., it is performed by groups (federations). Stellar is one of the most prominent adopter of this algorithm.

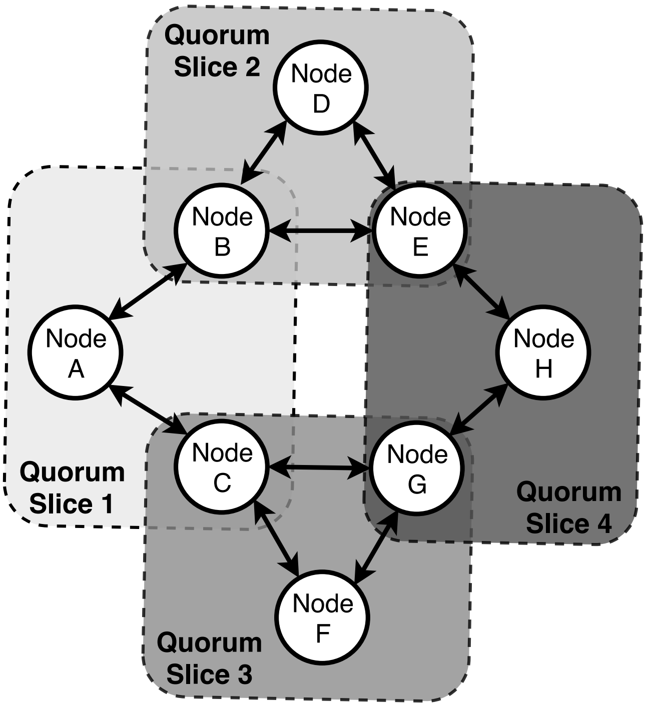


Figure 2.4: Quorum slices used in FBA (extracted from [80]).

Istanbul Byzantine Fault Tolerance (IBFT) also requires that more than 2/3 of active nodes in the blockchain to validate the new information to be inserted. However, the proposer (node that start/control the consensus procedure) can be selected in a “round robin” way. The proposer node starts the consensus without having to choose a leader. IBFT is considered an adaption of PBFT and can be used to produce new blocks in a constant rate by different nodes [27]. Due to the insertion of blocks in a constant rate, empty blocks (with no transactions/information) can be created. In IoT scenarios, these empty blocks can lead to unnecessary overhead.

RAFT also needs 3f+1 nodes to be setup in the network to have the capability to tolerate f faulty nodes and has a leader that starts the validation. However, unlike IBFT, it does not create empty blocks and the time to change the leader is randomized. The leader handles all node requests and sends them to all the followers (other nodes in the network) to perform the validation [253]. The main issue with RAFT is that all the information is serialized through a leader that will manage the consensus through a randomized amount of time. In an untrusted IoT scenario, this leader can be overloaded by requests or can be targeted by malicious nodes.

The adoption of the consensus depends mainly on three factors: the architecture in which it will be used, the hardware requirements and the attack vector that is intended to be mitigated. Consequently, the number of nodes and the processing overhead are important issues to be considered when choosing the consensus algorithm. Table 2.1 presents an overview of the consensus algorithms related to the access to the blockchain, blockchain control approach, and positive and negative aspects for IoT.

Table 2.1: Overview of consensus algorithms for blockchains in IoT

Consensus Algorithm	**Access**	**Control**	**Positive aspects for IoT**	**Negative Aspects for IoT**
PoW	Public	Permissionless	Few messages exchanged to achieve consensus	High energy and computing required
PoS	Public	Permissionless	Scalable, not high power consuming	Overload in few nodes can impact in operations of the blockchain
PoSpace	Public	Permissionless	Not high power consuming	Requires high amount of memory/storage
PBFT	Private or Consortium	Permissionless	Less hardware and energy requirements	Not scalable
dBFT	Private or Consortium	Permissioned	Scalable and Less hardware/energy requirements	Problems in dynamic scenarios
FBA	Private or Consortium	Permissioned	Scalable and Less hardware/energy requirements	“important” nodes should be trusted
IBFT	Private or Consortium	Permissionless	Less hardware and energy requirements	Not scalable, produce empty blocks
Raft	Private or Consortium	Permissionless	Less hardware and energy requirements	Not scalable, serialization of requests

There are other consensus algorithms that can be adopted in appendable-block blockchains, such as BFT-SMART, Proof-of-Authority (PoA), Proof-of-Personhood (PoP), Proof-of-Burn (PoB), and Tendermint [105]. However, each consensus algorithm brings a new discussion both on performance and possible security issues that should be addressed.

Also, it is important to note that consensus algorithms that use rewards as incentives are not suitable for appendable-block blockchain (there is no native cryptocurrency).

A seminal discussion about consensus algorithms for blockchain in IoT is presented in Christidis [65] research. This research investigated different consensus algorithms and concluded that the mechanism used in blockchains depends on two main factors: the architecture in which it will be used and the attack vector that it is intended to mitigate. Consequently, the number of nodes and the processing overhead are important issues to be considered when choosing the consensus algorithm.

Although a consensus algorithm is a key aspect of a blockchain, most of the blockchain proposals for IoT did not discuss or evaluate their usage [263][250][216][353][232][85]. It is an important issue to tackle, as dependability, security and performance can be affected when a consensus algorithm is introduced in the blockchain.

2.4 Data Management

One important aspect to be considered in blockchains for IoT is how the data is structured in the blockchain. Additionally, there are differences when considering which cryptography algorithm is adopted for what purpose in the blockchain. Based on existent approaches, there are differences related to the data layer adopted by different blockchain solutions [242][86][216][232][110][111][112].

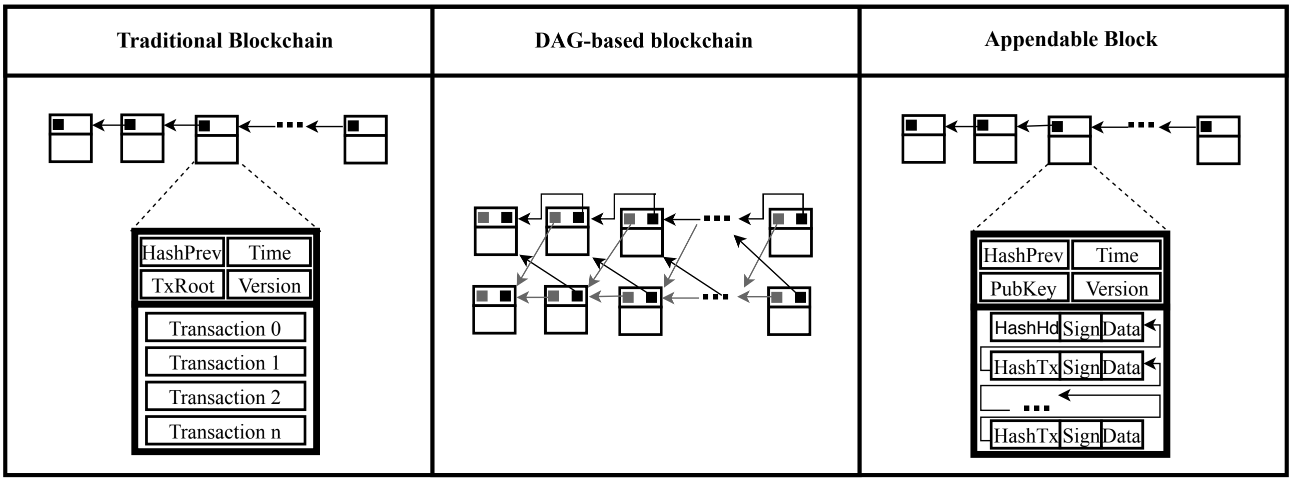


Figure 2.5: Block structure (extracted from [371]).

Initial blockchain proposals structured their data through blocks that contain both header and transactions. The main difference is that transactions are the data structure where some information is stored. Alternatively, the block is used to store the information required to create the chain, *i.e.*, the structure is defined to support the link between blocks, where each block has a reference to its direct ancestral. Transactions are organized inside each block and could follow different arrangements, such as, Merkle tree [242], linked list [242], Direct Acyclic Graph (DAG) [111], contained immutable [242][111][112], appendable data inside a block [216][232] or erasable [86] blocks. Some examples of how blocks and transactions are organized in a blockchain are described next:

- The Bitcoin blockchain uses a Merkle tree to arrange transactions inside a block (see Figure 2.6). In this case each block contains an immutable amount of transactions, *i.e.*, once the hash of transactions are inserted in the Merkle tree inside a block, no further action (adding or removing a transaction) can be taken (see Figure 2.5 - Traditional Blockchain). Also, the Bitcoin blockchain uses a linked list to arrange the blocks list, keeping the sequence, and linking a previous block to the previous block.
- In SpeedyChain the transactions are stored inside blocks, where the first transaction is linked to the block header (through block header hash), while other transactions are linked to the previous transaction, through the hash of the previous block (see Figure 2.5 - Appendable Block). Thus, this leads to an appendable block, *i.e.*, a block that can still receive new transactions after inserted into the blockchain. In SpeedyChain, blocks are not immutable, however after a new block header is created, it is linked by hash, preserving its integrity. Also, a new information inserted in block ledger is both signed and hashed, guaranteeing both integrity and non repudiation.
- IOTA, uses a DAG structure (called Tangle) [111] to arrange and link block in its blockchain. This DAG consists of a graph without direct cycles as shown in Figure 2.5.

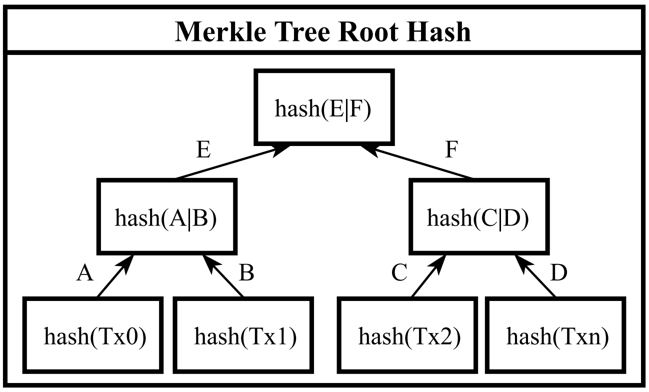


Figure 2.6: Merkle tree structure (extraxted from [371]).

In order to support these data arrangements and still ensure security - data integrity and privacy - the cryptography algorithms are a central piece in this structure. Among algorithms applied to blockchains we could highlight asymmetric and symmetric ciphers to guarantee privacy, and hash functions to keep data integrity. For example, the Bitcoin blockchain applies the Elliptic Curve Digital Signature Algorithm (ECDSA) [159] in order to create the public/private key pair, which is used as a wallet address, and for the block hash, it uses the SHA-2 [129] hash function. Ethereum [109], instead of using the SHA-2, uses a preliminary version of SHA-3 (based on original Keccak algorithm) [260]. SpeedyChain [232] relies in the SHA-2 hash function and Rivest–Shamir–Adleman (RSA) [282] for the asymmetric cryptography algorithm.

2.5 Blockchain Applications

The “Application” layer is responsible for managing the different applications that uses a blockchain, for example, cryptocurrencies or data management. Currently, there are several available cryptocurrencies [261]. Usually, a coin is used in transactions to represent a cryptocurrency exchange between two users in a blockchain. Basically, there are two manners to obtain a coin: acquiring (through a transaction) from a user or mining. In the IoT context, IOTA (and its coin MIOTA) [111] was created to be used for M2M payments. IOTA is, currently, the most known and representative cryptocurrency for IoT [67]. Coins can also be used in different applications, for example, Smart Grids [354].

Similarly to the coin concept, Tokens runs over an existing blockchain. Usually, Tokens are instantiated as a fraction of the main currency and represent an asset or utility. It is important to establish a main difference between Coins and Tokens: while, in general, a Coin is a currency and runs its own blockchain, a Token can be a currency or represent assets in the blockchain. So far the most popular Token definition and implementation is the one which runs on the Ethereum platform, where some standards were defined, the most popular standards are ERC-20 and ERC-721 [109].

Using decentralization, resilience, and transparency provided by blockchain, decentralized applications (dApps) became popular in blockchain applications [349]. In the blockchain, if a peer fails, any other available peer has an updated view of the blockchain. This property helps to improve the resilience of the application. Also, blockchain concept was conceived to not present a centralized control, and therefore, the application that runs on a blockchain infrastructure can improve its dependability [21]. And finally, any node can verify the information and the history of the executions in the blockchain.

Another concept that has opened new perspectives for developing dApps is smart contracts. A smart contract allows the execution of a code inside a blockchain without a centralized control. Once in the blockchain the smart contract will be permanent and it cannot be altered [65]. Any flaw in the logic of the contract will persist with it without the possibility of an update. A mechanism to disable a smart contract may be included in the development phase to provide flexibility for avoiding identified bugs. In that case the contract will still persist in the blockchain but the logic of it will prevent it from doing any operation. Additionally, after insertion in the blockchain, all smart contracts are available and are known to the other nodes in the network. Also, values in variables stored within a smart contract are available to everyone in the blockchain. Consequently, smart contract content can be a concern for privacy-sensitive content.

Since business logic can be applied to a smart contract, it has an ample scope of applications, such as resource allocation, traceability, and auditability [65]. For example, a smart contract can be deployed and made accessible to a specific manufacturer of IoT devices. In the smart contract, the device can check the last version of firmware available and receive a hash of the newest version. If necessary, it can update itself to the last firmware. Another usage is related to coin exchange. For example, on blockchains that provide cryptocurrencies, a device can sell services to other devices (*e.g.*, storage or information from sensors). Also, it can help to manage an IoT network with a list of devices and their permissions [64]. This list is dynamic so new devices can be removed or added and permissions can be changed. In this use case, non-authorized entities are not allowed to interact with the devices in the network. Additionally, smart contracts can perform a load balancing algorithm, analyzing the workload on the devices and assigning new tasks for idle devices.

Currently, smart grids are the most explored IoT application for smart contracts [248][124]. One of the drivers is a push for a decentralized market, where energy can be freely produced and consumed without trusted third parties. In this environment smart contracts can be used as market brokers where users of the network can offer excess or buy energy from the network in an automated way without the need for a central authority. An IoT device - controlling the energy grid of a house - can participate in a blockchain network and bid for energy in the region energy network. IoT devices controlling the energy grid can route the energy between the producer and consumer.The use of smart contracts and IoT to control access to energy in a house improves also auditability and increased transparency [115].

Moreover, industrial IoT devices can be used to control the production in an automated way. These devices can benefit from the use of blockchain for management and control. Smart contracts can be used to allow machine-to-machine communication, avoiding the need of human intervention in some extend. A smart contract can be used to control the access and permissions of IoT devices and users, increasing the security and giving a more transparent process where all activities can be audited. This approach is not exclusive to Industry 4.0, as any IoT network in most context can benefit from management solutions in the blockchain.

2.6 Research on Blockchain in IoT

In recent years, several researchers [88] [263] [43] [216] [250] have proposed different solutions that use the blockchain technology in IoT to solve security issues. Also, we can observe an increased interest in this field, as represented in Figure 2.7 by the increased number of published papers about blockchain in IoT from 2015 to 2020.

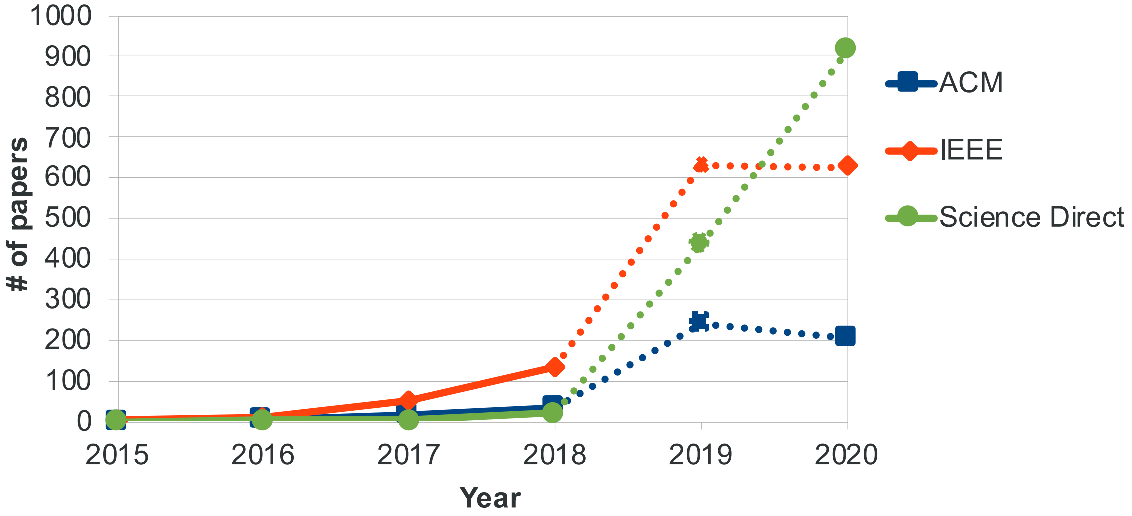


Figure 2.7: Increasing number of papers about blockchain in IoT.

The search string *"blockchain" AND ("iot" OR "internet of things" or "internet-of-things")* on IEEEXplore[1] and its equivalent search string on ACM Library[2] and Science Direct[3] databases were used to produce the graph in Figure 2.7. In order to refine the search, papers with 1 or 2 pages (usually extended abstracts and posters), duplicates (papers from IEEE were kept when conference was supported both by ACM and IEEE), and papers about subjects not related to blockchains in IoT (for example, using blockchain or IoT just as example of new technologies) were excluded.

After the refinement process, only one paper was published in 2015 (present in IEEExplore), 11 papers in 2016 (3 in ACM, 7 in IEEEXplore and 1 in Science Direct), 70 papers in 2017, and 184 in 2018 (133 in IEEExplore)[4]. We can also observe a large amount of works published in 2019 (1310 papers) and in 2020 (1752 papers). This represents the increasing effort of the academy and industry to use and discuss the adoption of blockchain in IoT.

After reading the abstracts of all of these papers, we selected the ones that presented relevant contributions for blockchains in IoT (they will be presented in details in the next sections and in Chapter 3). Some works are seminal to the blockchains in IoT. For example, propose novel blockchain architectures [85] [43], while others propose innovative blockchain data management solutions [216] [250].

For example, Dorri *et al.* [85] proposed a lightweight blockchain architecture for IoT as an authorization mechanism to access data in a Smart Home. Basically, the devices with limited hardware are more susceptible to attacks, specially to: *Denial of Service* (*DoS*),

[1] https://ieeexplore.ieee.org
[2] https://dl.acm.org/
[3] https://www.sciencedirect.com/
[4] The presented literature review was performed on December 13, 2018. Dashed lines in Figure 2.7 represents the update performed on January 10, 2020.

Modification Attack, *Dropping Attack*, and *Appending Attack*. Results obtained in their work presented reduction on the traffic generated through Smart Home's network and decreases the vulnerability of the devices. Although simulations point to a reduction on devices' processing overhead and on the number of packets on the network, it did not discuss how the devices are authenticated nor how limited power devices could be used in the environment.

Although there are important initiatives, there are security issues (*e.g.*, Sybil [89], DDoS [338] and Eclipse [137] attacks) that were not properly addressed. For example, in an Eclipse attack, a malicious node can control the information that is shared with another node (*e.g.*, a gateway). Consequently, the information sent by this node can be omitted from the blockchain in the other nodes. In a hierarchical P2P architecture this kind of attack could be worse, since a supernode controls data from multiple devices. There is a lack of discussion in the literature about Eclipse attack and its impact in blockchains for IoT. Furthermore, there is almost no discussion on how a blockchain can be affected by insecure APIs that access blockchain data.

Also, smart contracts could be affected by problems related to blockchains. For example, in comparison to traditional databases, the solution could present lower throughput [65]. This latency is caused by the mining process in some blockchains and could act as limiting factor, thus its application to real time solutions should be carefully evaluated before being used [145]. A deployed contract is permanent, in the Ethereum case, or have a great management cost to change, as in Hyperledger. Thus, the contract logic needs special attention to avoid flaws, which can be used to exploit vulnerabilities and expose a variety of risks to the network and users [65][108][252][81][347]. There are research initiatives [225] to help developers to avoid issues during the smart contract implementation. Despite of that, security issues in smart contracts can be further explored and discussed, as well the smart contract applications in IoT, such as: firmware update, M2M payment and transactions, and tracking devices.

In order to better categorize theses papers, we proposed a layer-based categorization (published in [371]), composed by four layers: communication, consensus, data, and application (as presented in Figure 2.8).

An overview of these works from 2015 to 2018 are presented in Table 2.2. It is important to note that they are categorized - based on the main contributions - into one of the presented layers or in “others” when the contribution cannot be categorized (*e.g.*, survey paper or position papers). Furthermore, it is important to note that the majority of the papers are categorized as “others” or “application” (layer). Thus, many researchers are discussing how to use blockchains in different contexts. However, it can be observed that the category consensus is the field with less published papers, hence there are new possible opportunities of research in this field.

This classification motivated our work to explore consensus algorithms for blockchains applied to IoT due to the lack of attention of the community from 2015 to 2018.

Table 2.2: Blockchain in IoT categorized using the layer-based model

	Communication	**Consensus**	**Data**	**Application**	**Others**
ACM	[206], [54], [33], [92], [7]	[167], [317] [101]	[132], [10], [52], [152], [232]	[358], [254], [235], [189], [9], [302] [83], [324], [229], [136], [168], [62], [154]	[310], [290], [287], [148], [227], [48], [342], [106], [13], [316], [207], [169], [113], [257], [267], [91], [96], [74], [153], [84], [237], [58]
IEEE	[289], [142], [209], [87], [327], [359] [308], [332], [357], [288], [43], [55], [305], [256], [72], [336], [133], [333], [131], [22], [220], [12], [278], [161], [163], [309], [293], [307], [258], [312], [365]	[325], [103], [128], [195]	[160], [277], [218], [259], [85], [205], [343], [201], [236], [369], [216], [345], [6], [42], [11], [77], [334], [164], [211]	[366], [269], [65], [289], [4], [198], [40], [75], [263], [82], [291],[145], [20], [355], [178], [351], [295], [177], [162], [64], [1],[95], [102], [353], [286], [125], [197], [118], [192],[196], [241],[250], [276], [314], [28], [51], [114], [193], [249], [204], [2], [16], [23], [114], [247], [90], [348], [140], [300],[63], [315], [130], [219], [340], [246], [120], [191], [304], [274], [279], [352], [156], [275], [117], [319], [165], [172], [150],	[68], [35], [134], [200], [361], [337] [330],[368], [370], [320][127],[266] [185], [180], [99], [301], [69], [56], [190], [297], [100], [15], [281], [177], [139], [212], [176], [285], [105], [335], [116], [230], [238], [344], [283], [202], [107], [18], [233], [179], [318], [350], [328], [296], [326], [313],[46], [97], [184], [362], [151], [264], [146], [228], [363], [373], [174], [3], [199], [144], [364], [135], [323], [372], [255], [237], [60], [122], [265], [294], [121], [356], [292]
Science Direct	[47], [59], [126], [306], [360], [203]		[86]	[147], [331], [268], [367], [24]	[143], [181], [173], [271], [166], [104], [234], [284], [183], [329],[29], [170], [14], [182], [311], [222], [280]

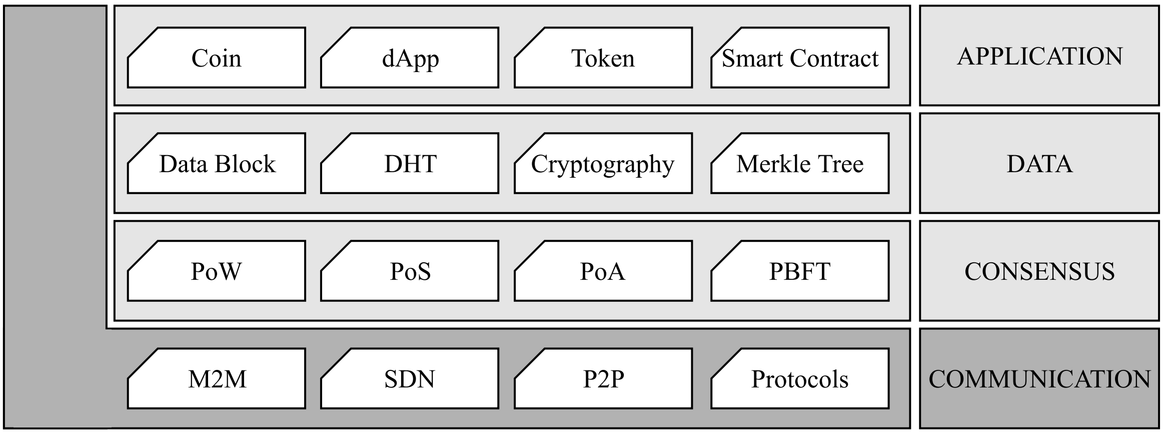


Figure 2.8: Layer-based model for blockchain categorization (extracted from [371]).

After this search in the literature was performed, other papers were published. Consequently, in the next section we present a discussion about the main works about consensus algorithm for blockchains in IoT and how the proposed work can help appendable-block blockchain application in different IoT environments.

Hence, although different blockchains for IoT were proposed, there are some issues that remain open. For example, there are few discussion about consensus algorithm and its performance in IoT scenarios. Also, there is few discussion about the security of the new proposed consensus algorithms, specially to IoT environments. And, finally, there are no discussion about how to integrate different blockchains and how to use the users behavior (including in different blockchains) in the consensus mechanisms in order to have a multi-level consensus algorithm.

2.7 Chapter Summary

In this chapter we presented the main concepts about blockchain that will be used or discussed in the next chapters. We presented the main architectures, different data structures, and how consensus algorithms are used. Also, we discussed the growing interest on the use of blockchain in IoT environments. In the next chapter, we will discuss relevant works on consensus algorithms for blockchains in IoT.

3. RELATED WORK

The adoption of blockchain technology could be a challenge for IoT environments [69]. The main problem regarding the use of blockchain in IoT is related to the limited hardware capabilities of the devices that run on the IoT context. This limitation requires lightweight solutions, and most of the public blockchain size makes them inapplicable for IoT. Another problem regarding hardware limitation is related to computing power of IoT devices. For example, as mentioned before, Bitcoin [242] applies the Proof-o-Work (PoW) consensus algorithm, which uses hash brute force calculation and, therefore, demands a lot of time, processing power and energy to achieve consensus.

Consequently, several researchers proposed solutions to address the two most common problems of blockchains: performance - response time to add both new blocks and new transactions [216][119][369] and scalability issues - capability of all IoT devices to interact with a blockchain - [69][250][374]. Although the new proposed solutions present innovative ideas, they are in development and do not present an appropriate evaluation in real scenarios. Consequently, this chapter focuses mainly in two aspects: **consensus algorithms** and **performance and security issues** for blockchains in IoT.

The consensus algorithm plays a crucial role for ensuring that each new block contains valid information, and any peer is able to verify the information in the blockchain. The unreliable peer environment, where a blockchain is being executed, could be considered in order to provide a solution to a common authentication problem, which is related to have a third party involved. In an authentication context, the third party is responsible to assure the trust in each party involved in the authentication process.

3.1 Searching Engine and Process

We used the similar search string presented previously on Section 2.6. The search string *"consensus" AND "blockchain" AND ("iot" OR "internet of things" or "internet-of-things")* on IEEEXplore[1] and its equivalent search string on ACM Library[2] and Science Direct[3] databases were used to help this study. After that, we read the abstract of the papers and selected the papers that had some proposal or contribution on consensus for blockchains in IoT. We did not select any of the papers proposed by the author of this work. The results of our findings are presented in the next section.

[1] https://ieeexplore.ieee.org

[2] https://dl.acm.org/

[3] https://www.sciencedirect.com/

3.2 Results & Discussion

An important discussion about consensus algorithms is presented in Christidis *et al.* [65] research. This research investigated different consensus algorithms such as Sieve, Practical Byzantine Fault Tolerant (PBFT), Proof-of-Stake and Proof-of-Work. As mentioned previously, Christidis *et al.* consider that both the network in which it will be used and the attack vector that is intended to be mitigated are the most important factors to decide which blockchain design should be adopted. Consequently, the number of nodes and the processing overhead are important issues to be considered. Christidis *et al.* research presented blockchain as a solution to provide security in IoT. Despite the application examples evaluated in their research, they did not discuss how to improve blockchain technology, in particular performance, to be used in different IoT environments.

PBFT can have some interesting characteristics for IoT environments: it does not require devices with high processing power and it does not require coins or tokens. In order to evaluate the performance of PBFT in IoT scenario, Sukhwani *et al.* [325] modeled PBFT using Stochastic Reward Nets. In their study (performed with up to 100 nodes), initial results were presented showing that PBFT can be a problem in a large scenario and is affected both by the number of nodes and by the number of transactions.

Han *et al.* [128] presented an evaluation of different Byzantine problem based consensus algorithms used in blockchain for IoT scenarios. They evaluated PBFT (in Hyperledger Fabric v0.6 blockchain), Ripple, and Byzantine Fault-Tolerant State Machine Replication (BFT-SMaRt) consensus algorithms using the same workload. The results showed that for a high number of requests and high number of nodes (simulating an IoT scenario) the throughput is drastically affected (going to zero successful with more than 3000 requests per second).

Feng *et al.* [103] proposed an Hierarchical Byzantine Fault Tolerant consensus algorithm in order to solve the scale issues presented by PBFT. The idea consists of clustering nodes and setting a leader for each cluster. Only these leaders will perform the consensus. Although it is an interesting approach, it can be also expressed by Gateways of networks performing a BFT-based consensus algorithm (that was also proposed by other works [85][216]).

Lao *et al.* [188] proposed a consensus algorithm called Geographic PBFT (G-PBFT). This algorithm uses geographic information of fixed IoT devices to perform consensus. As a consequence, only devices with fixed location can perform the consensus. This approach can help to avoid Sybil attack from devices geographically distant and use data from “trusted" nodes to perform the voting procedure. The approach can reduce the overhead in the message exchange procedure of PBFT and can be useful in some IoT scenarios. However, this approach is limited to IoT scenarios composed by fixed devices with a GPS or other global localization solution.

Hao *et al.* [130] proposed a blockchain for multi-agent based E-commerce. In their work, they proposed the adoption of an adapted Raft consensus algorithm. The results obtained were promising (few seconds to perform the consensus). However, Raft has some security issues (due to the centralization of the consensus) and the time required to perform the consensus can be high (few seconds that can elevated for IoT applications that usually requires fast response time).

Khan [166] proposed a novel consensus algorithm called FAST. FAST is based on MapReduce function to aggregate transactions in a block, in order to produce a faster rate of transactions per second. Three roles are used in this approach: user (also called as light client), worker, and master. The consensus is performed by a master node (elected among worker nodes). However, the trust relies on the master node. Hence, if the master node is a malicious node or it is attacked, new insertions in the blockchain are compromised.

Solat [317] proposed a consensus algorithm for blockchain-based IoT scenarios called RDV. This consensus algorithm is a voting based algorithm composed by three main steps: register, deposit, and vote. Every node that want to participate in the consensus have to register, and the list of all registered devices is stored in the blockchain. After registering, every node has to confirm the registration by “depositing” some coins. And, finally, every registered node can vote. Consequently, malicious nodes are discouraged to participate in the consensus procedure due to the cost to participate in the voting process. However, the proposed scenario was not evaluated or compared to other consensus mechanisms. For example, there is no discussion about how it is compared to another voting based consensus, such as PBFT, dBFT, or FBA.

Fan *et al.* [101] proposed Roll-DPoS, an adapted version of Distributed Proof-of-Stake (DPoS) designed for IoT environments. In their algorithm, for each block generation, a node is responsible to produce the block, and send it to be validate by the other nodes that are participating in consensus. After the block is validated (or not), the chosen node can be reelected to continue producing blocks, or it can be changed if it is not being fair. Although the algorithm is presented, few discussion is made about performance, security issues or its applicability in dynamic scenarios.

Huang *et al.* [141] proposed a credit-based proof-of-work consensus for IoT. It is based on decreasing the mining difficulty for honest nodes and increasing for dishonest nodes through the use of credits generated when new blocks are created. Also, the data structure is modified to use a directed-acyclic graph (DAG) instead of a chain. To assert performance, an evaluation is performed in a smart factory scenario. The results show a better performance than traditional PoW without compromising security.

Another work based on PoW is the Proof-of-Authentication (PoAh) consensus algorithm [221, 270]. PoAH uses media access control as addresses in the blockchain network for each node to reach consensus. The nodes are selected dynamically to verify transactions based on the address. In the work of Maitra *et al.* [221], performance evaluation shows energy consumption and latency. However, a comparison with other consensus algorithms is necessary to assert the algorithm performance.

Qiu *et al.* [272] presented a new blockchain framework to be used in IoT environments using a consensus called Dual Vote Confirmation based Consensus (DVCC). They use an hierarchical architecture with different roles. Some nodes participate in the consensus procedure. The consensus is based on a mining approach (similar to PoW), in which a miner and validators (called verifiers by the authors) are randomly selected. After a block is proposed, both verifiers and other nodes should vote. Authors did not compare the proposed solution with PBFT or any other voting based consensus. As a consequence, it is not clear the advantages over existing consensus algorithms.

Biswas *et al.* [36] proposed the Proof of Block and Trade (PoBT) consensus algorithm. This algorithm validates transactions (trades) and blocks while still maintain a lightweight algorithm suitable for IoT. One of the approaches to attain its lightweight is limiting the number of peers participating in a session to reduce the latency and increase throughput. This number depends on the total number of nodes in that session. Also, the ledger is split and distributed between nodes, which reduces the memory needs for IoT devices. The consensus algorithm was implemented in the Hyperledger Fabric and showed improvement regarding performance when compared to traditional Hyperledger Fabric operation.

Chai *et al.* [57] proposed a reputation based consensus algorithms to be used in blockchains applied to Industrial Internet-of-Things (IIoT). Their proposal aims to reduce the amount of processing required to insert new blocks by the usage of a trust system. In their proposal, only nodes with high reputation can propose new blocks. This reputation is based on tasking solving (similar to a PoW). Nodes that inspect new blocks also receive a reward (based on reputation). As a consequence, more blocks inserted in the blockchain means higher reputation. However, discussion on how this can be implemented on a IoT scenario is limited. Similarly, but with a very preliminary evaluation, Makhdoom *et al.* [223] proposed a system, called PLEDGE, that uses a reputation system similar to what was proposed by Chai *et al.* [57].

Another reputation based consensus was proposed by Liu *et al.* [210]. In their system they propose an anonymous reputation system that aims to preserve nodes identities. To do that they propose a mechanism based on a Proof-of-Stake consensus protocol using a blockchain-based reputation system. They use a zero-knowledge-proof mechanism to keep secret the identity of the users that propose the new transaction and the users that proceed with the validation of the transaction. This proposal has some interesting features to be used on privacy of the users. However, it cannot be adapted for different contexts and different IoT scenarios.

One more work that focus on reputation mechanism was proposed by Asheralieva and Niyato [19]. They use a consensus mechanism based on shards, in which each peer votes if the tasks outputs are correct. The votes are weighted based on the reputation of each node. As a consequence nodes that have a higher reputation have higher weight in the consensus voting procedure. As a result, their approach can achieve good performance and reduce the risk of bad reputation/malicious user to introduce incorrect information. However, no discussion is presented to the latency of transactions insertion, and their solution presents a low throughput (few transactions per minute).

Raghav *et al.*[273] proposed a consensus mechanism called PoEWAL. This consensus is based on a probabilistic proof of elapsed work and luck (PoEWAL). The first part is similar to the traditional hash based PoW but with a limited time window. The hash with more initial bit zeros is chosen and the node that produced it is rewarded. When a conflict occurs (same number of zeros in the hash), a luck mechanism chooses the hash and its miner is rewarded. This kind of approach can be used on non-cooperative IoT environments (public blockchains). The goal of this method is to reduce energy consumption of traditional PoW and to reduce the latency of transactions insertions. However, it can have many security issues both in the election of the winner hash and in the luck mechanism. Also, due to processing/energy consumption required in PoW it can be hard to be applied in many IoT environments.

Bai *et al.* [26] proposed a consensus algorithm for a two-layer IoT architecture. In their scheme, data are shared in what they called “base layer" where the basic information and transactions are maintained. In a top layer audition information and data are stored in another kind of block. They use a reputation based scheme to achieve consensus for the top layer and multiple proofs scheme for the base layer. This work presents consensus for more than one layer, but different from the multi-level consensus proposed on this thesis, it is closer to other Quorum approaches *e.g.*, FBA consensus algorithm [226]. Also, they cannot be adapted for different contexts.

Table 3.1 presents an overview of the presented research about consensus algorithms for blockchain in IoT. As can be observed, some proposals did not evaluated the consensus algorithm, and the ones that have some results did not had evaluation in real IoT hardware. Also, it can be observed that the proposals are mostly designed for private/permissioned scenarios. Finally, there are no discussion about adaptive solution to different contexts. It is important to note that authors did not discuss different contexts due to the adopted blockchain architecture or to be out of the scope of their research.

The main challenges for consensus algorithms to be used in blockchains for IoT are related to: security, trust, overhead (or performance) and scalability [50]. However some applications can have different demands. For example, some applications can demand more on the scalability than performance, *e.g.*, vehicles tracking based on GPS has a small rate of updating but has a large number of users; while others can be the opposite, *e.g.*, a limited

Table 3.1: Overview of consensus for blockchain in IoT

	Consensus	**Evaluation**	**Access and Control**	**Main Issues**	**Contexts**
Sukhwani *et al.*	PBFT	Emulated scenario with 100 nodes using Hyperledger Fabric	Private and Permissioned	Lacks discussion about node hardware constraints; Larger scale can be problematic.	N/A
Han *et al.*	PBFT, Ripple and BFT-SMaRt	Emulated scenario with Hyperledger Fabric	Private and Permissioned	Larger scenarios had problems to validate transactions.	N/A
Feng *et al.*	Hierarchical Byzantine Fault Tolerant	Simulated with 100 nodes	Private and Permissioned	Values can lead to problems in larger scale.	N/A
Lao *et al.*	G-PBFT	Simulated up to 202 nodes	Private and Permissioned	Limited to localization information and fixed devices.	N/A
Hao *et al.*	RAFT	Simulated with 4 nodes	N/A	Results are not conclusive. Only 4 nodes were used.	N/A
Khan	FAST	Simulated with 4 to 16 nodes	N/A	Consensus takes more than 10s.	N/A
Solat	RDV	N/A	Public and Permissionless	Reward based consensus. Evaluation was not performed.	N/A
Fan *et al.*	Roll-DPoS	N/A	N/A	A single node can be an target and compromise the insertion. Evaluation was not performed.	N/A
Huang *et al.*	PoW based	Single local device	N/A	PoW approaches can be problematic for limited IoT devices.	N/A
Maitra *et al.*	PoAh	3 local devices	N/A	Evaluation shows the power consumption but without comparison with other approaches	N/A
Qiu *et al.*	DVCC	Simulation with 32 verifiers	Private and Permissioned	Different procedures, random selection, voting, mining.	N/A
Biswas *et al.*	PoBT	Simulation with 100 nodes	N/A	Scalability issues.	N/A
Chai *et al.*	Proof-of-Reputation	Simulation, number of nodes not available	N/A	Lack of discussion about implementation on IoT nodes.	N/A
Liu *et al.*	Proof-of-Reputation and PoS	Simulation, number of nodes not available	N/A	Lack of discussion about implementation on IoT nodes.	N/A
Asheralieva and Niyato	Proof-of-Reputation and voting	Simulation with 20 nodes	N/A	Lack of discussion about implementation on IoT nodes.	N/A
Raghav *et al.*	PoEWAL	Simulation with 50 nodes	Public and Permissionless	Limited to powerful IoT devices	N/A
Bai *et al.*	two-layer	Simulation, number of nodes not available	Private and Permissioned	Lack of discussion about implementation on IoT nodes.	N/A

number of smoke sensors in a smart building requires a lower latency as possible. This two different contexts motivated our work to propose an adaptive solution that will be discussed in the following chapters.

3.3 Chapter Summary

In this chapter we presented different efforts to propose or adapt new consensus mechanism to be used in blockchains applied to IoT environments. We could observe interesting approaches that can be used, however none presented a proposal to be adapted to different contexts. The next chapter presents our blockchain called appendable-block blockchain, where this thesis proposal is inserted and where the PhD candidate contributed for its development.

4. APPENDABLE-BLOCK BLOCKCHAIN

In this chapter, we present the fundamental concepts of blockchain architecture that underpins our proposed framework. Appendable-block blockchains was proposed and designed by different researchers from CONSEG research group (in which the author of this thesis participates) during the last four years. Consequently, many discussions presented in this chapter were presented in the previous works of the group [216][232][215][251][76][214]. In the first stages of the appendable-block blockchain framework (formerly called R2AC and later on called SpeedyChain), Lunardi *et al.* [216] presented a lightweight permissioned blockchain that creates blocks on demand focused on IoT for Smart Homes/Smart Offices scenarios, using a layer-based architecture [157]. As can be observed in Figure 4.1, devices and gateways are separated in different layers (perception and transportation), thus they have different roles in the blockchain.

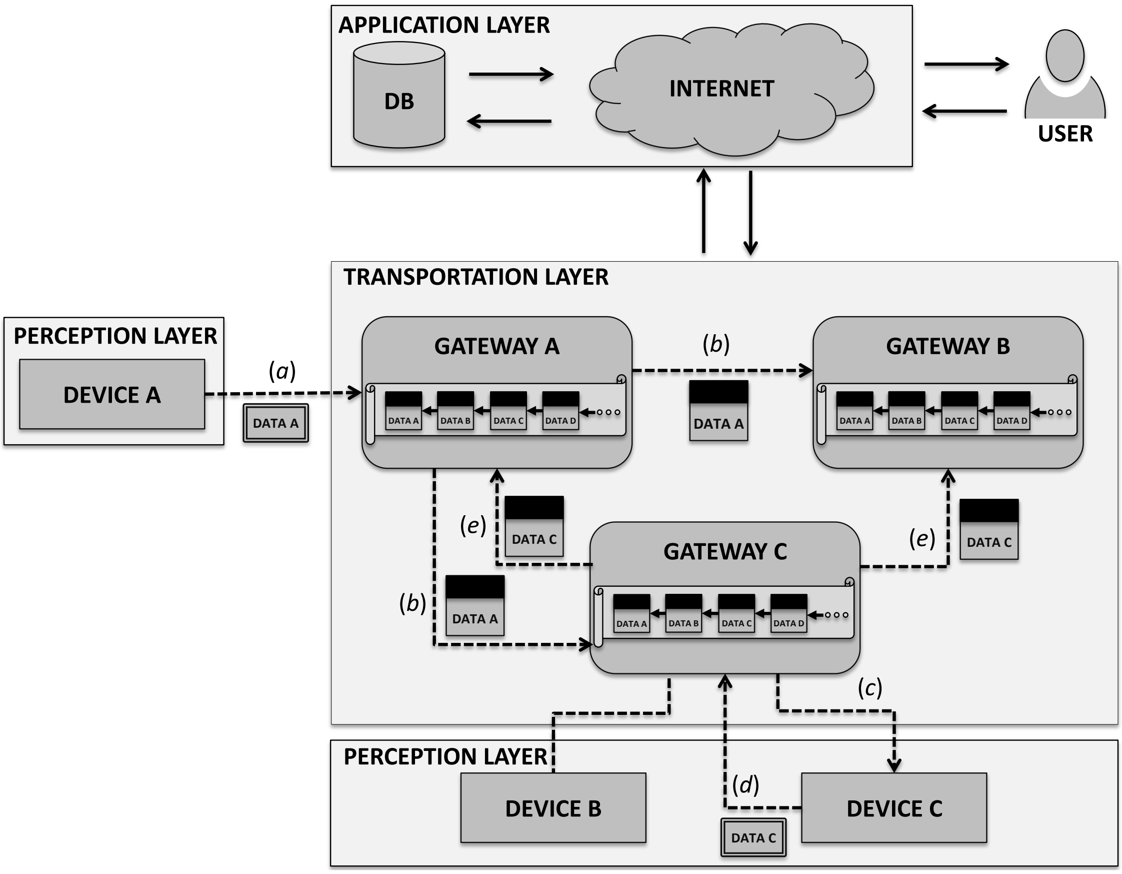


Figure 4.1: Gateway-based Architecture for IoT (extracted from [216])

Therefore, each device can produce information and send to the gateways to append data to its own block. Consequently, devices can keep producing and appending information into blockchain independently to the other devices operations. Gateways will maintain the blockchain, that is composed mainly by two parts: block ledger and block header (as shown in Figure 4.2). Additionally, gateways are able to maintain only the Block Header

- which contains important information about devices (especially their public keys) - without having every device's block ledger. The block ledger is composed by the information digitally signed (both by the device and gateway), and chained through the hash of the previous information (or to the block header if it is the first information of the block ledger).

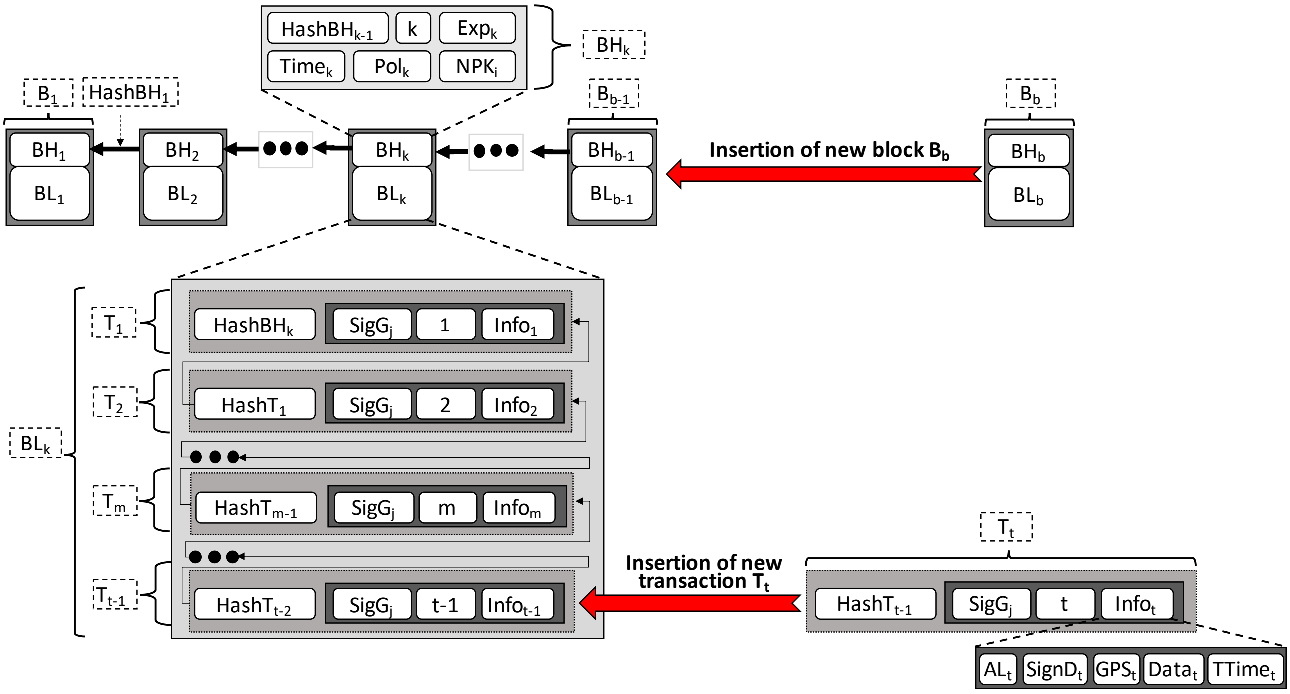


Figure 4.2: Appendable-block blockchain components (adapted from [215]).

After receiving feedback from different researchers and analyzing a broader scenarios for smart cities, the initial blockchain was modified to fit a smart city scenario [232]. In this improved version called SpeedyChain, the main changes were performed in the block header (inclusion of an expiration field to avoid a block that produce a too long chained information in the block ledger) and in the block ledger (different fields were added to support different data such as access level). After that, some other changes were performed mainly in operations allowed in the blockchain and in the consensus algorithms. In order to properly discuss the current version of the blockchain, the next section presents the current architecture, the blockchain definition, and main operations available in the SpeedyChain. Furthermore, preliminary results obtained in the evaluation, publications and other research results, and open issues in the SpeediChain are presented.

It is important to note that the next sections present the formalization of the appendable-blockchain (former called SpeedyChain framework), its main operations and evaluation. Part of the text of the next sections and chapters were extracted or adapted from different published [216, 232, 371, 215, 214, 76, 80] works from the CONSEG research group (in which the PhD candidate is on of the authors). Publications and other contributions were discussed in Section 1.6.

4.1 Architecture

Let $N = \{N_1, ..., N_n\}$ be the set of n nodes in the system with public-private key pairs (NPK_i, NSK_i). Also, consider that these nodes can have different roles in the architecture. Consequently, this system is composed by d devices, where $D = \{D_1, ..., D_d\}$, that usually produce information and could be controlled remotely; g gateways, where $G = \{G_1, ..., G_g\}$, that manage the access to information in a blockchain; not limited to this, different kind of nodes are supported such as s service providers $SP = \{SP_1, ..., SP_s\}$. Therefore, $N_i = \{D, G, SP\}$. Assume that all nodes in N can use the same cryptography algorithms. Moreover, every NPK_i should be different and accessible by any participant in this system. Also, assume that a key pair (public and secret keys) from a device will be represented as (DPK_j, DSK_j) and a key pair from gateway will be represented as (GPK_h, GSK_h). Consider that each device in D (Perception Layer) should be connected to a gateway in G (Transportation Level) through different (wired or wireless) network devices. Additionally, the gateways are responsible to manage the device access and provide an API that allows to manage the blockchain.

4.2 Data Model

Based on the IoT architecture presented in Figure 4.1, the blockchain will be maintained by gateways in G (Gateway Level in Figure 4.1). To ensure that every participant can access any NPK_i (*e.g.*, DPK_j or GPK_h) and information stored in a Gateway was not tampered with, let a blockchain $B = \{B_1, ..., B_b\}$ be a set of b blocks. Each B_k has a pair of different information (BH_k, BL_k), where BH_k is responsible to maintain the block header of B_k and the BL_k stores the block ledger, *i.e.*, the set of transactions of B_k as shown in details in Figure 4.2.

Therefore, BH_k is composed by ($HashBH_{k-1}, k, Time_k, Exp_k, Pol_k, NPK_i$), where

$$HashBH_{\text{k-1}} = \begin{cases} 0 & \text{, when } k = 1 \\ \text{hash digest of } BH_{k-1} & \text{, when } k \geq 2 \end{cases}$$

where hash digest is obtained through a hash function, *i.e.*, $HashBH_{k-1}$ contains the hash digest of previous block header (or zero when it is the first block); k is equal to the index of the block B_k in the blockchain; $Time_k$ is the timestamp from when the block was generated; Exp_k presents the threshold time to insert a new transaction in its block ledger, for example, after this time a device should create a new key pair (NPK, NSK) and create a new block; Pol_k presents the access policy that the device has to attend; and NPK_j is the node public key. It is important to mention that every node - independent of its type - should have a block in B, composed of at least a block header, and every *NPK* should be available in the blockchain.

Let BL_k={T_1, ..., T_t} be the set of *t* transactions on the block ledger of the block B_k. T_m is composed by ($HashT_{m-1}$, *m*, $SigG_m$, $Info_m$), where

$$HashT_{\text{m-1}} = \begin{cases} \text{hash digest of } BH_{\text{k}} & \text{, when } m = 1 \\ \text{hash digest of } T_{m-1} & \text{, when } m \geq 2 \end{cases}$$

where the $HashT_{m-1}$ contains the hash of the previous transaction (or the hash of its block header when it is the first transaction of the block ledger); *m* is equal to the index of the transaction T_{m} in the block ledger BL_k; $SigG_m$ represents the result of the cryptography using the GPK_h to sign $Info_m$.

The $Info_m$ can be different for each type of node. Devices provide a set of information ($SigD_m$, AL_m,GPS_m, $Data_m$, $TTime_m$), where AL_m is the access level required to access the information from outside of the blockchain that is defined by the device D_j, while the $SigD_m$ represents the signature of (AL_m, GPS_m, $Data_m$, and $TTime_m$) using DPK_j, where GPS_m represents the global position of the device (when it is available), while Data_{m} is the data collected/set from/to device D_j and $TTime_m$ is the timestamp when the $Data_m$ was generated/set. It is important to note that $Data_m$ could be formatted differently depending on the device. For example, it could store a single read of a sensor (an integer type) or a set of information, encrypted or not, depending on the configuration established in the API level. Before any device performs its first transaction, it should authenticate through a gateway. For example, in Figure 4.1, Device A is authenticated in the blockchain through Gateway A. After that, the device has to perform a Key Exchange procedure with the gateway to build a secure channel. This procedure is presented as follows:

1. Device A (represented in blockchain as D_a) sends a Hello message with its own Public Key DPK_a (*e.g.*, for encryption using the RSA algorithm) to Gateway A (represented in blockchain as G_a);

2. Gateway A perform the key exchange (*e.g.*, to build an Advanced Encryption Scheme (AES) secure channel) using the received DPK_a;

3. Device A sends a first transaction through an encrypted channel using the AES key generated by the gateway;

4. Gateway A starts the consensus with the other gateways to insert a new block B_a with DPK_a in the block header BH_a and the first transaction T_1 in the block ledger BL_a;

5. After the consensus, if the block is considered valid, the block B_a is inserted in the blockchain;

While the device private key should be kept secret, the public key (represented as DPK_a in Figure 2.5) will be publicized and used by the gateway to identify the device. After the B_a is in the blockchain, the device can produce transactions that will be appended in the block ledger BL_a until the Exp_a is lower than the $TTime_m$ in the transactions T_m. Anytime that a gateway receives a transaction with its timestamp $TTime_m$ with a higher value than the expiration time Exp_a the gateway will proceed with key update algorithm. Also, the node D_a can send to the gateway a request to update its public key DPK_a' previously the expiration time is reached. In both situations, a gateway will request to the device N_a its new public key DPK_a'. After the key validation (*e.g.*, if the key is not already in the blockchain), the gateway will append a new block into the blockchain with the new DPK_a' from device D_a.

4.3 Consensus

SpeedyChain was improved (as part of this thesis proposal) to support different consensus algorithms. Before discussing different consensus algorithms, first we need to present what is a valid block or transaction. For a transaction to be considered valid, it should have a node (device, gateway, service provider, etc) NPK_i that is already in the blockchain, a valid signature (based on the data transmitted and NPK_i), and a $TTime_m$ lower than its Exp_k (present in the block header) to ensure that no transactions are inserted in an expired block. Moreover, to ensure that a block header is valid: (*i*) the gateways should agree that a new node NPK_i can be part of the blockchain B; (*ii*) the access policy Pol_k for this node NPK_i should be defined; (*iii*) the Exp_k should be calculated to avoid a large block in size. In this work we assume that this validation is performed by the gateways through predefined rules.

Four different consensus algorithms were incorporated to appendable-block blockchains: (*i*) validation based on the authority of gateways and using a specific number of witness, where every block should be signed by at least a predefined number of witness (2 witness were adopted in this work); (*ii*) adapted PBFT algorithm, where more than 2/3 of the active gateways should validate and sign the block; (*iii*) adapted dBFT algorithm, where more than 2/3 of delegated gateways should validate and sign the block; (*iv*) a simplified PoW algorithm, used for comparison since it is adopted in many different blockchains, where a gateway achieves a hash with a certain characteristic (in this work, first 12 bits should be equal to zero). All consensus algorithms, except PoW, could be summarised in Algorithm 4.1.

In order to encapsulate the new block B_k, every information from the block header BH_k is set, such as the hash of the previous block header BH_b (line 2), block index I_k (line 3), the timestamp using the time of block creation $Time_k$ (line 4), an expiration time Exp_k to control the validity of the block (line 5), and the access policy Pol_k that the new node is submitted to. It is important to note that both Exp_k and Pol_k are defined in API level. After the block header is created, the consensus is performed (line 7). It is important to note that the consensus is performed only by gateway nodes. After the consensus is performed

Algorithm 4.1 Generic consensus algorithm

```
Require: receive a $NPK_i$ to perform consensus
 1: $b \leftarrow$ lastIndex($B$)
 2: $HashBH_{k-1} \leftarrow$ hash($BH_b$)
 3: $k \leftarrow b + 1$
 4: $Time_k \leftarrow$ getTime()
 5: $Exp_k \leftarrow$ defineExp()
 6: $Pol_k \leftarrow$ setPolicy()
 7: $BH_k \leftarrow \{HashBH_{k-1}, k, Time_k, Exp_k, Pol_k, NPK_i\}$
 8: $consensusResponses \leftarrow$ performConsensus($BH_k$)
 9: if $consensusResponses > minimumResponses$ then
10:     broadcast($BH_k$)
11: end if
```

and it receives more than the minimum responses for each consensus algorithm, the new block is broadcast to the peers (line 10).

4.4 Smart Contracts

The appendable-block blockchain supports the use of smart contracts. This feature uses a unique model proposed in the work of Nunes *et al.* [251], called Context-based model for smart contracts. This model allows the execution of groups of smart contracts in parallel to process a high number of transactions while still maintaining low latency. These two qualities are important in IoT Domain. Also, the smart contract feature can help in the management and maintenance of IoT Devices as discussed by Christides [65].

However, despite the benefits of this model, there are limitations to its use that may negate its benefits. The most important is that Smart Contracts exist in a context, which is a structure that isolates this group of smart contracts from others. Therefore, a smart contract in a context can not interact with another smart contract in a different context. Thus, a group of smart contracts in one context will have sequential processing and the parallelism feature is attained by processing different contexts in parallel. Therefore, it is important to properly select a program that will execute on top of this model. Programs that can be split into smaller not interacting parts are desirable because these parts can be inserted in different contexts to attain parallelism.

Due to initial development of smart contracts on appendable-block blockchains and some limitation that it presents, we did not evaluated smart contracts transactions in this thesis.

4.5 Main open issues

Appendable-block blockchains use an hierarchical architecture and a bespoke data structure (with separated insertion of blocks and transactions), which allows to insert transactions in parallel across nodes. In appendable-block blockchains, consensus is only performed when creating and inserting a new block for a node [215]. Once the block is created for a specific node, the node can attach transactions to the block. As a consequence, where is no consensus to insert such transactions in appendable-block blockchains. That is, nodes have to trust its gateway - a full node that controls the access to other nodes to the blockchain - to insert valid transactions in their blocks. In addition, appendable-block blockchains assume that devices connect to only one gateway at a time.

Therefore, without a consensus at transaction level, appendable-block blockchains are susceptible to misuse and attacks through malicious or tampered gateways. Such gateways can compromise the insertion of information (e.g., insert an invalid execution of a smart contract) and can eclipse devices or hide devices information (not inserting that into the blockchain).

4.6 Chapter Summary

In this chapter we presented the main concepts of appendable-block blockchain, as well an adaption to support four different consensus algorithm to insert new blocks. However, not performing consensus to insert transactions can be problematic to appendable-block blockchains. In particiular, due to the issue that malicious node can hide information from devices or insert wrong smart contracts execution in the blockchain. In Chapter 5 we propose a solution to solve this issue and to help in the performance and the adaptability of the blockchain to different contexts.

5. MULTI-LEVEL CONSENSUS ALGORITHM

Appendable-block blockchain is a solution to allow the insertion of new transactions into already inserted blocks. In the preliminary experiments with appendable-block blockchains [232][215], consensus is only performed when creating and inserting new blocks, and there is no consensus to insert transactions into blocks [215]. However, some issues were not properly addressed by theses works. Firstly, that proposals of appendable-block blockchains have a communication protocol that allows a device to connect to only a single gateway. As a consequence, transactions produced by devices can be omitted by a malicious gateway. Secondly, the consensus is performed only to insert new blocks, which means that invalid transactions can be included. Finally, scalability can be a problem as the usage of consensus in the current appendable-block blockchain would be performed individually for each transaction, leading to latency issues as we will present in the evaluation. The contributions of this chapter were presented in part in a previous work [214].

5.1 Context-based transaction consensus

We propose a context-based consensus algorithm to address the limitations of the current version of SpeedyChain. In particular, to solve the first issue, we propose that every device should connect to a minimum initial set of gateways. When a device connects to multiple gateways, this eliminates a malicious gateway from cheating as other gateways will detect it, differently to what would happen if a device was connected to a single gateway. A simplified version of the connection protocol is presented in Fig. 5.1. The main steps are described as follows:

1. Device a (represented *Dev a* in Fig. 5.1) sends a Hello message with its own public key *Dev a Pubkey*, *e.g.*, for encryption using the asymmetric cryptography, to gateway A (*Gw a*);

2. *Gw a* verifies if *Dev a PubKey* is in a block header of the blockchain, *i.e.*, a block for that device was inserted previously in the blockchain:

   (a) In the case that the *Dev a PubKey* is not in a block header and the device is allowed to access the network, *Gw a* starts a consensus to include a new block for *Dev a* containing its *PubKey*;

      i. Other gateways (*Gw b* and *Gw c* in Fig. 5.1) verify the proposed new block, they vote (signed voting) and send the result back to the gateway that started the consensus;

3. After the consensus (if the block is considered valid), the block containing *Dev a PubKey* is inserted in the blockchain. Then, if *Dev a PubKey* is in the blockchain, *Gw a* and *Dev a* can establish an encrypted channel using symmetric cryptography;

4. After a device connects to a gateway, they can exchange information. Our proposal allows a device to send the same transaction to multiple gateways. These multiple connections with gateways can help to avoid a device from being eclipsed by a tampered gateway. Allowing the connection to multiple gateways is an improvement to appendable-block blockchain, as discussed previously. Also, it is important to note that devices' transactions have a timestamp and a digital signature.

5. Any update from a device is a transaction in the blockchain.

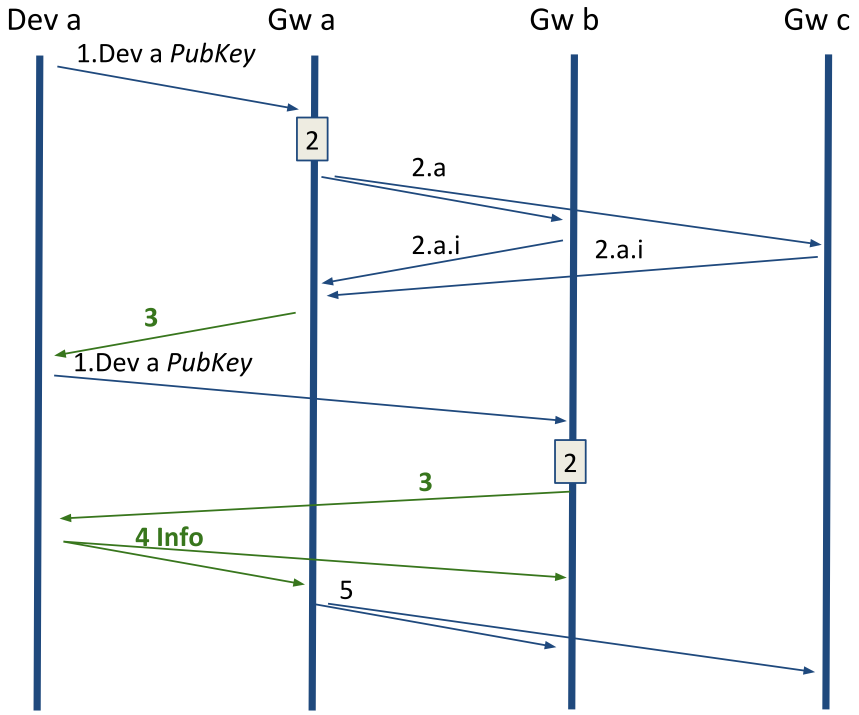


Figure 5.1: Device connection simplified protocol.

As mentioned previously, every device can connect and send its update (a new transaction) to multiple gateways at the same time. For instance, *Dev a* is connected to three gateways (Gw a, Gw b and Gw c), as depicted in Figure 5.1.

As presented previously, appendable-block blockchains allow nodes to insert transactions into their own blocks at the same time, independently from each other. However, there are two remaining issues. One of them is that the current version of SpeedyChain does not present consensus at transaction level, *i.e.*, a gateway - to which a device is connected - inserts the transaction in the device's block ledger and sends it to the other gateways. In this case, different gateways can insert the same transaction in the blockchain (duplicating the same transaction), or in the worst case, propose an invalid transaction, *e.g.*, wrong result of smart contract execution. The other issue is that using one consensus procedure for each inserted transaction, without changing the way how transactions are inserted in the current version of appendable-block blockchains, can lead to scalability problems. A solution to these problems is to separate devices into different contexts. Consensus will be executed inside each context, and then propagated to gateways from different contexts. Thus, a new field called context should be added to the Block Header (presented previously in Fig. 4.2. This will allow to define that a device will participate in a specific context. The definition of which context a device will be part of is made by gateways during the device's block insertion. The rules to define this can be based on the type of information handled (gas sensor, lightning sensor, etc.) or other definitions that an organization/consortium will agree upon previously. In our proposal, we assume that the definition of contexts is based on existing predefined rules.

Context-based consensus consists of different contexts, where each context contains a number of devices. Consensus is performed in a context independently from other contexts. Consequently, each context can have different consensus or different parameters to be considered to append new information in the blockchain. Gateways can participate in consensus of different contexts, allowing them to participate in different consensus mechanisms (see Fig. 5.2). For example, Context Blue (*CB*) is composed of a set of gateways {*GW A, GW B, GW C, GW D, GW E*}, Context Yellow (*CY*) = {*GW E, GW F, GW G, GW H, GW I, GW J*}, and Context Red (CR) = {*GW E, GW F, GH K, GW L, GW M, GW N*}. In this example, the consensus algorithm used in CB can be different from the consensus algorithm used in CY and CR.

After a consensus is performed inside a context, a gateway can share/propagate the new consented set of transactions to the gateways from the other contexts. For example, after a consensus in *CR*, *GW F* can share new information from *CR* with gateways in *CY*. This can be performed in two different approaches:

- (*i*) using the existing approach by sending the transactions with signed votes to a list of known gateways that do not participate in the context in which the consensus was performed, *e.g.*, *GW D* in Fig. 5.2 can share a set of valid transactions from CB with gateways to known gateways from CR. Every gateway that receives a transaction from known devices can share that with other gateways;
- (*ii*) using a new approach by sending transactions from a context or specific devices when they are requested by a gateway (on demand), *e.g.*, if *Gw I* wants to ensure that it has an updated view from a device from CB.

Approach (*i*) is similar to what is adopted in dBFT [71] and any other consensus algorithm that has a limited group of nodes performing consensus. This approach maintains an updated view (but not synchronous) of all transactions from every node. One issue that

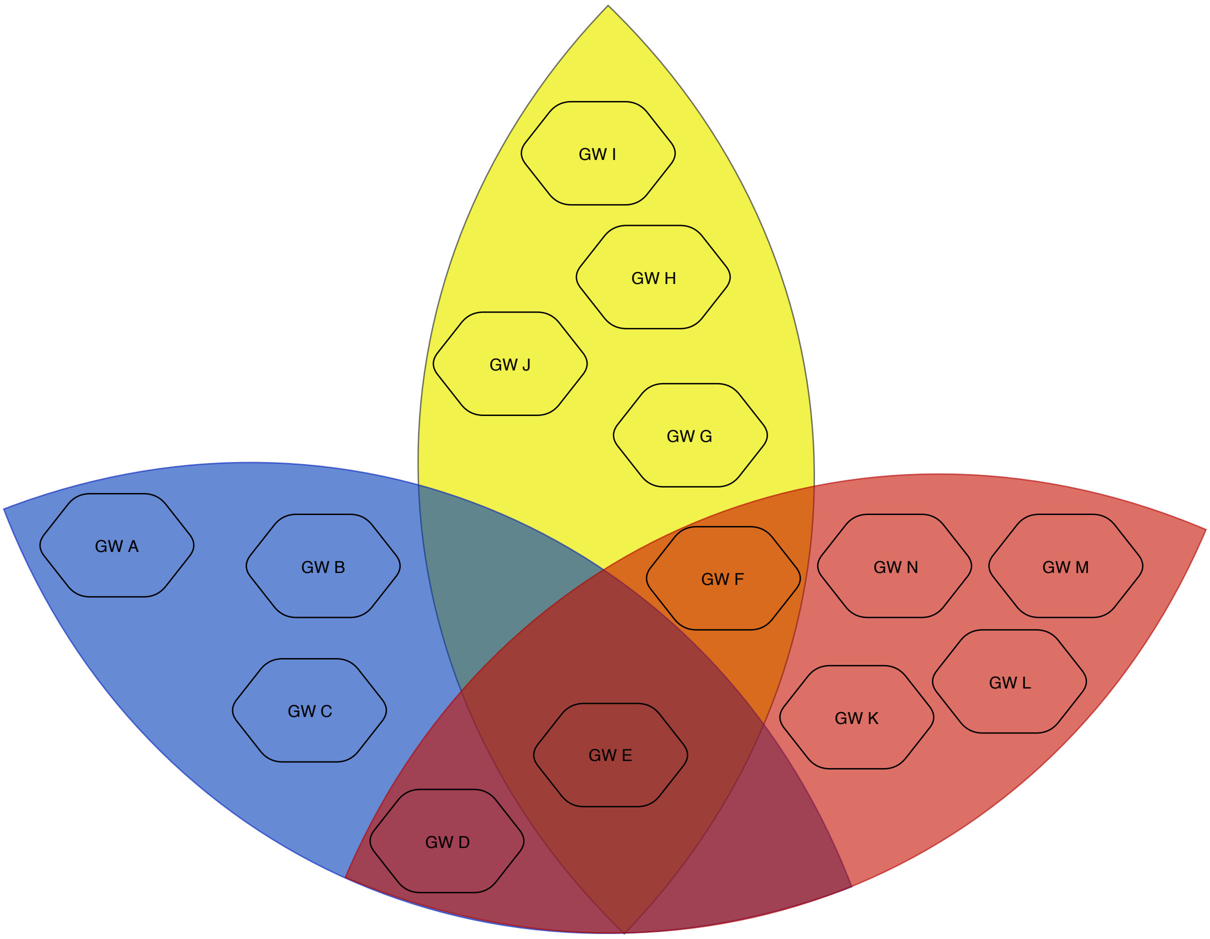


Figure 5.2: Gateways in three different contexts (extracted from [214]).

this approach can have is related to scenarios of a large amount of contexts and, consequently, many messages are exchanged between gateways to update all gateways' ledgers. However, the number of messages will be far less than performing the consensus by all gateways in the blockchain.

Approach (*ii*) can be adopted as a mechanism to avoid many update messages and, also, it can be used as a mechanism to update gateways that do not participate in the same context when requested. As the gateways do not participate in the consensus for that context, the information about transactions may not be used by that gateway. Additionally, this approach can be used to reduce the amount of data that is stored in each gateway. These data can be required if a gateway needs to use them for some processing, decision making, or they are requested by a Service Provider. This approach does not affect the replication of block headers, but can compromise the reliability and the number of copies of transactions.

Each context can have different configurations or different consensus algorithms. Each round can be defined by a set of transactions, that can be designed in different configurations:

- (*a*) one transaction (from that context) per time;
- (*b*) a set of transactions (from that context) generated during the time required to perform the previous context without a limit of transactions;
- (*c*) fixed maximum number of transactions per consensus.

Configuration *a* presents the same configuration used by the current version of appendable-block blockchain to insert transactions, *i.e.*, one transaction per time. This configuration can have a reduced latency to process the transaction for a device in a scenario that gateways are not overloaded. It is a simple approach, and each gateway can start a round of consensus. However, it can lead to a high number of consensus performed in a scenario with a high number of devices or a high rate of updates from devices from a context. In the end, the latency can be increased by the bottleneck in gateways.

Configuration *b* presents the same configuration available in many blockchain, *i.e.*, a limited set of transactions for each consensus. This configuration can reduce the number of consensus performed in the same context and, as a consequence, reduce the number of messages. However, it can lead to more time spent to verify all transactions and, as a consequence, more time can be required to perform the consensus. The gateway that starts the consensus (also known as leader) has to use all transactions produced in the context. However, this approach can increase the number of messages exchanged before the consensus (every gateway will have to send proposed insertions to the leader when requested). A problem with this approach is that overloading the gateways with many new transactions can lead to a time-consuming consensus.

Configuration *c* presents an alternative configuration, which may help to avoid high latency (or starvation) of transactions in overloaded situations. However, this configuration can increase the latency to insert a single transaction, but the number of messages exchanged will be reduced. The gateway that starts the consensus (also known as leader) has to use a limited set of transactions from the context. A problem that can happen is when too many transactions are produced in a small amount of time, *i.e.*, this approach can have a problem to handle an overloaded situation.

A context-based approach can reduce the number of messages exchanged to perform consensus for the transactions. However, some issues can happen when using this approach. For example, gateways that participate in many contexts can have issues regarding the high number of consensus messages, *e.g.*, GW E (in Fig. 5.2) participates in all contexts. A maximum amount of contexts for each gateway should be defined. Also, scenarios in which a small number of gateways participate in the consensus for a particular context is susceptible to attacks, similarly to shard approaches [123] or consensus with limited gateways [71].

Lunardi *et al.* [215] discussed the usage of PBFT in appendable-block blockchains to insert new blocks. That approach can still be used to insert new blocks, *i.e.*, having different consensus to block insertion similarly to the ones used for transaction insertion. In the next subsections, we present the algorithms for each different configuration.

5.1.1 Consensus for single transactions (Configuration a)

New *data* that are produced by a node from a specific context (C_j) will be processed by a gateway (Gw_i) from that context, and it will be sent for a consensus. The *prepareConsensus(Tm)* and *commitConsensus(Tm)* functions used can be different for each scenario. Although, we assume, in this work, that operations are the same as used in PBFT[53], *i.e.*, every node receives a copy of the transaction in the prepare phase, and sends the vote to every other gateway (in the same context C_j) approving or not the new transaction on commit phase.

Algorithm 5.1 Perform transaction consensus - Configuration a

```
Require: $Info_m$ and $Di$
$validInfo \leftarrow$ verifyInfo($Info_m$)
if $validInfo$ is true then
    $T_m \leftarrow$ createTransaction($B$, $Info_m$, $NPK_i$)
    for all $Gw_i$ in $D_j$ do
        prepareConsensus($T_m$)
    end for
    for all $Gw_i$ in $D_j$ do
        $responseList \leftarrow$ commitConsensus($T_m$)
    end for
    if $|positive(responseList)| > minResponses$ then
        addTransaction($T_m$))
    end if
end if
```

5.1.2 Consensus for unlimited transactions in a time-window (Configuration b)

A gateway Gw_i will receive new *data* produced by a node from a specific context C_j. After processing that data, Gw_i will send it to a transactions list (or pool) and then process it in the next consensus.

Algorithm 5.2 Send transactions pool - Configurations b and c

```
Require: $Info_{m}$ and device $NPK_{i}$
 1: $validInfo \leftarrow$ verifyInfo($Info_{m}$)
 2: if $validInfo$ is true then
 3:     sendTransactionPool($Info_{m}$, $BH_{b}$)
 4: end if
```

Algorithm 5.3 Perform transaction consensus - Configurations b and c

```
Require: $transactionPool$ and $C_{j}$
 1: $setT_{m} \leftarrow$ getTransactions($transactionPool$, $z$)
 2: $validInfo \leftarrow$ verifyInfo($Info_{m}$)
 3: if $validInfo$ is true then
 4:     $T_{m} \leftarrow$ createTransaction($B$, $Info_{m}$, $NPK_{i}$)
 5:     for all $Gw_{i}$ in $D_{j}$ do
 6:         prepareConsensus($setT_{m}$)
 7:     end for
 8:     for all $Gw_{i}$ in $D_{j}$ do
 9:         $responseListperT \leftarrow$ commitConsensus($setT_{m}$)
10:     end for
11:     for all $T_{k}$ in $responseListperT$ do
12:         if $|valid(responseList)| > minResponses$ then
13:             addTransaction($T_{k}$))
14:         end if
15:     end for
16: end if
```

Differently from *Configuration a*, we assume, in *Configuration b*, that operations prepare and commit use a set of transactions that will be voted as valid or not. It is important to note that variable *z* (line 1 in Alg. 5.1), which represents the limit of transactions, is set to zero (no limit is used in Configuration b). Also, we assume that a leader is elected for each consensus round. Similarly to *Configuration a*, we based the prepare and commit phases in what is adopted by PBFT [53]. Thus, every node receives a copy of the set of transactions in the prepare phase. After that, on the commit phase, every node sends the vote (approving or not each transaction in the set) to all other gateways (in the same context C_j). As a result, there is a list of votes (from all gateways) for each transaction.

5.1.3 Consensus for a fixed maximum number of transactions (Configuration c)

Similar to *Configuration b*, all *data* produced by a device from a specific context C_j will be processed by Gw_i from that context. Also, a list will be processed in the consensus (Alg. 5.2). We assume, in *Configuration c*, that operations prepare and commit use a set of transactions with a predefined limit (*z* in line 1 in Alg. 5.3) that will be voted as valid or not. Also, we assume that a leader is elected for each consensus round. Also, every time that a consensus is finished a new one will be started but with a maximum amount of transactions per time, *i.e.*, the consensus is based on the time for each round but with a limited amount of z transactions.

5.2 Multiple consensus

Context-based consensus can have some benefits, such as increasing the throughput and reducing the latency in the transactions insertions. But an important improvement is the possibility to use different configurations/approaches for each context. This can help to provide a solution that fits in a large variety of applications. For example, considering the adoption of appendable-block blockchain in a smart building. We can imagine that information produced by smoke detectors can be updated to all gateways from the blockchain (approach I), at the same time that information produced by lightning system is sent to other gateways only when requested (approach II).

Another important feature is that using multi-level consensus (consensus at the block level and transaction level) allows appendable-block blockchains to use different consensus algorithms. For example, a more strict configuration can be defined for insertion of the new blocks than it is used to insert transactions. Also, at the transaction insertions, different consensus can be performed for each context. This feature can help in the flexibility and adaptability of appendable-block blockchains to be used in complex and heterogeneous IoT scenarios.

5.3 Security Discussion

The usage of contexts and different consensus can help to use appendable-block blockchains in different scenarios. However, some security issues should be discussed. Firstly, using different consensus can expose the blockchain to a large variety of attacks. For example, PoW consensus algorithm can be more susceptible to certain attacks (*e.g.*, double spending and fork based attacks) than PBFT (*e.g.*, sybil and eclipse atacks). These attack are discussed in more details in Section 7.2.

Another important issue is related to context-based consensus for transactions. This can introduce 2 mains issues:

- The number of gateways in each context;
- The resilience of the information when using approach II.

The number of gateways in each context can affect the number of faulty (or malicious) nodes supported by the consensus. For example, to support 2 malicious nodes we need at least 7 nodes in each context when using PBFT. It means that if we want to protect the information produced in a context against 2 malicious/tampered gateways, each context should be composed at least of 7 nodes for the consensus procedure. Additionally, when a context adopt the approach II, only nodes from that context will maintain the information, unless other gateway request for that. This is interesting approach to reduce the number of messages and the storage required, but can decrease the resilience of the information produced. In Section 6.4 we present a discussion about the impact in performance when reducing the number of gateways and using approach II.

5.4 Chapter Summary

In this chapter we presented the multi-level consensus for appendable-block blockchains. This proposal can help to provide a solution that better fit different IoT contexts needs. For example, different applications with different requirements can have different consensus algorithms being used to insert information in the blockchain. The context-based consensus can also provide a higher throughput due to the parallelism of the solution. More details about performance is discussed in Chapter 6.

6. EXPERIMENTAL EVALUATION

We evaluated consensus algorithms in appendable-block blockchain in different scenarios and in different stages of development of appendable block blockchains. We separated the evaluation in 2 different categories: (*i*) consensus for blocks, which presents the evaluation of consensus performed to insert new blocks (no consensus is performed for transactions); and (*ii*) multi-level consensus evaluation, which we discuss the evaluation of consensus for both blocks and transactions.

In the next sections, we discuss two different evaluation for each of the categories. Also we present a summary of the findings.

6.1 Consensus for blocks: initial consensus evaluation

As presented previously on Chapter 4, appendable-block blockchains use an hierarchical architecture and an appendable data structure, which allows to insert transactions in parallel across nodes. In our initial evaluation, we performed consensus only to insert a new block for a node. The discussions presented in this section were previously presented in [215].

We used the CORE emulator platform [5] to evaluate the performance of PBFT and simplified consensus algorithm based on witness. The evaluation was run on a VMware Fusion 8.5.10 with 6 processors and *12GB* of *RAM* on an Intel *i7@2.8Ghz* and *16GB* of *RAM*. We performed the evaluation using 10 gateways, where each gateway runs in a container based-virtualized machine; in 9 different scenarios (as presented in Table 6.2) using 100, 500 and 1000 devices connected through theses gateways (10, 50 and 100 per gateway) and 100, 500 and 1,000 transactions per device (*e.g.*, 1,000,000 transactions in Scenario I). All the scenarios are presented in Table 6.1. For each scenario, we performed experiments with PBFT and witness-based consensus.

Table 6.1: Description of 8 scenarios evaluated (from "A" to "I")

	Scenarios								
	A	B	C	D	E	F	G	H	I
Number of connected Devices per Gateway	10	10	10	50	50	50	100	100	100
Number of produced transactions per Device	100	500	1K	100	500	1K	100	500	1K
Number of blocks	100	100	100	500	500	500	1K	1K	1K
Number of transactions produced (total)	10K	50K	100K	50K	250K	500K	100K	500K	1M

6.1.1 Metrics

In order to perform an evaluation of the four different consensus on appendable-block blockchain, we adopted 4 different metrics:

- **T1**: Time to reach the consensus and insert a new block (first time that device is connected) in the leader gateway. This metric represents the time of the consensus not considering the time that all other gateways take to insert the block;
- **T2**: Average time to of processing and inserting a new block in the block ledger of the consensus leader. This time helps to understand the cost (in time) to proceed with block insertion operation while the consensus leader still receiving information from other nodes;
- **T3**: Average time to of processing and inserting a new block in the block ledger of the consensus leader. This time helps to understand the cost (in time) to proceed with block insertion operation in other nodes;
- **T4**: Average time to insert a transaction in the blockchain after a gateway receives it. This metric represents the overhead of the transaction insertion procedure;
- **T5**: Average time to insert a transaction in the blockchain for all gateways (from when it is created to its insertion in the ledger of each gateway). This is important to measure the gateway performance for each transaction insertion (not only in the gateway that is communicating with the device).

All metrics represent the average time in milliseconds (ms) of ten repetitions for each scenario, and using a confidence interval of 95% (represented by error bars in the charts).

6.1.2 Results

All times presented in Table 6.2 represent the average time considering the whole execution in all gateways.

The Witness-based consensus was used as a baseline in terms of time to append blocks and information. As expected, it can be observed in Table. 6.2 that varying the consensus algorithm has impact in the performance in the task to achieve consensus (metric **T1**) on inserting a block (used to insert block header with public key of each device). For example, in Scenario A, witness-based consensus takes 58.20ms to achieve the consensus against 102.82ms using PBFT and in Scenario I (scenario with highest number of devices

Table 6.2: Performance Evaluation in 8 different scenarios (all times in milliseconds)

	A	B	C	D	E	F	G	H	I
T1 - Witness-based	58.20	64.01	65.25	64.51	71.02	71.73	69.13	72.47	79.22
T1 - PBFT	102.82	119.53	121.68	121.98	126.56	132.37	129.14	136.86	160.35
T2 - Witness-based	3.72	3.56	4.42	4.66	4.82	5.81	5.33	5.95	6.28
T2 - PBFT	3.40	4.45	5.16	4.21	4.87	5.88	5.29	5.93	6.52
T3 - Witness-based	0.22	0.22	0.23	0.22	0.23	0.23	0.23	0.24	0.25
T3 - PBFT	0.22	0.22	0.23	0.23	0.23	0.26	0.24	0.24	0.27
T4 - Witness-based	2.66	2.82	2.91	3.24	3.49	3.54	3.89	4.29	4.28
T4 - PBFT	2.69	2.80	2.90	3.30	3.46	4.00	3.96	4.16	4.55
T5 - Witness-based	0.94	1.18	1.48	1.30	1.58	1.89	1.73	2.11	2.33
T5 - PBFT	0.94	1.17	1.47	1.31	1.55	2.03	1.73	2.03	2.39

and transactions), witness-based consensus takes 72.47ms against 160.35ms using PBFT (more than twice the time). However, witness-based consensus is more likely to be affected by different attacks (*e.g.*, Eclipse and Sybil attacks) in comparison to PBFT.

In the other blockchain operations - for instance, time to add a new block in the leader gateway (metric **T2**), as well as the time to update the blockchain (metric **T3**), to append a new transaction in a gateway (where devices are connected to) (**T4**) and to update the blockchain with the new transaction (**T5**) - presented few or no impact using both consensus algorithms. However, the number of transactions and nodes influenced in the processing time to append a transaction (**T4** + **T5**) in the most demanding scenario (Scenario I) takes less than 7ms to both append the transaction (4.28ms in Witness-based and 4.55ms in PBFT) and to update a new transaction in the other gateways (2.33ms in Witness-based and 2.39ms in PBFT). Also, consensus algorithm has few impact in the average time to append transactions in the block, as can be observed in Figure 6.1.

Additionally, it can be observed that growing the number of transactions (overload of processing in gateways) has more impact than the number of devices that a gateway is handling. For example, scenario D has half of transactions and 5 times more nodes than C, but takes almost the same time to reach the consensus for a block. Differently, scenario F has half of nodes and 5 times more transactions than scenario G, resulting in F spending around 3% more time to achieve the consensus than G. Figure 6.2 presents a comparison of the time to achieve consensus of a block in different scenarios.

As a comparison, the Bitcoin network has around 10,000 [37] active nodes in a 24-hour slice, consequently, the experiment in Scenario I represents approximately 10% of the Bitcoin network. As a comparison, the Bitcoin has more than 150,000 confirmed transactions per day [70] with a peak of 490,644 confirmed transaction in a day [38], which means that the evaluation in Scenario I, at least represents more than twice the transactions in the Bitcoin blockchain in a day. A more effective comparison could be made with IOTA [111] - a blockchain developed for IoT - which has around 8.7 transactions per second [299]. This means around 750,000 transactions processed in a day (around 75% of the transactions

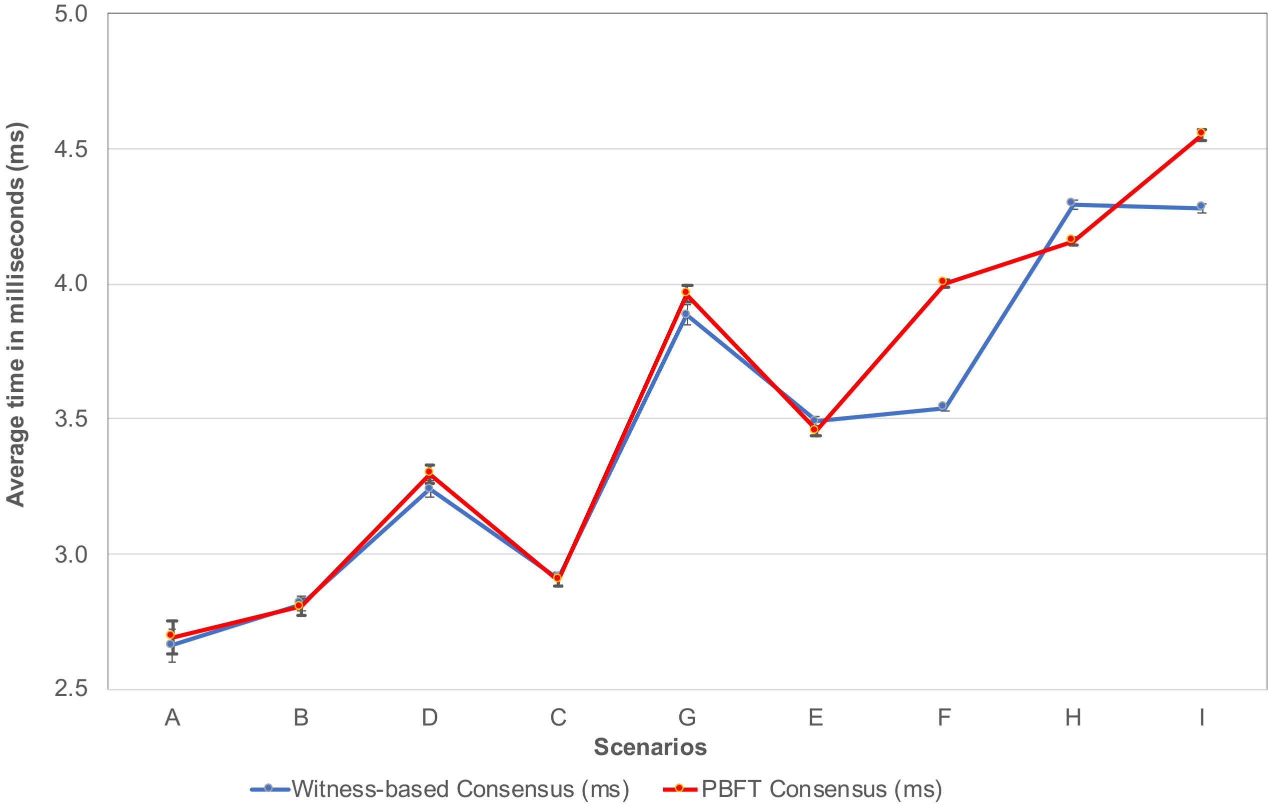


Figure 6.1: Average time to append a transaction in the Gateway connected to the producer Device

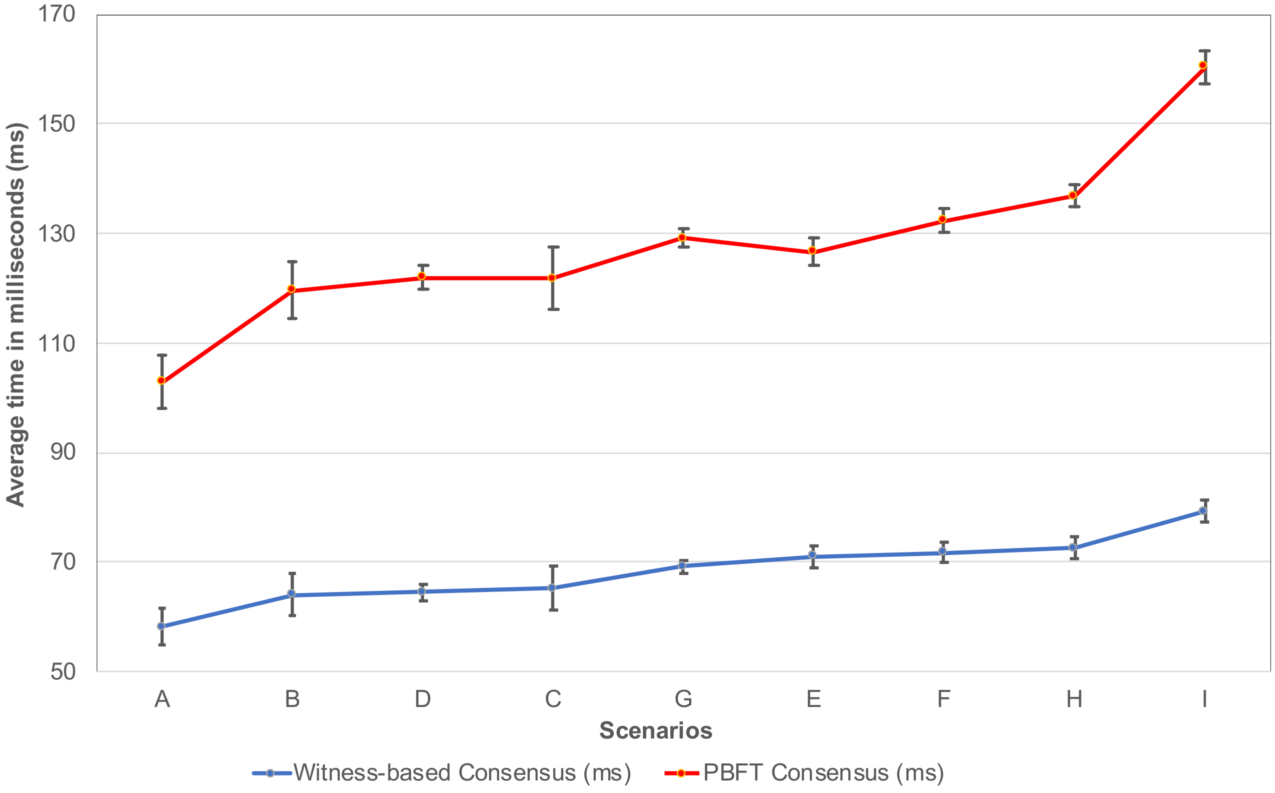


Figure 6.2: Average time to perform the consensus for a block

processed in Scenario I). Also, it represents that IOTA transaction processing time is around 115ms. Consequently, the transactions processing time in our solution represents less than 6% of the time that is spent in IOTA - 115ms in IOTA and 7ms in our solution (4.55ms to append a transaction in a gateway summed with 2.39ms to update the entire blockchain using PBFT). [1]

This presented good results in the emulated IoT scenarios with different number of devices and transactions. It is important to note that the code that implements the proposed blockchain was developed using the Python programming language and a set of libraries. The code is available at GitHub [2] and could be used to replicate the experiments.

6.2 Consensus for blocks: Consensus algorithms comparison

Appendable-block blockchain was improved in order to provide security against byzantine failures and the most common blockchain attacks through the implementation of PBFT consensus algorithm. We used an emulated scenario for smart buildings (again, using CORE emulator) using 10 gateways and varying from 100 (A) to to 1000 (I) devices, where each device produced 100 (A) to 1,000 transactions (I), *i.e.*, more than 1,000,000 total of transactions in the largest scenario.

We evaluated four different consensus algorithms using the SpeedyChain framework (an appendable-block blockchain)[232]: PBFT, dBFT, PoW and Witness-based consensus. We used a PoW with 12 bits for the hash difficulty (number of bits zero required in the first bits of the block hash). This difficulty was chosen due to present performance close to others consensus. Also, we used the same emulated IoT environment presented in the previous work [215]. The IoT environment was emulated using the Core emulator [5] to create a container-based network composed by network equipment, gateways and devices. The experiments were performed on a Virtual Machine (VM) with 6-core processor, 16GB of memory and 64MB of graphics memory running Ubuntu 18.04 operating system using a Virtual Box hypervisor over a Macbook Pro with 2.3 GHz 8-Core Intel Core i9 processor, 32GB DDR4 memory.

For all consensus experiments, a network with ten (10) gateways and one thousand (1,000) devices were used. We generated one million (1,000,000) transactions for each experiment. A transaction on the experiments represents sensors readings (temperature, CO2, etc) signed by a device and signed by a gateway. This type of scenario is similar to the largest and most demanding scenarios evaluated in our previous work [215].

[1] Comparison based on data collected in February of 2018.

[2] https://github.com/regio/r2ac/tree/2019consensus

6.2.1 Metrics

In order to perform an evaluation of the four different consensus on appendable-block blockchain, we adopted 4 different metrics:

- **T1**: Time to perform consensus and insert a new block (first time that device is connected) in the leader gateway. This metric represents the time of the consensus not considering the time that other gateways take to insert the block;
- **T2**: Time to perform consensus and replicate it to all gateways (after consensus). It can be understood as the overall time spent for each block insertion procedure;
- **T3**: Time to insert a transaction in the blockchain after a gateway receives it. This metric represents the overhead of the transaction insertion procedure;
- **T4**: Average time to insert a transaction in the blockchain for all gateways (from when it is created to its insertion in the ledger of each gateway). This is important to measure the gateway performance and average latency for each transaction insertion.

All metrics represent the average time in milliseconds (ms) of ten repetitions for each scenario, and using a confidence interval of 95%.

6.2.2 Results

As expected, witness-based consensus presented better performance for all metrics and PoW presented the worst results. However, witness-based consensus was used as a baseline for the results and it is more likely to be affected by different attacks (*e.g.*, Sybil attacks) in comparison to PBFT and dBFT. Additionally, PoW with 12 bits is not well suited to protect against malicious gateways. As a comparison, Bitcoin's PoW started with a difficulty of 32 bits and, currently, the difficulty is over 70 bits[39]). As shown in Figure 6.3, the consensus procedure (metric **T1**) using witness-based approach was performed in 66.94±8.47ms (first bar, in blue). It is nearly the half of the time expend by the dBFT (second bar, in red) with 118.96±7.69ms, the second best evaluated consensus algorithm. PBFT (third bar, in green) achieved consensus in 138.61±12.47ms and PoW (fourth bar, in yellow) achieved consensus in 149.23±108.84ms. Even using the same number of gateways in dBFT (delegates) and PBFT to perform the consensus procedure, dBFT reduced in 15% the time compared to PBFT. Moreover, average results in PoW present a high deviation due to lottery characteristics of this consensus algorithm (finding a hash with a specific characteristic).

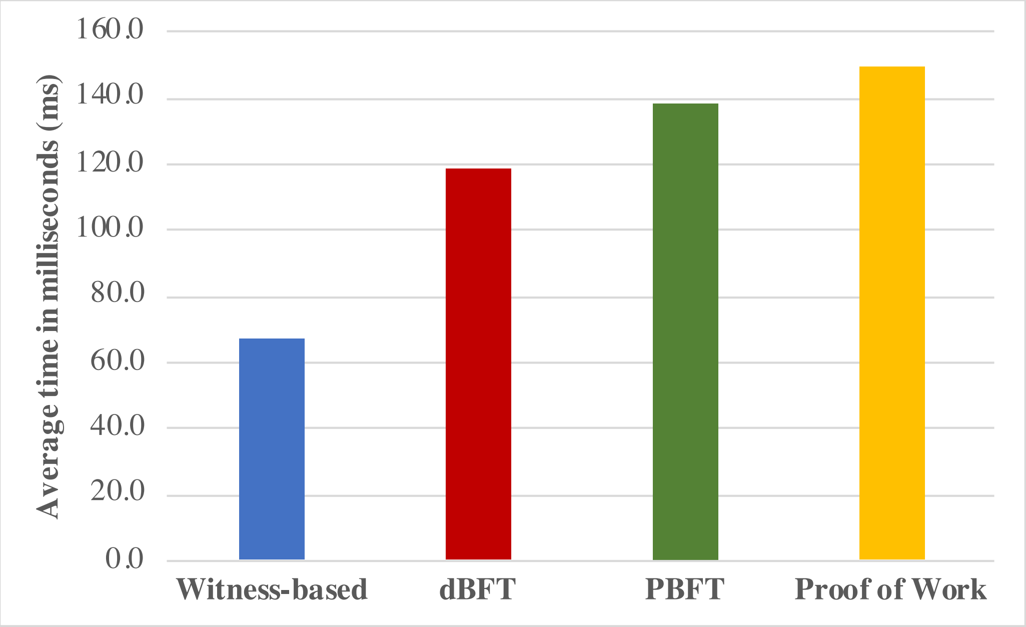


Figure 6.3: **T1**: Average time to perform the consensus

Considering the total time to perform consensus and propagate the block to all gateways (metric **T2**), witness-based approach had similar results to dBFT, an average of 162.69±65.51ms and 196.33±16.06ms respectively (Figure 6.4). In this case, both PBFT and PoW increased in nearly twice the time required to perform **T2**, with an average of 370.71±56.51ms and 390.05±171.65ms. This shows that, considering an overall view of the blockchain network, dBFT can perform the consensus close to witness-based approach, but with much better results than PBFT. Also, it is important to note that, again, PoW presents a high deviation.

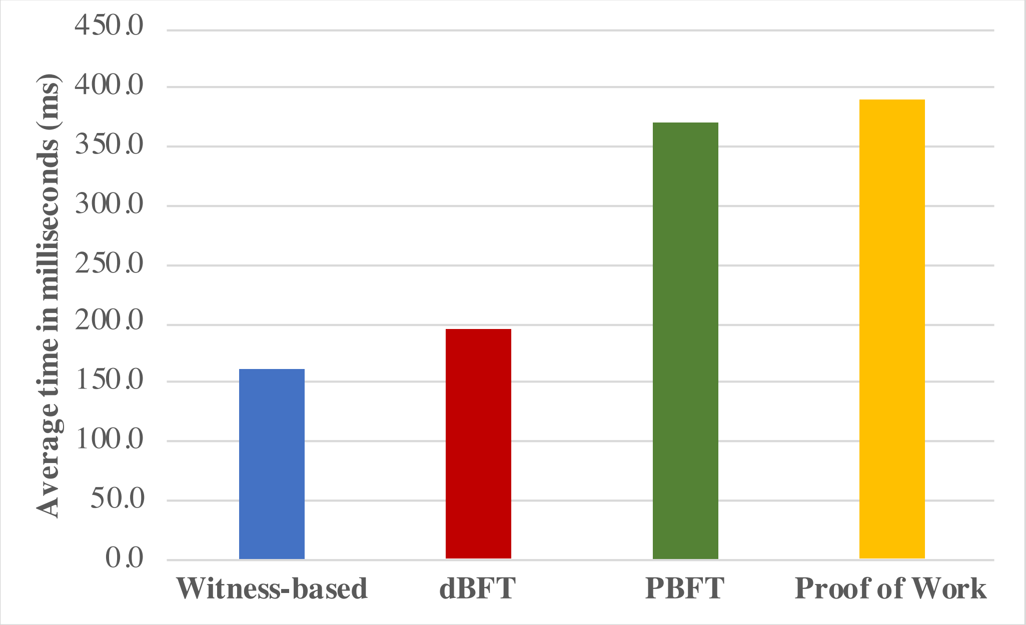


Figure 6.4: **T2**: Average time to perform the consensus and propagate the block

We also analysed the impact of each consensus algorithm on the performance of transactions insertion. The overhead to insert a transaction in the gateway (metric **T3**) presents very similar results in all consensus algorithms (as can be observed in Figure 6.5). For all consensus algorithms, it takes between 4.10±0.24ms (obtained using witness-based) and 4.60±0.35ms (obtained using PoW).

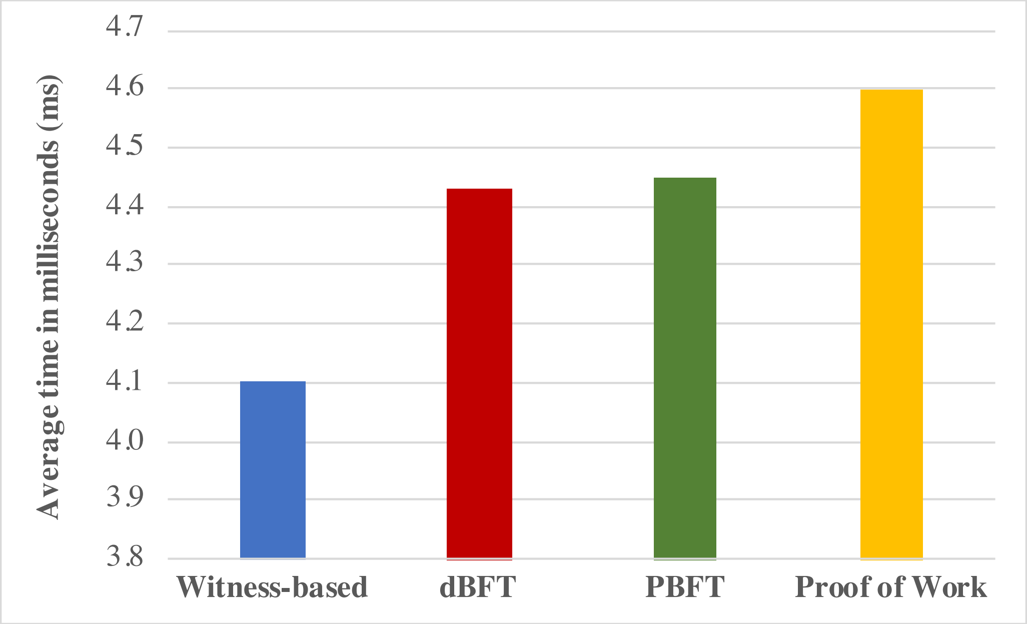


Figure 6.5: **T3**: Overhead of the transaction insertion procedure

An important aspect that should be considered in IoT environments is the latency to insert information. This can impact in the processing of a sensor reading, for example. Consequently, the time that takes to all gateways to insert produced information is crucial. In relation to this metric (T4), good results were obtained in all consensus, varying from 68.31±2.17ms (in witness-based) to 148.71±3.51ms (in PoW)). Also, dBFT (77.60±0.69ms) and PBFT (77.61±1.55ms) presented nearly the same results considering transaction latency.

The evaluation presented good results in the emulated IoT scenarios with the four used consensus algorithms. However, it is important to note that witness-based approach was used only as a baseline to other results and it has some important security issues. Also, using PoW with a difficulty of 12 bits is not suitable in heterogeneous environment (where devices with high computing power can take the power of the mining procedure). Additionally, it is important to mention that the code that implements the proposed blockchain was developed using the Python programming language and is available at GitHub [3].

[3] https://github.com/conseg/speedychain/tree/Multilevel

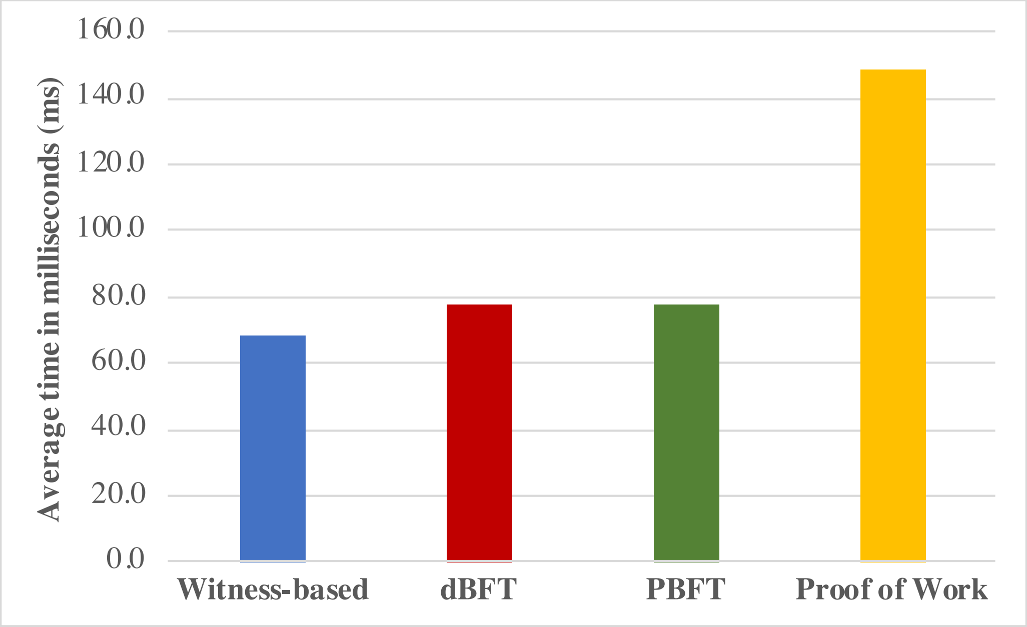


Figure 6.6: **T4**: Latency to insert a transactions in all gateways

6.3 Consensus for blocks:: Geo-distributed IoT

The evaluation presented previously did not consider variables present in real network environments, such as latency in the communication between nodes. Thus, that did not considered network latency and communication in geographically distributed scenarios. Consequently, the evaluation of different consensus algorithms can be impacted when considering geographically distributed IoT networks [149], in special the consensus algorithms that use a high number of messages. Consequently, a misleading latency evaluation can lead to a not properly chosen consensus algorithm to an IoT scenario that is not limited to local network. Also, due to the use of an emulated environment, built on a single VM running over a single physical host, the performance evaluation may have been influenced by the fact that the gateways competed for the same resources (processor, memory and I/O) of the VM and, consequently, of the physical host.

In order to understand the behavior of appendable-block blockchain in a geographically distributed scenario, this work presents the evaluation of different consensus algorithms on appendable-block blockchains. This type of scenario is very important, for example, to control a global disease in a pandemic situation; every country could send data to the closest host possible; and, these data would be collected from different patients, for example, through IoT devices in a wireless network.

Thus, we evaluated performance tests in a geographically distributed IoT environment created on the infrastructure of a commercial cloud services provider. The evaluation environment, shown in Figure 6.7, is composed by Amazon Web Services (AWS) Virtual Private Clouds (VPCs) geographically distributed in different AWS regions.

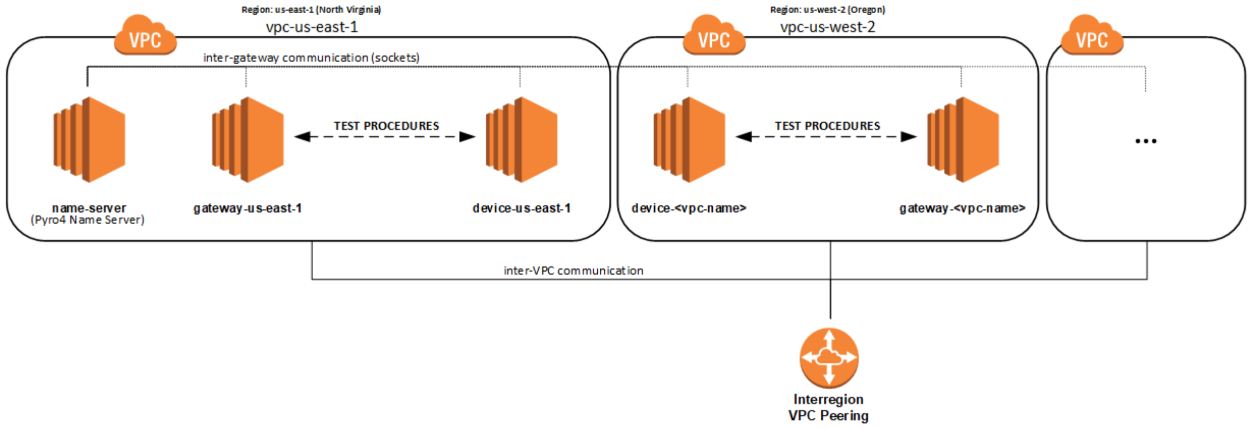


Figure 6.7: Testbed environment (extracted from [76]).

Due to limitations imposed by Amazon, which determines that the use of certain regions should be preceded by an explicit authorization requested by the AWS account manager, the VPCs were created only in 4 regions: Asia Pacific (Tokyo), South America (São Paulo), US East (N. Virginia), and US West (Oregon).

SpeedyChain was encoded using the Python programming language version 2.7. It uses the Pyro4 [78], a native Python library that enables to build applications using Remote Procedure Calls (RPCs) to connect gateways and devices in a peer-to-peer network. Thus, it is necessary to have an instance of the Pyro4 Name Server accessible to all gateways and devices connected to the SpeedyChain network, as shown in Figure 6.7 to help to identify gateway address. In our test environment the name server (name-server - Pyro4 Name Server) runs in the AWS region us-east-1 (North Virginia).

In each AWS region, there is a VPC composed by an EC2 VM running instances of the SpeedyChain gateway and another EC2 VM to simulate the devices. In total, like the simulation performed in Michelin *et al.* [232], the test environment consists of 15 gateway instances. However, unlike that previous work, in this work, the network traffic between gateways, and between gateways and devices is subject to latency times and other characteristics of geographically distributed IoT networks. The evaluation of latency and other characteristics, including the analysis of AWS Service Level Agreements (SLAs), is beyond the scope of this work. Gateways located in the same VPC, that is, in the same network infrastructure, and gateways distributed in other VPCs, that is, geographically distributed between regions are subject to different latency times.

An automation script using the AWS CloudFormation service was created to generate the testing environment. Thus, all the VPC network components were specified in a JSON templateand CloudFormation Stacks were then created based on the template in each of the regions.

6.3.1 Metrics

In order to perform a broader evaluation of SpeedyChain, the time required to perform various operations involved in the insertion of new blocks and new transactions was measured:

- **T1**: Time to process and to add a new transaction into a local gateway;
- **T2**: Time to add a new transaction into a remote gateway;
- **T3**: Time to verify information from a device and to create a new device block (before consensus);
- **T4**: Time to add a new device block into the leader gateway (after consensus);
- **T5**: Time to add and to replicate a device block into all gateways (after consensus).

The measured times were collected using the Python Logger library, which stores the collected data in text format files. The impacts that the collection and time recording operations of the library introduce to the SpeedyChain operation are not evaluated, but we can expect them to be relatively low due to the simplicity of these operations.

The test procedures are shown in Table 6.3. Figure 6.8 shows a graphical representation of the test procedures. Due to time constraints, all test procedures performed simulated devices connected to the AWS region us-east-1 (North Virginia).

Table 6.3: Test procedures.

Devices	**Transactions**	**Consensus**
50	10	None
50	10	PoW (12 bits)
50	10	PoW (16 bits)
50	10	dBFT
50	10	PBFT

6.3.2 Results

After the execution of all planned test procedures and the collection and consolidation of the logs, the average times for each defined performance indicator were calculated. All metrics are represented by an average time (in milliseconds) over a minimum of one hundred repetitions for each scenario, with a confidence level of 95%.

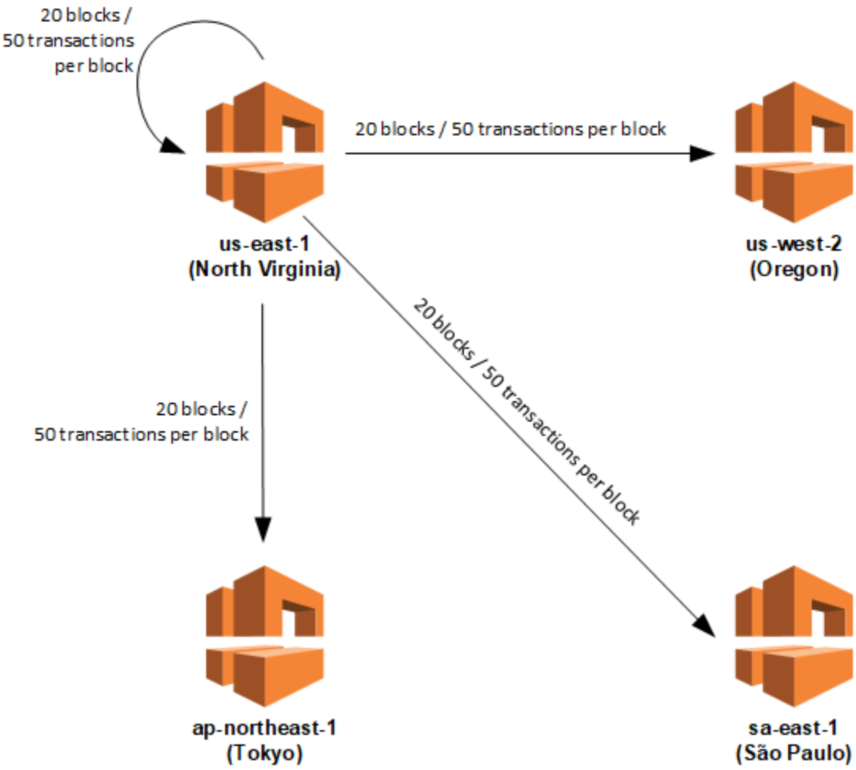

20 blocks / 50 transactions per block
20 blocks / 50 transactions per block
us-east-1 (North Virginia)
us-west-2 (Oregon)
20 blocks / 50 transactions per block
20 blocks / 50 transactions per block
ap-northeast-1 (Tokyo)
sa-east-1 (São Paulo)

Figure 6.8: Test Procedure

For the **T1** metric - Time to process and to add a new transaction into a local gateway, Table 6.4 shows that there is no significant variation in the times measured among the consensus algorithms since it is a local operation that occurred before the consensus execution.

Table 6.4: Indicator **T1**: Time to add a new transaction (local gateway)

Consensus	**Average time** (in milliseconds)
None	4.1473 ± 0.0046
PoW (12 bits)	4.0365 ± 0.0049
PoW (16 bits)	4.2066 ± 0.0110
dBFT	4.0617 ± 0.0091
PBFT	4.1144 ± 0.0045

The same is true for the **T2** metric - Time to add a new transaction into a remote gateway, since this metric is also related to a local operation, *i.e.* a remote gateway must store locally a transaction received from another gateway (see Table 6.5).

T3 metric - Time to verify information from a device and to create a new device block (before consensus), is also a local operation executed prior to the execution of the consensus algorithm, so it should not be affected by that (see Table 6.6).

Differently, the **T4** metric - Time to add a new device block into the leader gateway (after consensus), represents the execution of the consensus algorithm, so it is largely influ-

Table 6.5: Indicator **T2**: Time to add a new transaction (remote gateway)

Consensus	**Average time** (in milliseconds)
None	0.6873 ± 0.0033
PoW (12 bits)	0.5806 ± 0.0007
PoW (16 bits)	0.5818 ± 0.0007
dBFT	0.6142 ± 0.0010
PBFT	0.6514 ± 0.0008

Table 6.6: Indicator **T3**: Time to produce a new device block (before consensus)

Consensus	**Average time** (in milliseconds)
None	0.0336 ± 0.0003
PoW (12 bits)	0.0337 ± 0.0003
PoW (16 bits)	0.0334 ± 0.0003
dBFT	0.0332 ± 0.0004
PBFT	0.0335 ± 0.0003

enced by communication latency, except in the test scenario in which there is no consensus algorithm execution (see Table 6.7). By the obtained results, it is possible to observe that the execution time of the PoW algorithm has a direct influence on the difficulty derived from the number of nonce challenge bits. When increasing that number from 12 to 16 bits, the time spent increased less than expected, *i.e.* around 30%.

Table 6.7: Indicator **T4**: Time to add a new device block (after consensus)

Consensus	**Average time** (in milliseconds)
PoW (12 bits)	1470.8114 ± 6.3921
PoW (16 bits)	1911.1309 ± 96.2341
dBFT	5162.4721 ± 0.8595
PBFT	4250.0792 ± 182.3843

The **T5** metric - Time to add and to replicate a device block to all gateways (after consensus), is directly influenced by network latency times, since it is the measurement of the time needed to replicate a block of a device to the other SpeedyChain nodes (see Table 6.8).

Table 6.8: Indicator **T5**: Time to add a new device block (after consensus)

Consensus	**Average time** (in milliseconds)
PoW (12 bits)	4314.9415 ± 8.4733
PoW (16 bits)	4736.4641 ± 96.5417
dBFT	8230.6308 ± 10.0098
PBFT	11607.8835 ± 234.3939

In the smart building evaluation, the results were not influenced by high latency times and other variables present in distributed IoT environments. Consequently, in those test scenarios, the time required to validate and register a device block with 10 transactions was close to 20ms [232] without consensus and around 102ms using PBFT [215]. In the present work, the measured times ranged from 1,442.5ms, without the execution of a consensus algorithm, to 11,607.8ms with the PBFT consensus algorithm. Hence, we can show the impact that network latency in real environments has on the performance of the SpeedyChain blockchain. This same impact may affect different blockchains.

Due to messages exchanged during the consensus algorithms, latency has impact even in the PoW algorithm, which does not present a high number of exchanged messages. This could be observed when using a low difficulty (12 bits) in the PoW algorithm, in which more than 4,000ms were required to include a new block into the blockchain. The usage of very small difficulty in the PoW algorithm was used to show that latency is an important factor during the consensus procedure.

Consequently, this work shows that current consensus algorithms used in SpeedyChain can be used to geographically distributed IoT environments when latency to create new blocks can be around few seconds. It is important to remember that block insertion is performed only in the first time that a device connects to a gateway or when a device needs to change its key pair. Additionally, improvements can be achieved adapting other consensus algorithms that consider environments with high latency. The code is available at GitHub [4] and could be used to replicate the experiments.

6.4 Multi-level consensus

In order to evaluate context-based consensus algorithms in appendable-block blockchain, we performed testing with a different number of contexts, different configurations and different approaches for updating the nodes. Also, we used the Core Emulator [5] to create a container-based network to emulate network equipment, gateways and devices. For all executed tests, a network with ten (10) gateways and 1,000 devices was adopted in order to emulate a smart building.

We present the description scenarios used in the evaluation in Table 6.9, the configurations used in the context-based consensus in Table 6.10 and the approaches used to propagate the transactions after consensus in Table 6.11. The emulation was performed in a Virtual Machine (VM) with 6-core processor, 16GB of memory and 64MB of graphics memory running Ubuntu 18.04 operating system using a Virtual Box hypervisor over a Macbook Pro with 2.3 GHz 8-Core Intel Core i9 processor, 32GB DDR4 memory.

[4] https://github.com/conseg/speedychain/tree/Multilevel

Table 6.9: Evaluated scenarios.

Scenario	Description
1	1,000,000 transactions sent by 1,000 devices, varying from 1 to 10 contexts, where all gateways participate in all contexts
2	1,000,000 transactions sent by 1,000 devices, varying from 1 to 10 contexts, each context having exactly 5 gateways (gateways can participate in more than one context)

Table 6.10: Evaluated configurations.

Configuration	Description
A	All contexts using PBFT for a single transaction
B	All contexts using PBFT with no limit of transactions per consensus
C	All contexts using PBFT with limited number of transactions (100, 1000 and 10000) per consensus

Table 6.11: Evaluated approaches.

Approach	Description
I	After the consensus, transactions are sent to all gateways that do not participate in the context
II	After the consensus, transactions are not sent to gateways that do not participate in the context

In *Scenario 1*, we intend to show how multiple contexts can perform when all gateways participate in all contexts. This is to show the most demanding scenario due to high processing and communication demand. In *Scenario 2*, we intend to show the impact of limiting the number of contexts that a gateway can participate in on latency and throughput. Unlike *Scenario 1*, in this scenario it is not possible for a gateway to participate in all contexts as there are only five gateways in every context. Hence, it is possible to have some gateways that participate in multiple contexts.

Also, we evaluated the 3 configurations proposed in Section 5.1 and presented in Table 6.10. These different configurations were evaluated to show how the number of transactions in each consensus affects the throughput and latency in context-based consensus. Finally, as presented in Table 6.11, we used 2 different transaction update approaches: inserting transactions in all gateways that do not participate in the consensus, or not inserting them while they are not requested. These approaches were evaluated only for Scenario 2. Thus, we used only the approach I code for Scenario 1 since all gateways belong to all contexts, *i.e.*, they do not need any additional updates. Consequently, we executed 150 different tests, as a result of different combinations of scenarios, transaction limit configurations in each context, different transaction propagation approaches and a different number of contexts used in each test.

6.4.1 Metrics

We used two metrics to evaluate context-based consensus for transactions in appendable-block blockchains, *i.e.* **latency** and **throughput**. Latency was calculated based on the time spent from creating a transaction to inserting it in the blockchain. Consequently, the latency captures the whole time spent in different processes such as the time it takes to propagate the transaction to gateways, the time the transaction spends in the transaction pool, and the time spent in the consensus. We considered as **throughput** the rate of insertion of transactions per second (tps) in the blockchain. It is important to note that the evaluation was performed in a local network, where the communication times are reduced.

6.4.2 Results

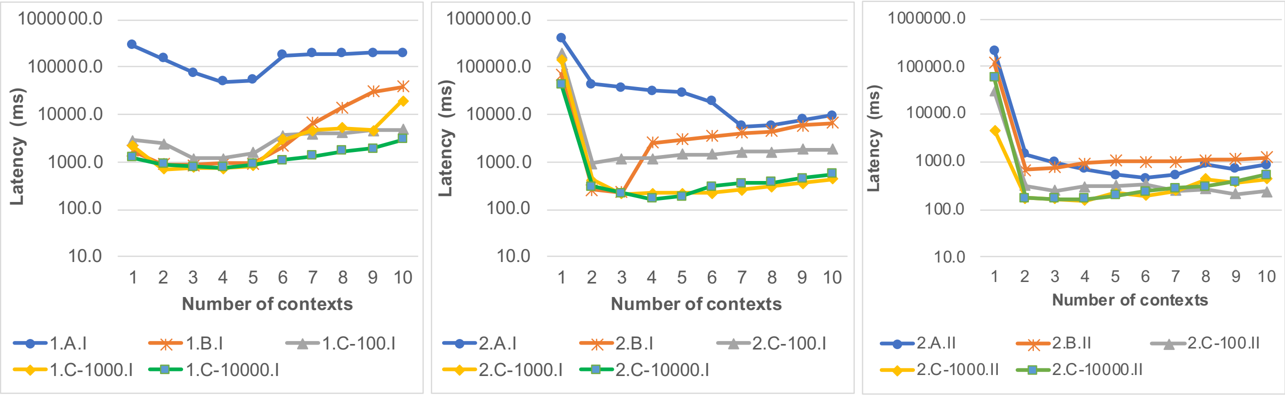
We can observe in Table 6.12 the average (with the 95% confidence interval) transactions **latency** (in milliseconds) in all scenarios, approaches and configurations. Hence, lower latency results are better. We present in the table only the results for 1, 2, 4 and 8 contexts to help the visualization of the data. The first row indicates the scenario (1 or 2), the configuration (A, B and C, where C can take 100, 1,000 and 10,000 transactions) and update approach (I or II). For each scenario/configuration/approach, we collect results from 1 to 10 contexts (for one context, all devices in that context; for two contexts, half of the devices in each context; and so on).

We can observe that when using only one context (in all scenarios, configurations and approaches), the average latency is always higher than 10,000ms (10 seconds), indicating that using only a single context, *i.e.*, only one consensus for all transactions, is not sufficient to insert a transaction before a new one is produced by the same device (every 10 seconds). Additionally, considering two or more contexts, for almost all cases configurations/approaches, *scenario 1* presented worse results than evaluation over *scenario 2*. For *scenario 1*, the lowest transaction latency was 706.5±1.3ms using two contexts with *Configuration C* (with limit of 1,000 transactions). This value is more than 463% of the best result (**152.5±0.3ms**) in *scenario 2* (four contexts with *Configuration c* with limit of 1,000 transactions and update *approach II*). Consequently, the results show that the number of gateways in each context can impact the latency.

In order to help to better understand the differences between transaction limit configurations and update approaches, we present the results separated in scenarios in Fig. 6.9 using logarithmic curves. We can observe that best results for two or more contexts are achieved by *Configuration c* with 1,000 and 10,000 transactions, represented respectively by yellow (with diamond) and green (with square) lines. In special for both evaluations over

Table 6.12: Latency (in milliseconds) to insert transaction.

Scen.Conf.Appr.	**Number of Contexts**			
	1	**2**	**4**	**8**
1.A.I	284867.0±336.1	150477.9±577.2	48413.7±407.1	189562.5±731.9
1.B.I	168544.0±245.3	881.1±1.3	906.9±2.5	14328.5±32.1
1.C-100.I	287921.4±544.4	2402.2±5.4	1152.5±3.8	4106.3±8.3
1.C-1000.I	210587.2±359.4	706.5±1.3	734.2±2.2	5160.3±27.2
1.C-10000.I	122431.8±353.8	895.8±2.1	762±1.8	1661±3.9
2.A.I	402376.2±1294.3	43833.7±194.5	31170.9±217.8	5714.2±49.0
2.B.I	70143.3±226.3	256.7±0.6	2507.3±4.6	4455.8±18.1
2.C-100.I	189105.4±345.6	937.1±2.5	1186.9±2.8	1619.8±15
2.C-1000.I	145247.4±262.9	416.7±0.9	216.6±0.6	303.4±2.2
2.C-10000.I	42636.7±117.4	305.8±0.7	168.0±0.4	363.2±1.2
2.A.II	214997±804.3	1407.9±4.4	686.9±4.4	897.3±5.6
2.B.II	118258.2±484.4	670.0±2.0	924.6±2.3	1095.4±6.2
2.C-100.II	30212.2±170.7	309.3±0.8	306.1±2.8	261.7±0.7
2.C-1000.II	46555.1±275.6	173.3±0.6	152.5±0.3	430.2±1.5
2.C-10000.II	56736.4±89.3	169.5±0.4	164.6±0.4	302±0.8



(a) Scenario 1

(b) Scenario 2, approach I

(c) Scenario 2, approach II

Figure 6.9: Average latency for each transaction in context-based consensus.

scenario 2, the average latency was under 1 second for two or more contexts using *Configuration c* with 1,000 and 10,000 transactions. Additionally, *approach II* had better general results than *approach I*. Also, it is important to note that *approach II* using *Configuration c* (100, 1,000 and 10,000) achieved an average latency lower than 550ms for 2 or more contexts. Thus, we can assume that a context with less gateways (*scenario 2*), with a configuration with a limit between 100 and 10,000 transactions, and updating by request can have a reduced latency.

As expected, when considering only one context, the **throughput** was considerably lower than with multiple contexts, as presented in Table 6.13. Similar to what happened to latency, best results were obtained when less gateways per context were used (*scenario 2*). As a comparison, the best result was obtained when using three contexts, in *scenario 2*, with *Configuration c* with limit of 100 transactions and *approach II* (not updating), having a consensus throughput of **154.8±1.4tps**.

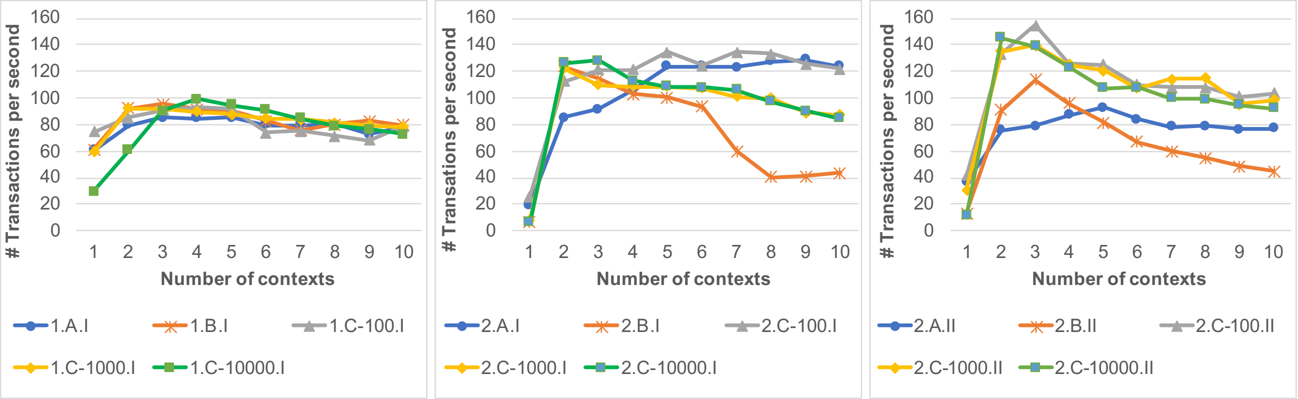
It is important to note that throughput is affected by many factors. The number of transactions is an important aspect, for one transaction in each consensus (*Configuration a*) means a consensus procedure that will be performed for just one transaction. Although, a high number of transactions in each consensus means more time to verify and perform consensus. Additionally, increased load on gateways during parallel execution of consensus and more messages exchanged (during the consensus or receiving updates from different contexts) can affect the performance. Furthermore, it is important to note that a higher rate of transactions would affect the throughput. However, in all evaluation instances we used the same number of devices, gateways and transactions in order to have the same parameters for all 150 different performed tests.

Table 6.13: Transactions throughput (transactions per second) using context-based consensus.

Scen.Conf.Appr.	**Number of Contexts**			
	1	**2**	**4**	**8**
1.A.I	30.26±0.4	79.4±0.4	84.6±0.5	80.4±0.6
1.B.I	30.48±2.0	92.2±1.5	91.7±0.9	79.9±1.2
1.C-100.I	37.28±1.5	85.4±1.4	93±0.8	71.5±2.3
1.C-1000.I	29.66±2.1	92.4±1.3	89.7±1.1	81.4±0.9
1.C-10000.I	29.8±0.4	60.6±0.7	99.4±2.0	78.6±3.2
2.A.I	19.3±0.2	85.2±0.7	105.9±1.3	127.7±1.5
2.B.I	6.4±0.6	123.9±1.2	102.7±0.8	40.6±0.6
2.C-100.I	26±1.0	112.1±1.7	121.4±2.5	133.3±2.4
2.C-1000.I	8.1±0.7	122.6±1.1	108.6±0.8	99.5±0.6
2.C-10000.I	6.2±0.5	126.1±1.6	113.1±0.8	97±0.7
2.A.II	36.4±0.2	75.7±0.4	87.3±0.4	78.8±0.3
2.B.II	13.5±1.2	91.2±2.0	96.2±2.0	54.8±1.0
2.C-100.II	41.5±1.2	133.2±1.1	125.8±0.9	108±0.6
2.C-1000.II	30.7±0.9	134.9±1.1	124.7±0.9	114.9±0.8
2.C-10000.II	11.1±1.4	145.3±1.6	123.1±0.8	99±0.6

Fig. 6.10 shows the impact of the number of contexts, configurations and update approaches on the throughput. We can observe that increasing the number of contexts can improve throughput. This shows that parallelism of insertion using different contexts can help to improve appendable-block blockchains performance. Additionally, between 2 and 6 contexts in scenario 2 (for both approach I and II) and using configuration C (100, 1,000 and 10,000 transactions limit) the throughput is above 100 transactions per second.

This experiment shows that the context-based approach can present improvements both over a single context (or non-existence of contexts) and to a single transaction insertion in the blockchain. Also, our evaluation shows that context-based consensus can guarantee



(a) Scenario 1

(b) Scenario 2, approach I

(c) Scenario 2, approach II

Figure 6.10: Average transactions consensus throughput (transactions per second inserted in the blockchain).

lower latency and higher throughput. However, there are some threats to validity of our evaluation since the evaluation tests were performed in a controlled environment.

The code is available at GitHub [5] and could be used to replicate the experiments.

6.5 Multi-level consensus: different consensus algorithms

In order to evaluate different consensus algorithms fo context-based consensus algorithms in appendable-block blockchain, we compared the results of using PBFT (as presented in Section 6.4) and a Proof-of-Authority (PoA) consensus at the transaction level. Also, we used the same infrastructure used in experiments performed in Section 6.4, i.e., we used Core Emulator [5] to create a container-based network to emulate network equipment, gateways and devices. For all executed tests, a network with ten (10) gateways and 1,000 devices was adopted in order to emulate a smart building.

We present the description of the 2 scenarios used in the evaluation in Table 6.14. The emulation was performed in a Virtual Machine (VM) with 6-core processor, 16GB of memory and 64MB of graphics memory running Ubuntu 18.04 operating system using a Virtual Box hypervisor over a Macbook Pro with 2.3 GHz 8-Core Intel Core i9 processor, 32GB DDR4 memory.

In *Scenario 1*, we performed the same experiment performed in the case 1.C-100.I from Section 6.4. We intend to use this case due to not present the worse nor the best results for that evaluation, i.e., it can be used as an average case. In *Scenario 2*, we intend to show the impact of using different consensus algorithms for block level and transaction level. With this experiment we can compare how the multi-level consensus can be adapted to use different consensus algorithm for both block and transactions. As a simplification, we

[5] https://github.com/conseg/speedychain/tree/speedychainApp

Table 6.14: Evaluated scenarios using PBFT and PoA.

Scenario	Description
1	1,000,000 transactions sent by 1,000 devices, varying from 1 to 10 contexts, where all gateways participate in all contexts. PBFT is used as consensus for both block insertion and transactions insertions (for all contexts).
2	1,000,000 transactions sent by 1,000 devices, varying from 1 to 10 contexts, where all gateways participate in all contexts. PBFT is used as consensus for block insertion and PoA is used as consensus for transactions insertions (for all contexts).

considered that all gateway has the authority to insert transactions. As a consequence, any gateway can insert any transaction. The transaction will be validated based on the usual the same criteria used in PBFT: valid information, hash and signatures and non-duplicate transactions.

6.5.1 Metrics

We used the same two metrics adopted in the evaluation presented in Section 6.4 to evaluate different consensus, *i.e.* **latency** and **throughput**. As used previously, latency was calculated based on the time spent from creating a transaction to inserting it in the blockchain. We considered as **throughput** the rate of insertion of transactions per second (tps) in the blockchain.

6.5.2 Results

We can observe in Table 6.15 the average (with the 95% confidence interval) transactions **latency** (in milliseconds) for the 2 scenarios. Hence, lower latency results are better. We present in the table only the results for 1, 2, 4 and 8 contexts to help the visualization of the data. For each scenario, we collect results from 1 to 10 contexts (for one context, all devices in that context; for two contexts, half of the devices in each context; and so on).

We can observe that when using only one context, the average latency is higher when compared to the usage of multiple contexts for both Scenarios 1 and 2. However, we can observe that in Scenario 2 (using PoA) the latency for 1 context (3467.5±6.3ms) has smaller difference to the usage of multiple contexts (1104.4±2.1ms in the best case). Also, we can observe that the difference in the performance in Scenario 1 and 2 is reduced when it is adopted 4 contexts. This smaller difference can be explained by the lower number of messages exchanged in the PoA algorithm. Consequently, when it adopts only one contexts, fewer message are exchanged and gateways are not overloaded with many processing/messaging when compared to PBFT (scenario 1).

Table 6.15: Latency (in milliseconds) to insert transaction using PBFT and PoA.

Scenario	Number of Contexts			
	1	2	4	8
1 - PBFT	287921.4±544.4	2402.2±5.4	1152.5±3.8	4106.3±8.3
2 - PoA	3467.5±6.3	1306.0±2.1	1104.4±2.1	1901.4±4.6

In order to help to better understand the differences between Scenarios 1 and 2, we present the results for 1 to 10 contexts in Fig. 6.11 using logarithmic curves. We can observe that, *Scenario 2* had better general results than *Scenario 1*. However, results are very similar when using 3 to 5 contexts, i.e., the best results for each contexts. This shows that context-based approach helps to improve the performance even when using a less strict solution, that requires less processing and message exchanging.

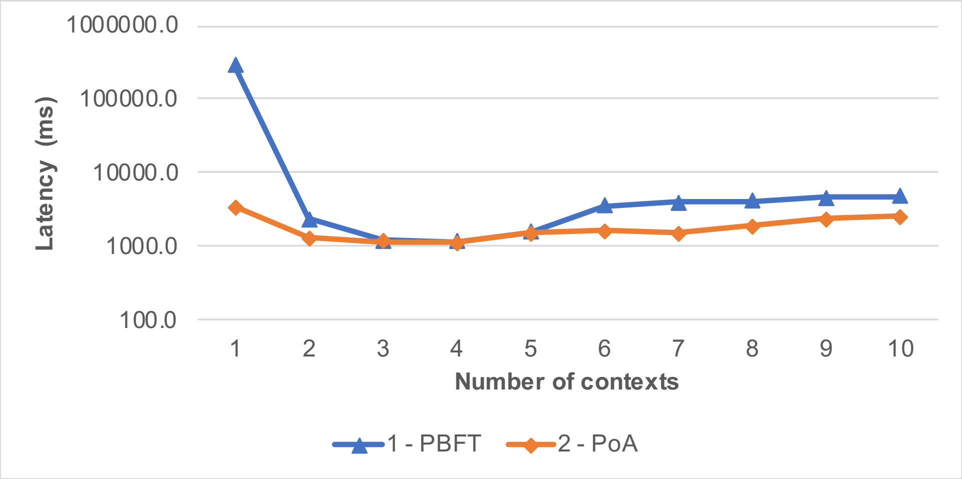


Figure 6.11: Average latency for each transaction in context-based consensus using PBFT and PoA.

As expected, when considering only one context, the **throughput** was considerably lower than with multiple contexts, as presented in Table 6.16. Similar to what happened to latency, best results were obtained when using PoA (*Scenario 2*). As a comparison, the best result was obtained when using four contexts, in *Scenario 2*, having a consensus throughput of **98.5±1.0tps**.

Fig. 6.12 shows the impact of the number of contexts and the usage of different consensus algorithms. We can observe that PoA have higher throughput even when using only one context. However, throughput is increased when using multiple contexts. This shows that parallelism of insertion using different contexts can help to improve appendable-block blockchains performance even when using a less strict consensus algorithms. Additionally, using multiple contexts, the throughput in Scenario 2 is above 80 transactions per second.

Table 6.16: Transactions throughput (transactions per second) using context-based consensus.

Scenarios	Number of Contexts			
	1	2	4	8
1 - PBFT	37.3±1.5	85.4±1.4	93±0.8	71.5±2.3
2 - PoA	77.9±1.8	94.5±1.1	98.5±1.0	89.1±0.8

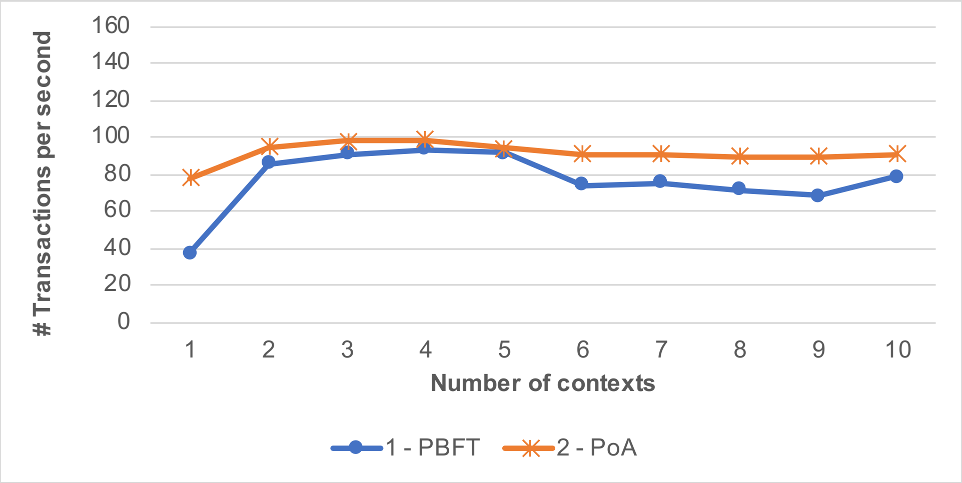


Figure 6.12: Average transactions consensus throughput (transactions per second) using PBFT and PoA.

This experiment shows that the context-based approach can present improvements both over a single context (or non-existence of contexts) and to the usage of a single consensus algorithms. This evaluation can help one to decide to use different consensus for each application or, even, different consensus for each context. This can help to provide a solution that fits better the needs of each IoT environment. However, there are some threats to validity of our evaluation since the evaluation tests were performed in a controlled environment.

The code is available at GitHub [6] and could be used to replicate the experiments.

6.6 Chapter Summary

The experimental evaluation presented in this chapter showed that the consensus can be performed with good results. In particular, we presented the use of different consensus algorithms in different scenarios. We also observed that the context-based approach can present improvements both over a single context (or non-existence of contexts) and to a single transaction insertion in the blockchain. Additionally, we presented the evaluation of

[6] https://github.com/conseg/speedychain/tree/speedychainApp

appendable block-blockchain using different consensus algorithms in each consensus level. This can help to provide a solution that fits in heterogeneous IoT environments.

Also, our evaluation shows that context-based consensus can guarantee lower latency and higher throughput. However, there are some threats to validity, limitations, security issues and new possible applications that we discuss in Chapter 7.

7. DISCUSSIONS

Multi-level consensus algorithms presented good performance results. However, the experiments performed in Chapter 6 can have some threats to validity and limitations. Moreover, some security issues can be present in some scenarios. Additionally, appendable-block blockchain can be adopted in different applications, not limited only for IoT environments. In the next sections we present some discussions on threats to validity, security issues, and applications of the presented solution.

7.1 Threats to Validity and Limitations

The different evaluations presented in this thesis were performed both in a controlled environment and in a cloud scenario. Thus, there are three main threats to validity of our evaluation. The first internal threat is the instrumentation used in the evaluation. The hardware used to perform the evaluation can impact on the obtained results, as well different hardware and network configurations lead to different results. For example, this can be observed comparing the results presented in Section 6.1 and Section 6.2. We can observe that the performance was slightly different using different hardware, but the relation between different scenarios were very similar. Also, in Section 6.3 the experiments were performed in a cloud environment with nodes in 4 geographically distributed nodes. Therefore, the results showed that geographically distributed environments can impact in the performance of the solution.

The second internal threat is related to the selection of values used to set the scenarios. A different number of gateways and devices, as well as different rate of transactions per second can influence the obtained results. Additionally, different gateway workloads are not explored in our solution. In our experiments, we evaluated the same number of devices (and transactions) connected to each gateway. Moreover, different types of transactions - such as smart contracts transaction on appendable-block blockchains (as proposed by Nunes *et al.* [251]) - can lead to different execution results. We intend to consider different hardware and different types of transactions in a future work.

A relevant limitation present in this work is that we performed the evaluation in emulated scenarios except for the evaluation performed in Section 6.2. This can mask real issues that can compromise our proposal. However, we did an extensive evaluation over different configurations to try to catch different issues that can affect the performance of our solution. Moreover, we adopted the same libraries and cryptography algorithms that were used in a previous evaluation with real hardware [216]. Consequently, we expect that IoT hardware can be capable to execute the same operations that were presented in this work.

Another important limitation is that we assumed that the information sent by devices in the secure channel are correct. The validations performed consists in verifying the signature, type of data and the timestamp. In the case that a device is compromised and send correct signature, type of data and timestamp, then our solution is not capable to detect a malicious device. Another aspect that we assume is that a device can connect to multiple gateways.

7.2 Security issues

In this section, we present a discussion about known attacks that could affect appendable-block blockchains and the evaluated consensus algorithms. Even though we mention different attacks, we focus on the main attacks that compromise the consensus layer, such as 51% Attack, Bribery Attack, Double Spending, Finney Attack, Eclipse Attack and Vector76 Attack. We briefly describe those attacks next.

Double Spending, Finney, Vector 76%, and Transaction Malleability attacks are aimed at spending coins in multiple transactions. In **Double Spending attack** [70], a malicious user sends multiple transactions to reachable peers in order to spend the same coin more than once. Alternatively, **Finney attack** [70] consists of a dishonest miner holding a pre-mined block, and spending the same coin that is used in a transaction of the pre-mined block. Combining these two attacks, **Vector 76% attack** [70] consists of requesting to withdraw the value of a transaction that was confirmed and sending the same value to another transaction, exploring the fork resolution algorithm (generating conflicts in the longest chain).

Many proposals that adopt blockchain in IoT scenarios [43, 85, 200, 216, 232] do not use cryptocurrencies. Consequently, Double spending, Finney, and Vector 76 and Transaction Malleability attacks are not attractive for malicious users. However, some of blockchains proposals for IoT support tokens for M2M (machine-to-machine) payments. Considering appendable-block blockchains, these attacks do not represent a threat, in particular when using dBFT and PBFT due to the voting procedure. In both consensus algorithms, sending multiple transactions with the same timestamp, signature, and information will be discarded in case of collision. Also, in case of incorrect order, the transaction will be discarded. In the case of PoW and witness-based approach, these attacks can be effective if a token structure is created to appendable-block blockchains. However, tokens were not introduced or discussed in appendable-block blockchains.

There are different attacks that explore vulnerabilities in the mining mechanism of PoW, such as 51%, Selfish Mining, Block-Withholding, Fork-After-Withholding, and Bribery attacks. The **51% attack** consists of a malicious user controlling more than 50% of network processing power, thus this user could rewrite the blockchain blocks and define the blockchain behaviour [119]. Similarly, **Selfish Mining** attack consists of a malicious user (or a pool) keeping own mined blocks private until its chain reach a length longer than the main blockchain. As per the fork rule, the attacker chain will now become the main chain [98]. **Block-Withholding** happens when a malicious miner - which is participating in a mining pool - finds a valid hash value and sends it directly to the blockchain network, thus avoiding division of the reward for mining the block [25]. Similarly, in **Fork-After-Withhold (FAW)** a malicious miner holds the block until another miner (from the same pool) identifies a block. Then, the malicious miner sends its block, forcing the pool to generate a fork (this block could be sent to multiple pools in order to increase its reward) [186]. **Bribery attack** [41] consists of a malicious user exploring the mining power of different nodes (through financial incentives) to include conflicting transactions in the blockchain (*e.g.*, can be used to force a Double Spending). **Sybil attack** relies on a malicious node assuming multiple identities in the network with the ultimate goal of influencing the network [89]. The **Eclipse attack** consists of a malicious user aiming to monopolise the incoming and outgoing connections of a victim, thus isolating the victim from the main blockchain network [137].

Selfish Mining, **51%**, **Block-Withholding**, **FAW** and **Bribery** attacks are based on strategies adopted by PoW consensus algorithms. Consequently, choosing a solution for IoT that uses a different consensus algorithm (*e.g.*, dBFT, PBFT, and witness-based approach) can help to avoid these kind of attacks. A key aspect to be considered is related to the hardware constraints in IoT devices, such as computing power, memory, and storage.

Biryukov *et al.* [34] present a **Deanonymization** technique where it is possible to identify users retrieving a list of Internet clients on different servers and linking them to transactions in the blockchain. However, appendable-block blockchain was proposed to be used in a private/consortium and permissioned environment. Consequently, the access to the information is managed by gateways. Consequently, this attack can be effective if a gateway is tampered to leak information maintained by the gateway.

Johnson *et al.* [158] presented that **Distributed Denial of Service (DDoS) attack** can be used to reduce the performance of a set of nodes in a blockchain, *e.g.*, mining capability in Bitcoin blockchain. This attack can be effective if a PoW consensus algorithm is adopted in appendable-block blockchain due to the high hardware demand. PBFT, dBFT and witness-based approaches can be affected by DDoS performed against the network. However, this kind of attack requires to that malicious users share the access to the network (which is controlled in a private/consortium environment).

Eclipse attacks occurs when a set of malicious nodes control the communication of a node to the rest of the blockchain network nodes [138]. This attack is effective in appendable-blockchain (in any consensus algorithm adopted), particularly due to the hierarchy of nodes. However, this problem was mitigated, as presented in Chapter 5, where each device can connect and send information to multiple gateways at the same time. This solution requires that a malicious entity (or a group of entities) tamper or take control of all gateways that a device is connected. In our experiments we considered that a device has a full connection to all gateways, *i.e.*, a device can connect to any gateway. Further discussions can be made by the number of minimum number of gateways that a device should connect to avoid this attack.

Also, context-based consensus can present some security issues, particularly due to the limited number of gateways controlling the consensus in each context. This issue is similar to the issues in the adoption of shards in blockchains [123]. Different from many shard approaches, all block headers are kept by all nodes in context-based consensus for appendable-block blockchains. This can reduce the impact of an attack (e.g., 1% attack), but further investigations will be discussed in a future work.

7.3 Promising applications

Appendable-block blockchains and the multi-consensus algorithm were evaluated in few IoT applications, such as Smart Buildings [216] [215], Smart Cities [232] [251], and Geo-distributed scenario [76]. However, our proposal can be useful for many other applications. In the next sections we present the main benefits and issues for different applications.

7.3.1 Supply Chain

Current supply chain solutions have to deal with diverse business processes and interactions involving many stakeholders, *e.g.*, primary producers, suppliers, retailers, government, etc. Consequently, the integration of supply chain and IoT can present many benefits. For example, this integration can enable stakeholders to collect data and monitor business processes in real time using sensors and devices to improve efficiency and transparency of the process. Recently, blockchain has been proposed as a promising solution to address the lack of visibility in supply chain data by providing a data recording and sharing platform with immutability, transparency, and traceability features [224].

Appendable-block blockchains can provide a platform for sharing supply chain data among the stakeholders. The integrity of the supply chain data is guaranteed by the blockchain structure. Furthermore, context-based smart contracts can be used for executing automated actions based on the data on context blocks. The parallelism capability of appendable-block blockchains improves the transaction throughput and latency for supply chains that generate large amounts of data. Additionally, the proposed multi-consensus algorithm can adapt the consensus mechanism, for example, to fit better in each context of a supply chain. For example, data from supplies can have a different consensus algorithm that a data produced by the machinery. Consequently, each context can have a best fit consensus algorithms.

7.3.2 Data Preservation

Data preservation is a process to manage and store data in a safe and integer manner [30]. Over the years, many solutions were developed, in particular to backup data. Currently, cloud-based solutions [341] is the most used approach. However, some distributed systems can have problems sharing and retrieving users' data. For example, many government systems do not provide an updated and global view of citizen data. Another common issue is the complete student curriculum, *i.e.*, every time that a student changes to another institution his data has to be somehow imported into a new system. Thus, blockchain can help in data preservation in distributed applications. Appendable-block blockchain helps to organize that data from the same user in the same block. This feature changes how the data is viewed in the blockchain systems. Therefore, this kind of blockchain is centered on the client information. In this way, all data is signed by the user. An advantage of this approach is that the same user can save its information in a non-conflicting way. Systems that validate and retrieve that information also can have a global and updated view from the user.

Appendable-block blockchain was designed for private/consortium architectures, where full-nodes manage who can access the information. Consequently, how the data are retrieved for different applications relies on the application-nodes. In this way, a system should be designed in a way that this application-nodes provide privacy for user data, and at the same time, deny access to that data when necessary. These features are not covered by current implementations of appendable-block blockchains. Lunardi *et al.* (2018) and Michelin *et al.* (2018) discussed the possibility of storing part of the data in cloud storage. This can introduce problems to the resilience of the solution, but it could help to reduce the size of the blockchain. Moreover, appendable-block blockchain can help to have an incremental data (and its history of changes). This feature can help to recover information from a specific period of time stored in a tamper-resistant and distributed way.

7.3.3 Healthcare

Healthcare is a complex environment that comprehends many entities and their relations. Healthcare systems have to manage data from patients, staff, hospital, clinics, drugs, and many regulations. This kind of environment is very complex and usually use different systems leading to interoperability problems and inconsistent information.

Electronic Health Records (EHR) - patients data - covers past patient diagnosis, laboratory tests, treatments, vaccination history, current and past diseases, etc. All data are highly relevant and critical to the patient health. However, in many systems, this information is not shared among different hospitals, clinics, and laboratories. Also, it is hard for a patient, by himself, to maintain all this information. Consequently, it is very important and desired that a solution can share all this information among different actors in the system (see figure 7.1). Thus, blockchain emerged as a possible solution to deal with issues in patient data management [231] [217] [194]. Appendable-block blockchain, in particular, can handle patients data in a single block for each patient.

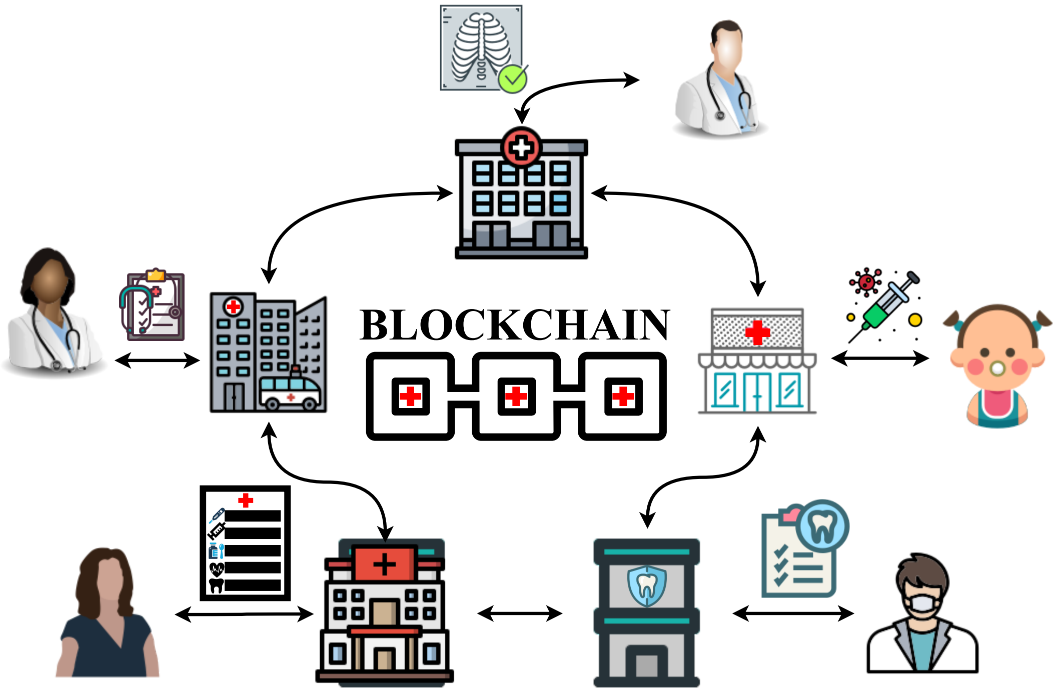


Figure 7.1: Healthcare integration (extracted from [213])

Also, every human resource (every staff from an institution) can be part of the system. Considering the adoption of smart contracts, the allocation of staff can be optimized by a system that monitors patients and sensors connected to them. A blockchain can be used both to store information and to execute optimization in the scheduling of treatments [73]. It is important to note that appendable-block blockchains can be used to both pure data blocks and to store/run smart contracts. Moreover, other information can be used for health care systems. For example, there is a lack of discussion about the integration of gym sensors, smart bands, and other fitness data that could be used in patients' health analysis. Appendable-block blockchains could be used as a solution to integrate different information about the users. For example, a single blockchain shared between different entities that will store and use information about the same user. Pharmacy and health insurance are important entities that could not be excluded from health care systems.

7.3.4 Other Applications

Citizen identity and other governmental systems are also well suited for appendable-block blockchains. In this kind of environment, the government can maintain the full-nodes (can be understood as gateways in our architecture) and control the access of the light-nodes (citizens), controlling the usage of identity in different domains, such as validation of virtual drive license to rent a car, passport and travel authorizations, usage of a health care system. Consequently, appendable-block blockchain has potential to be used to integrate different systems that require identification and needs to log historical data from citizens. Appendable-block blockchain can be suitable to many other applications such as education (students records), surveillance cameras, justice system (evidences, legal process, etc) and any other system that requires a shared control and between different entities and, at same time, needs a solution to ensure the tamper-resistant of the recorded information.

7.4 Out of Scope

In this section, we present some of the main aspects that are out of the scope of this thesis. This may be a relevant discussion for future works. Firstly, there are discussions about some features present in blockchain that we do no discuss in this work.

It is important to note that this thesis did not discuss issues present in the insertion of smart contracts transactions. Consequently, we do not discuss the impact of our proposed solution to problems such as the Verifier's Dilemma [8] and smart contracts correctness [225]. A model for smart contracts in appendable-block blockchain was presented by Nunes *et al.* [251]. Also, we do not propose a solution to perform payments using cryptocurrencies or any other kind of exchange of coins or tokens.

Considering IoT devices, we do not tackle any physical attack to the devices. This thesis focus on the discussion about the integrity of the data sent by IoT devices. We assume that the data produced by devices are not tampered. Physical and logical tempering (attack on chip, cracking, root access, and others attacks) are not tackled by our work.

8. FINAL CONSIDERATIONS

In this work we proposed and presented a multi-level consensus model for appendable-block blockchains. This model supports consensus at the block level and transaction level, as well as supporting the execution of different consensus for each context. We presented different experiments, showing the results of using different consensus algorithms at the block level in appendable-block blockchain. Even using scenarios composed by a million of transactions, the results presented an average to perform consensus lower than a second in emulated scenarios and few seconds in a geo-distributed scenario.

Furthermore, we evaluated a context-based consensus at the transaction level. Our solution can solve two existing issues in appendable-block blockchains, namely the Eclipse attack performed by a single malicious gateway and the lack of transactions consensus. Also, the evaluation achieved a total latency under 550ms and throughput above 100 transactions per second. The best results (latency under 550ms, achieving average results lower as 152.5m) were obtained using multiple contexts with a limited number of gateways and a limited number of transactions per consensus round. Although our proposed solution uses a blockchain with different architecture and data structure, the results presented were better than many commercial blockchain solution (such as Bitcoin and IOTA).

Next, we present the contributions made by this thesis, the answers to the research questions and directions to future work.

8.1 Thesis Contributions

In this thesis, we proposed a model for multi-level consensus algorithm that can consider different IoT contexts and applications, providing better relation among security and performance for IoT environments composed by different contexts. To fulfill our goals, we made the following contributions:

1. *A study about consensus algorithms for blockchain in IoT (Chapter 3).* We performed a study to identify the main advances made on consensus algorithms for blockchain in IoT. We could identify many works that are using a reputation system to be used in the consensus mechanism. Also we could observe that some of them are based on PoS solution. However, we observed that the majority of the approaches are based on variations of Byzantine fault tolerance. Besides, the works did not focused on how to adapt their solutions to different applications, problems and requirements. Consequently, different contexts or adaptive solutions are not supported.

2. *Improvement in appendable-block blockchain to support different consensus algorithms (Chapter 4).* To overcome the drawbacks of not adopting consensus algorithms on block level or the usage of a single consensus algorithms, we proposed an important improvement on appendable-block blockchain (and implemented on SpeedyChain) to allow the usage of four different consensus algorithms. This can help to use appendable-block blockchain on different scenarios.

3. *Proposal of multi-level consensus (Chapter 5.* To tackle many issues of not using consensus algorithms at transaction level, we proposed a model that can perform consensus for transactions on appendable-block blockchain. Also, this model supports the adoption of different consensus algorithms (different parameters/configurations) for each context. This allows a high level of parallelism on the transactions insertions, ensuring high throughput of transactions and low latency to insert transactions.

4. *Experimental evaluation of the contributions on different scenarios.* We evaluated consensus both on block level and transaction level on different scenarios. We conducted experiments mainly on emulated scenarios (except for the experiment performed on a cloud environment as presented on Section 6.3 due to the restrictions to access large amount of real IoT devices. We explored performance issues (or performed improvements) of using different consensus algorithms. Also, we tried to explore the advantage of appendable-block data structure to provide a high degree of parallelism on transactions insertions, improving the performance. The results obtained points out that it is possible to achieve high performance when we use multiple contexts (and multiple consensus) to insert transactions (achieving a reduction on latency - 1,000 times when comparing using 1 context and 2 contexts). Also, it was possible to improve the results reducing the number of nodes that participate in each context and when the nodes are updated by request. This shows that the possibility of adapting the consensus for each context can help to use appendable-block blockchains in different IoT scenarios.

8.2 Hypothesis Foundation

In this thesis, we presented evaluations using a prototype called SpeedyChain to show improvements obtained with the proposed model. All these efforts helped to validate the research hypothesis and to answer the research questions (presented in Section 1.3). The answers to each of our research question are discussed next.

1. *How existent consensus algorithms perform in different IoT scenarios?*

   We presented a discussion about the main research on consensus algorithms in IoT on Chapter 3 and also we conducted different experiments to evaluate different consensus algorithms on appendable-block blockchains in different IoT scenarios in Chapter 6, in particular Sections 6.1, 6.2, 6.3 and 6.5. We presented results for different consensus algorithms, such as PoW, PBFT and dBFT at the block level consensus and PBFT and PoA at the transaction level. These consensus algorithms can be used for different IoT scenarios. Each consensus algorithm presents its characteristics considering performance, resilience and scalability. For example, in general, voting-based (*e.g.*, PBFT) consensus algorithms present better performance results for private/permissioned blockchains with a limited number of nodes that perform the consensus.

2. *What are the security issues associated with each consensus algorithm that can impact IoT?*

   We presented a discussion about security issues on Section 7.2, where we presented the main impact of each attack/issue on appendable-block blockchain. Each consensus algorithm present specific attacks. In a nutshell, proof-based consensus (*e.g.*, PoW) are more susceptible to variations of double-spending and voting-based consensus (*e.g.*, PBFT) can present issues with scalability, and in appendable-block blockchains it can be susceptible to Eclipse attacks. In our context-based approach, we helped to tackle Eclipse attack issues in appendable-block blockchains.

3. *How multiple consensus algorithms can be performed to allow parallel insertion in the blockchain?*

   Due to the characteristics of the data structure of appendable-block blockchain, block insertion and transaction insertion can be performed in parallel. This approach can help to have different approaches/configurations to insert new blocks (and, in the case of appendable-block blockchains, new public key from nodes) and new transactions (data produced by devices). We presented the model for this insertion on Chapters 4 and 5, at same time that we presented evaluation on Chapter 6.

4. *How a consensus algorithm can be defined in order to be adapted for different contexts or to help in the interoperability among different applications?*

   We presented a solution to allow different consensus for different each context on Chapter 5. This allows to use different configurations and parameters on the same consensus or different consensus for each context. This can help to provide a solution that can fit different requirements of complex scenarios. Also, we evaluated the proposed solution on Chapter 6 showing that the proposed solution can improve the performance, reducing the latency to insert new transactions and increasing the throughput.

After answering the research questions we verified that our hypothesis "*It is possible to have a consensus algorithm that can handle different kind of block insertion, allowing* *interoperability among different blockchain applications and a better relation among performance and security for IoT environments*" is valid. The experiments demonstrated that we can achieve good performance when using our multi-level consensus. Furthermore, using different consensus and configurations leads to improvements on performance or resilience. Consequently, we could present a consensus mechanism that can handle different blockchain applications (through the usage of contexts) that can be adapted to have a better relation among performance and security for different IoT environments.

8.3 Directions for Future Work

As future work, we intend to scale our context-based consensus by increasing the number of gateways and devices, so that it can help to understand possible issues when using higher number of nodes. Also, the usage on real-world IoT environments can help to identify some possible requirements not identified on emulated scenarios.

We intend to model and implement the appendable-block blockchain in a blockchain simulator (*e.g.*, BlockSim[8]). The evaluation through simulation can help to increase the scale of the scenarios to evaluate the scalability of the solution. Also, using simulators can help to verify the impact of attacks on the solution. Additionally, further discussion on simulation can help to improve reliability and security of insertion of transactions.

Some improvements, considering different consensus algorithms for each context [214], for example a discussion should be performed considering different consensus algorithms (*e.g.*, BFT-SMART [32] and Roll-DPoS [101]). This can help the versatility of appendable-block blockchains.

Additionally, the evaluation of different cryptography algorithms such as Blake for hash [239], and Unbalanced Oil and Vinegar (UOV) for digital signatures scheme [262] can help to improve the performance of some basic operation in SpeedyChain or help to protect against quantum computers. These (and other) cryptography algorithms can affect the performance of consensus algorithms, in particular, Proof-of-Work based consensus mechanisms.

Finally we intend to evaluate the proposed model in an hostile environment to evaluate the effect of attacks, for example, to generate a detailed analysis and impact of attacks such as Sybil and 1% attack.

REFERENCES

[1] Afanasev, M. Y.; Krylova, A. A.; Shorokhov, S. A.; Fedosov, Y. V.; Sidorenko, A. S. “A Design of Cyber-physical Production System Prototype Based on an Ethereum Private Network”. In: Conference of Open Innovations Association, 2018, pp. 3–11.

[2] Agrawal, R.; Verma, P.; Sonanis, R.; Goel, U.; De, A.; Kondaveeti, S. A.; Shekhar, S. “Continuous Security in IoT Using Blockchain”. In: IEEE International Conference on Acoustics, Speech and Signal Processing, 2018, pp. 6423–6427.

[3] Ahmad, N. M.; Razak, S. F. A.; Kannan, S.; Yusof, I.; Amin, A. H. M. “Improving Identity Management of Cloud-Based IoT Applications Using Blockchain”. In: International Conference on Intelligent and Advanced System, 2018, pp. 1–6.

[4] Ahram, T.; Sargolzaei, A.; Sargolzaei, S.; Daniels, J.; Amaba, B. “Blockchain technology innovations”. In: IEEE Technology Engineering Management Conference, 2017, pp. 137–141.

[5] Ahrenholz, J.; Danilov, C.; Henderson, T. R.; Kim, J. H. “CORE: A real-time network emulator”. In: IEEE Military Communications Conference, 2008, pp. 1–7.

[6] Alessi, M.; Camillo, A.; Giangreco, E.; Matera, M.; Pino, S.; Storelli, D. “Make Users Own Their Data: A Decentralized Personal Data Store Prototype Based on Ethereum and IPFS”. In: International Conference on Smart and Sustainable Technologies, 2018, pp. 1–7.

[7] Alexopoulos, N.; Habib, S. M.; Mühlhäuser, M. “Towards Secure Distributed Trust Management on a Global Scale: An Analytical Approach for Applying Distributed Ledgers for Authorization in the IoT”. In: Workshop on IoT Security and Privacy, 2018, pp. 49–54.

[8] Alharby, M.; Castagna Lunardi, R.; Aldweesh, A.; van Moorsel, A. “Data-Driven Model-Based Analysis of the Ethereum Verifier’s Dilemma”. In: IEEE/IFIP International Conference on Dependable Systems and Networks, 2020, pp. 209–220.

[9] Ali, A. A.; El-Dessouky, I. A.; Abdallah, M. M.; Nabih, A. K. “The Quest for Fully Smart Autonomous Business Networks in IoT Platforms”. In: Africa and Middle East Conference on Software Engineering, 2017, pp. 13–18.

[10] Ali, M. S.; Dolui, K.; Antonelli, F. “IoT Data Privacy via Blockchains and IPFS”. In: International Conference on the Internet of Things, 2017, pp. 14:1–14:7.

[11] Ali, S.; Wang, G.; Bhuiyan, M. Z. A.; Jiang, H. “Secure Data Provenance in Cloud-Centric Internet of Things via Blockchain Smart Contracts”. In: IEEE

SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation, 2018, pp. 991–998.

[12] Alphand, O.; Amoretti, M.; Claeys, T.; Dall'Asta, S.; Duda, A.; Ferrari, G.; Rousseau, F.; Tourancheau, B.; Veltri, L.; Zanichelli, F. "IoTChain: A blockchain security architecture for the Internet of Things". In: Wireless Communications and Networking Conference, 2018, pp. 1–6.

[13] Álvarez Díaz, N.; Herrera-Joancomartí, J.; Caballero-Gil, P. "Smart Contracts Based on Blockchain for Logistics Management". In: International Conference on Internet of Things and Machine Learning, 2017, pp. 73:1–73:8.

[14] Andoni, M.; Robu, V.; Flynn, D.; Abram, S.; Geach, D.; Jenkins, D.; McCallum, P.; Peacock, A. "Blockchain technology in the energy sector: A systematic review of challenges and opportunities", *Renewable and Sustainable Energy Reviews*, vol. 100, Feb 2019, pp. 143 – 174.

[15] Angeletti, F.; Chatzigiannakis, I.; Vitaletti, A. "The role of blockchain and IoT in recruiting participants for digital clinical trials". In: International Conference on Software, Telecommunications and Computer Networks, 2017, pp. 1–5.

[16] Arenas, R.; Fernandez, P. "CredenceLedger: A Permissioned Blockchain for Verifiable Academic Credentials". In: IEEE International Conference on Engineering, Technology and Innovation, 2018, pp. 1–6.

[17] Arseni, S.; Mitoi, M.; Vulpe, A. "Pass-IoT: A platform for studying security, privacy and trust in IoT". In: International Conference on Communications, 2016, pp. 261–266.

[18] Arslan, H.; Aslan, H.; Karkı, H. D.; Yüksel, A. G. "Blockchain and Security in the IoT Environments: Literature Review". In: International Conference on Computer Science and Engineering, 2018, pp. 254–257.

[19] Asheralieva, A.; Niyato, D. "Reputation-Based Coalition Formation for Secure Self-Organized and Scalable Sharding in IoT Blockchains With Mobile-Edge Computing", *IEEE Internet of Things Journal*, vol. 7–12, Dec 2020, pp. 11830–11850.

[20] Aung, Y. N.; Tantidham, T. "Review of Ethereum: Smart home case study". In: International Conference on Information Technology, 2017, pp. 1–4.

[21] Avizienis, A.; Laprie, J. .; Randell, B.; Landwehr, C. "Basic concepts and taxonomy of dependable and secure computing", *IEEE Transactions on Dependable and Secure Computing*, vol. 1–1, Jan 2004, pp. 11–33.

[22] Awasthi, S.; Johri, P.; Khatri, S. K. “IoT based Security Model to Enhance Blockchain Technology”. In: International Conference on Advances in Computing and Communication Engineering, 2018, pp. 133–137.

[23] Ayoade, G.; Karande, V.; Khan, L.; Hamlen, K. “Decentralized IoT Data Management Using BlockChain and Trusted Execution Environment”. In: IEEE International Conference on Information Reuse and Integration, 2018, pp. 15–22.

[24] Badr, S.; Gomaa, I.; Abd-Elrahman, E. “Multi-tier Blockchain Framework for IoT-EHRs Systems”, *Procedia Computer Science*, vol. 141, Sep 2018, pp. 159 – 166.

[25] Bag, S.; Ruj, S.; Sakurai, K. “Bitcoin Block Withholding Attack: Analysis and Mitigation”, *IEEE Transactions on Information Forensics and Security*, vol. 12–8, Aug 2017, pp. 1967–1978.

[26] Bai, H.; Xia, G.; Fu, S. “A Two-Layer-Consensus Based Blockchain Architecture for IoT”. In: IEEE International Conference on Electronics Information and Emergency Communication, 2019, pp. 1–6.

[27] Baliga, A.; Subhod, I.; Kamat, P.; Chatterjee, S. “Performance Evaluation of the Quorum Blockchain Platform”. Source: http://arxiv.org/abs/1809.03421, 26 feb 2019.

[28] Banerjee, M.; Lee, J.; Chen, Q.; Choo, K. R. “Blockchain-Based Security Layer for Identification and Isolation of Malicious Things in IoT: A Conceptual Design”. In: International Conference on Computer Communication and Networks, 2018, pp. 1–6.

[29] Banerjee, M.; Lee, J.; Choo, K.-K. R. “A blockchain future for internet of things security: a position paper”, *Digital Communications and Networks*, vol. 4–3, Aug 2018, pp. 149 – 160.

[30] Berman, F. “Got data? a guide to data preservation in the information age”, *Communications of the ACM*, vol. 51–12, Dec 2008, pp. 50–56.

[31] Bertoglio, D. D.; Girotto, G.; Neu, C. N.; Lunardi, R. C. “Pentest on an Internet Mobile App: A Case Studyusing Tramonto”. In: International Conference for Internet Technology and Secured Transactions, 2019, pp. 1–6.

[32] Bessani, A.; Sousa, J.; Alchieri, E. E. P. “State Machine Replication for the Masses with BFT-SMART”. In: IEEE/IFIP International Conference on Dependable Systems and Networks, 2014, pp. 355–362.

[33] Bezahaf, M.; Cathelain, G.; Ducrocq, T. “BcWAN: A Federated Low-Power WAN for the Internet of Things (Industry Track)”. In: International Middleware Conference Industry, 2018, pp. 54–60.

[34] Biryukov, A.; Khovratovich, D.; Pustogarov, I. “Deanonymisation of Clients in Bitcoin P2P Network”. In: ACM SIGSAC Conference on Computer and Communications Security, 2014, pp. 15–29.

[35] Biswas, K.; Muthukkumarasamy, V. “Securing Smart Cities Using Blockchain Technology”. In: IEEE International Conference on High Performance Computing and Communications; IEEE International Conference on Smart City; IEEE International Conference on Data Science and Systems, 2016, pp. 1392–1393.

[36] Biswas, S.; Sharif, K.; Li, F.; Maharjan, S.; Mohanty, S. P.; Wang, Y. “PoBT: A Lightweight Consensus Algorithm for Scalable IoT Business Blockchain”, *IEEE Internet of Things Journal*, vol. 7–3, Mar 2020, pp. 2343–2355.

[37] BITONODES. “Global bitcoin nodes distribution”. Source: https://bitnodes.earn.com/, 26 feb 2019.

[38] Blockchain. “Confirmed transactions per day in bitcoin”. Source: https://www.blockchain.com/charts/n-transactions, 26 jan 2021.

[39] Blockchain. “Network hash difficulty in bitcoin”. Source: https://www.blockchain.com/charts/difficulty, 15 jan 2021.

[40] Bocek, T.; Rodrigues, B. B.; Strasser, T.; Stiller, B. “Blockchains everywhere - a use-case of blockchains in the pharma supply-chain”. In: IFIP/IEEE Symposium on Integrated Network and Service Management, 2017, pp. 772–777.

[41] Bonneau, J. “Why buy when you can rent?” In: International Conference on Financial Cryptography and Data Security, 2016, pp. 19–26.

[42] Bottone, M.; Raimondi, F.; Primiero, G. “Multi-agent Based Simulations of Block-Free Distributed Ledgers”. In: International Conference on Advanced Information Networking and Applications Workshops, 2018, pp. 585–590.

[43] Boudguiga, A.; Bouzerna, N.; Granboulan, L.; Olivereau, A.; Quesnel, F.; Roger, A.; Sirdey, R. “Towards Better Availability and Accountability for IoT Updates by Means of a Blockchain”. In: IEEE European Symposium on Security and Privacy Workshops, 2017, pp. 50–58.

[44] Branco, V.; Lippert, B.; Lunardi, R.; Nunes, H.; Neu, C.; Zorzo, A.; Pirolla, D.; Bernucio, R.; Spacov, S. “Modelo de negócio para saúde colaborativa usando smart contracts: caso TokenHealth”, *Revista Brasileira de Computação Aplicada*, vol. 12–1, Apr 2020, pp. 134–144.

[45] Branco, V.; Lippert, B.; Nunes, H.; Lunardi, R.; Zorzo, A. “Avaliação do uso de smart contracts para sistema de saúde colaborativa”. In: Escola Regional de Redes de Computadores, 2019, pp. 9–16.

[46] Bruneo, D.; Chillari, S.; Distefano, S.; Giacobbe, M.; Minnolo, A. L.; Longo, F.; Merlino, G.; Mulfari, D.; Panarello, A.; Patanè, G.; Puliafito, A.; Puliafito, C.; Scarpa, M.; Tapas, N.; Visalli, G. “Building a Smart City Service Platform in Messina with the #SmartME Project”. In: International Conference on Advanced Information Networking and Applications Workshops, 2018, pp. 343–348.

[47] Bruneo, D.; Distefano, S.; Giacobbe, M.; Minnolo, A. L.; Longo, F.; Merlino, G.; Mulfari, D.; Panarello, A.; Patanè, G.; Puliafito, A.; Puliafito, C.; Tapas, N. “An IoT service ecosystem for Smart Cities: The #SmartME project”, *Internet of Things*, vol. 5, Mar 2019, pp. 12 – 33.

[48] Buccafurri, F.; Lax, G.; Nicolazzo, S.; Nocera, A. “Overcoming Limits of Blockchain for IoT Applications”. In: International Conference on Availability, Reliability and Security, 2017, pp. 26:1–26:6.

[49] Cachin, C. “Architecture of the Hyperledger blockchain fabric”. In: Workshop on Distributed Cryptocurrencies and Consensus Ledgers, 2016, pp. 1–4.

[50] Cao, B.; Li, Y.; Zhang, L.; Zhang, L.; Mumtaz, S.; Zhou, Z.; Peng, M. “When Internet of Things Meets Blockchain: Challenges in Distributed Consensus”, *IEEE Network*, vol. 33–6, Dec 2019, pp. 133–139.

[51] Caro, M. P.; Ali, M. S.; Vecchio, M.; Giaffreda, R. “Blockchain-based traceability in Agri-Food supply chain management: A practical implementation”. In: IoT Vertical and Topical Summit on Agriculture - Tuscany, 2018, pp. 1–4.

[52] Casado-Vara, R.; de la Prieta, F.; Prieto, J.; Corchado, J. M. “Blockchain Framework for IoT Data Quality via Edge Computing”. In: Workshop on Blockchain-enabled Networked Sensor Systems, 2018, pp. 19–24.

[53] Castro, M.; Liskov, B. “Practical Byzantine Fault Tolerance”. In: Symposium on Operating Systems Design and Implementation, 1999, pp. 173–186.

[54] Cebe, M.; Kaplan, B.; Akkaya, K. “A Network Coding Based Information Spreading Approach for Permissioned Blockchain in IoT Settings”. In: EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2018, pp. 470–475.

[55] Cha, S.; Chen, J.; Su, C.; Yeh, K. “A Blockchain Connected Gateway for BLE-Based Devices in the Internet of Things”, *IEEE Access*, vol. 6, Dec 2018, pp. 24639–24649.

[56] Cha, S.; Tsai, T.; Peng, W.; Huang, T.; Hsu, T. “Privacy-aware and blockchain connected gateways for users to access legacy IoT devices”. In: IEEE Global Conference on Consumer Electronics, 2017, pp. 1–3.

[57] Chai, H.; Leng, S.; Zhang, K.; Mao, S. “Proof-of-Reputation Based-Consortium Blockchain for Trust Resource Sharing in Internet of Vehicles”, *IEEE Access*, vol. 7, Dec 2019, pp. 175744–175757.

[58] Chakraborty, P.; Shahriyar, R.; Iqbal, A.; Bosu, A. “Understanding the Software Development Practices of Blockchain Projects: A Survey”. In: ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2018, pp. 28:1–28:10.

[59] Chakraborty, R. B.; Pandey, M.; Rautaray, S. S. “Managing Computation Load on a Blockchain – based Multi – Layered Internet – of – Things Network”, *Procedia Computer Science*, vol. 132, Sep 2018, pp. 469 – 476.

[60] Chalaemwongwan, N.; Kurutach, W. “State of the art and challenges facing consensus protocols on blockchain”. In: International Conference on Information Networking, 2018, pp. 957–962.

[61] Chaudhary, R.; Aujla, G. S.; Garg, S.; Kumar, N.; Rodrigues, J. J. P. C. “SDN-Enabled Multi-Attribute-Based Secure Communication for Smart Grid in IIoT Environment”, *IEEE Transactions on Industrial Informatics*, vol. 14, Jun 2018, pp. 2629–2640.

[62] Chen, J. “Hybrid blockchain and pseudonymous authentication for secure and trusted iot networks”, *Special Interest Group on Embedded Systems Review*, vol. 15–5, Nov.

[63] Chen, W.; Ma, M.; Ye, Y.; Zheng, Z.; Zhou, Y. “IoT Service Based on JointCloud Blockchain: The Case Study of Smart Traveling”. In: Symposium on Service-Oriented System Engineering, 2018, pp. 216–221.

[64] Choi, S. S.; Burm, J. W.; Sung, W.; Jang, J. W.; Reo, Y. J. “A Blockchain-based Secure IoT Control Scheme”. In: International Conference on Advances in Computing and Communication Engineering, 2018, pp. 74–78.

[65] Christidis, K.; Devetsikiotis, M. “Blockchains and Smart Contracts for the Internet of Things”, *IEEE Access*, vol. 4, May 2016, pp. 2292–2303.

[66] Cirani, S.; Davoli, L.; Ferrari, G.; Léone, R.; Medagliani, P.; Picone, M.; Veltri, L. “A Scalable and Self-Configuring Architecture for Service Discovery in the Internet of Things”, *IEEE Internet of Things Journal*, vol. 1–5, Oct 2014, pp. 508–521.

[67] CoinMarketCap. “Top 100 cryptocurrencies by market capitalization”. Source: https://coinmarketcap.com/, 26 feb 2019.

[68] Conoscenti, M.; Vetro, A.; Martin, J. C. D. “Blockchain for the Internet of Things: A systematic literature review”. In: IEEE/ACS International Conference of Computer Systems and Applications, 2016, pp. 1–6.

[69] Conoscenti, M.; Vetrò, A.; Martin, J. C. D. “Peer to Peer for Privacy and Decentralization in the Internet of Things”. In: IEEE/ACM International Conference on Software Engineering Companion, 2017, pp. 288–290.

[70] Conti, M.; Sandeep Kumar, E.; Lal, C.; Ruj, S. “A survey on security and privacy issues of bitcoin”, *IEEE Communications Surveys Tutorials*, vol. 20–4, May 2018, pp. 3416–3452.

[71] Crain, T.; Gramoli, V.; Larrea, M.; Raynal, M. “Leader/Randomization/Signature)-free Byzantine Consensus for Consortium Blockchains”. Source: http://arxiv.org/abs/1702.03068, 26 feb 2019.

[72] Danzi, P.; Kalor, A. E.; Stefanovic, C.; Popovski, P. “Analysis of the Communication Traffic for Blockchain Synchronization of IoT Devices”. In: IEEE International Conference on Communications, 2018, pp. 1–7.

[73] Dasaklis, T. K.; Casino, F.; Patsakis, C. “Blockchain Meets Smart Health: Towards Next Generation Healthcare Services”. In: International Conference on Information, Intelligence, Systems and Applications, 2018, pp. 1–8.

[74] Dasu, T.; Kanza, Y.; Srivastava, D. “Geofences in the sky: Herding drones with blockchains and 5g”. In: ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2018, pp. 73–76.

[75] Daza, V.; Pietro, R. D.; Klimek, I.; Signorini, M. “CONNECT: CONtextual NamE disCovery for blockchain-based services in the IoT”. In: IEEE International Conference on Communications, 2017, pp. 1–6.

[76] de Arruda, E. H. P.; Lunardi, R. C.; Nunes, H. C.; Zorzo, A. F.; Michelin, R. A. “Appendable-block Blockchain Evaluation over Geographically-Distributed IoT Networks”. In: IEEE International Black Sea Conference on Communications and Networking, 2020, pp. 1–6.

[77] De Biase, L. C. C.; Calcina-Ccori, P. C.; Fedrecheski, G.; Duarte, G. M.; Rangel, P. S. S.; Zuffo, M. K. “Swarm Economy: A Model for Transactions in a Distributed and Organic IoT Platform”, *IEEE Internet of Things Journal*, vol. 6–3, Jun 2019, pp. 4561–4572.

[78] de Jong, I. “Pyro - Python Remote Objects - 4.60 Documentation”. Source: https://pythonhosted.org/Pyro4, 26 feb 2020.

[79] Dedeoglu, V.; Dorri, A.; Jurdak, R.; Michelin, R. A.; Lunardi, R. C.; Kanhere, S. S.; Zorzo, A. F. “A Journey in Applying Blockchain for Cyberphysical Systems”. In: International Conference on COMmunication Systems NETworkS, 2020, pp. 383–390.

[80] Dedeoglu, V.; Jurdak, R.; Dorri, A.; Lunardi, R. C.; Michelin, R. A.; Zorzo, A. F.; Kanhere, S. S. “Blockchain Technologies for IoT”. Springer Singapore, 2020, chap. 3, pp. 55–89.

[81] Destefanis, G.; Marchesi, M.; Ortu, M.; Tonelli, R.; Bracciali, A.; Hierons, R. “Smart contracts vulnerabilities: a call for blockchain software engineering?” In: International Workshop on Blockchain Oriented Software Engineering, 2018, pp. 19–25.

[82] Dey, T.; Jaiswal, S.; Sunderkrishnan, S.; Katre, N. “HealthSense: A medical use case of Internet of Things and blockchain”. In: International Conference on Intelligent Sustainable Systems, 2017, pp. 486–491.

[83] Di Pietro, R.; Salleras, X.; Signorini, M.; Waisbard, E. “A Blockchain-based Trust System for the Internet of Things”. In: ACM Symposium on Access Control Models and Technologies, 2018, pp. 77–83.

[84] Ding, A. Y.; Janssen, M. “Opportunities for applications using 5g networks: Requirements, challenges, and outlook”. In: International Conference on Telecommunications and Remote Sensing, 2018, pp. 27–34.

[85] Dorri, A.; Kanhere, S. S.; Jurdak, R. “Towards an Optimized BlockChain for IoT”. In: International Conference on Internet-of-Things Design and Implementation, 2017, pp. 173–178.

[86] Dorri, A.; Kanhere, S. S.; Jurdak, R. “MOF-BC: A memory optimized and flexible blockchain for large scale networks”, *Future Generation Computer Systems*, vol. 92, Mar 2019, pp. 357 – 373.

[87] Dorri, A.; Kanhere, S. S.; Jurdak, R.; Gauravaram, P. “Blockchain for IoT security and privacy: The case study of a smart home”. In: IEEE International Conference on Pervasive Computing and Communications Workshops, 2017, pp. 618–623.

[88] Dorri, A.; Steger, M.; Kanhere, S. S.; Jurdak, R. “BlockChain: A Distributed Solution to Automotive Security and Privacy”, *IEEE Communications Magazine*, vol. 55–12, Dec 2017, pp. 119–125.

[89] Douceur, J. R. “The Sybil Attack”. In: International Workshop on Peer-to-Peer Systems, 2002, pp. 251–260.

[90] Duan, X.; Yan, Z.; Geng, G.; Yan, B. “DNSLedger: Decentralized and distributed name resolution for ubiquitous IoT”. In: IEEE International Conference on Consumer Electronics, 2018, pp. 1–3.

[91] Dukkipati, C.; Zhang, Y.; Cheng, L. C. “Decentralized, BlockChain Based Access Control Framework for the Heterogeneous Internet of Things”. In: ACM Workshop on Attribute-Based Access Control, 2018, pp. 61–69.

[92] Durand, A.; Gremaud, P.; Pasquier, J. “Resilient, Crowd-sourced LPWAN Infrastructure Using Blockchain”. In: Workshop on Cryptocurrencies and Blockchains for Distributed Systems, 2018, pp. 25–29.

[93] Dwork, C.; Naor, M. “Pricing via processing or combatting junk mail”. In: Advances in Cryptology, 1993, pp. 139–147.

[94] Dziembowski, S.; Faust, S.; Kolmogorov, V.; Pietrzak, K. “Proofs of Space”. In: Advances in Cryptology, 2015, pp. 585–605.

[95] Ellul, J.; Pace, G. J. “AlkylVM: A Virtual Machine for Smart Contract Blockchain Connected Internet of Things”. In: IFIP International Conference on New Technologies, Mobility and Security, 2018, pp. 1–4.

[96] Elsts, A.; Mitskas, E.; Oikonomou, G. “Distributed Ledger Technology and the Internet of Things: A Feasibility Study”. In: Workshop on Blockchain-enabled Networked Sensor Systems, 2018, pp. 7–12.

[97] Esposito, C.; Palmieri, F.; Choo, K. R. “Cloud Message Queueing and Notification: Challenges and Opportunities”, *IEEE Cloud Computing*, vol. 5–2, Mar 2018, pp. 11–16.

[98] Eyal, I.; Sirer, E. G. “Majority Is Not Enough: Bitcoin Mining Is Vulnerable”. In: International Conference on Financial Cryptography and Data Security, 2014, pp. 436–454.

[99] Fabiano, N. “Internet of Things and Blockchain: Legal Issues and Privacy. The Challenge for a Privacy Standard”. In: IEEE International Conference on Internet of Things and IEEE Green Computing and Communications and IEEE Cyber, Physical and Social Computing and IEEE Smart Data, 2017, pp. 727–734.

[100] Fabiano, N. “The Internet of Things ecosystem: The blockchain and privacy issues. The challenge for a global privacy standard”. In: International Conference on Internet of Things for the Global Community, 2017, pp. 1–7.

[101] Fan, X.; Chai, Q. “Roll-DPoS: A Randomized Delegated Proof of Stake Scheme for Scalable Blockchain-Based Internet of Things Systems”. In: EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2018, pp. 482–484.

[102] Fayad, A.; Hammi, B.; Khatoun, R. “An adaptive authentication and authorization scheme for IoT’s gateways: a blockchain based approach”. In: International Conference on Security of Smart Cities, Industrial Control System and Communications, 2018, pp. 1–7.

[103] Feng, L.; Zhang, H.; Lou, L.; Chen, Y. “A Blockchain-Based Collocation Storage Architecture for Data Security Process Platform of WSN”. In: IEEE International Conference on Computer Supported Cooperative Work in Design, 2018, pp. 75–80.

[104] Feng, Q.; He, D.; Zeadally, S.; Khan, M. K.; Kumar, N. “A survey on privacy protection in blockchain system”, *Journal of Network and Computer Applications*, vol. 126, Jan 2019, pp. 45 – 58.

[105] Fernández-Caramés, T. M.; Fraga-Lamas, P. “A Review on the Use of Blockchain for the Internet of Things”, *IEEE Access*, vol. 6, Dec 2018, pp. 32979–33001.

[106] Fezeu, H. K.; Djotio, T.; Thami, R. O. H. “Safe and Irrefutable Decentralized Communication: Bringing Non-Repudiation to Mesh Networks”. In: International Conference on Big Data, Cloud and Applications, 2017, pp. 37:1–37:6.

[107] Florea, B. C. “Blockchain and Internet of Things data provider for smart applications”. In: Mediterranean Conference on Embedded Computing, 2018, pp. 1–4.

[108] Fotiou, N.; Polyzos, G. C. “Smart Contracts for the Internet of Things: Opportunities and Challenges”. In: European Conference on Networks and Communications, 2018, pp. 256–260.

[109] Foundation, E. “Ethereum Specification”. Source: https://github.com/ethereum/go-ethereum/wiki/Ethereum-Specification, 26 feb 2019.

[110] Foundation, E. “Ethereum White Paper”. Source: https://github.com/ethereum/wiki/wiki/White-Paper, 26 feb 2019.

[111] Foundation, I. “IOTA - Next Generation Blockchain”. Source: https://iota.org/, 26 feb 2019.

[112] Foundation, L. “Hyperledger architecture”. Source: https://hyperledger-fabric.readthedocs.io/en/release-1.2/, 26 feb 2019.

[113] Fournier, G.; Petrillo, F. “Challenges and solutions on architecting blockchain systems”. In: International Conference on Computer Science and Software Engineering, 2018, pp. 293–300.

[114] Gallo, P.; Pongnumkul, S.; Nguyen, U. Q. “BlockSee: Blockchain for IoT Video Surveillance in Smart Cities”. In: International Conference on Environment and Electrical Engineering, 2018, pp. 1–6.

[115] Gao, J.; Asamoah, K. O.; Sifah, E. B.; Smahi, A.; Xia, Q.; Xia, H.; Zhang, X.; Dong, G. “GridMonitoring: Secured Sovereign Blockchain Based Monitoring on Smart Grid”, *IEEE Access*, vol. 6, Feb 2018, pp. 9917–9925.

[116] Gao, W.; Hatcher, W. G.; Yu, W. “A Survey of Blockchain: Techniques, Applications, and Challenges”. In: International Conference on Computer Communication and Networks, 2018, pp. 1–11.

[117] Garamvölgyi, P.; Kocsis, I.; Gehl, B.; Klenik, A. “Towards model-driven engineering of smart contracts for cyber-physical systems”. In: IEEE/IFIP International Conference on Dependable Systems and Networks Workshops, 2018, pp. 134–139.

[118] Gattolin, A.; Rottondi, C.; Verticale, G. “BlAsT: Blockchain-Assisted Key Transparency for Device Authentication”. In: IEEE International Forum on Research and Technology for Society and Industry, 2018, pp. 1–6.

[119] Gervais, A.; Karame, G. O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; Capkun, S. “On the Security and Performance of Proof of Work Blockchains”. In: ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 3–16.

[120] Grabatin, M.; Hommel, W. “Reliability and scalability improvements to identity federations by managing SAML metadata with distributed ledger technology”. In: IEEE/IFIP Network Operations and Management Symposium, 2018, pp. 1–6.

[121] Gries, S.; Meyer, O.; Wessling, F.; Hesenius, M.; Gruhn, V. “Using Blockchain Technology to Ensure Trustful Information Flow Monitoring in CPS”. In: IEEE International Conference on Software Architecture Companion, 2018, pp. 35–38.

[122] Gupta, Y.; Shorey, R.; Kulkarni, D.; Tew, J. “The applicability of blockchain in the internet of things”. In: International Conference on Communication Systems Networks, 2018, pp. 561–564.

[123] Hafid, A.; Hafid, A. S.; Samih, M. “New Mathematical Model to Analyze Security of Sharding-Based Blockchain Protocols”, *IEEE Access*, vol. 7, Dec 2019, pp. 185447–185457.

[124] Hahn, A.; Singh, R.; Liu, C.; Chen, S. “Smart contract-based campus demonstration of decentralized transactive energy auctions”. In: IEEE Power Energy Society Innovative Smart Grid Technologies Conference, 2017, pp. 1–5.

[125] Hammi, M. T.; Bellot, P.; Serhrouchni, A. “BCTrust: A decentralized authentication blockchain-based mechanism”. In: IEEE Wireless Communications and Networking Conference, 2018, pp. 1–6.

[126] Hammi, M. T.; Hammi, B.; Bellot, P.; Serhrouchni, A. “Bubbles of Trust: A decentralized blockchain-based authentication system for IoT”, *Computers & Security*, vol. 78, Sep 2018, pp. 126 – 142.

[127] Han, D.; Kim, H.; Jang, J. “Blockchain based smart door lock system”. In: International Conference on Information and Communication Technology Convergence, 2017, pp. 1165–1167.

[128] Han, R.; Gramoli, V.; Xu, X. “Evaluating Blockchains for IoT”. In: IFIP International Conference on New Technologies, Mobility and Security, 2018, pp. 1–5.

[129] Handschuh, H. “SHA Family (Secure Hash Algorithm)”. Springer US, 2005, chap. 10, pp. 565–567.

[130] Hao, X.; Xiao-Hong, S.; Dian, Y. “Multi-Agent System for E-commerce Security Transaction with Block Chain Technology”. In: International Symposium in Sensing and Instrumentation in IoT Era, 2018, pp. 1–6.

[131] Hao, Z.; Ji, R.; Li, Q. “FastPay: A Secure Fast Payment Method for Edge-IoT Platforms using Blockchain”. In: IEEE/ACM Symposium on Edge Computing, 2018, pp. 410–415.

[132] Hardjono, T.; Smith, N. “Cloud-Based Commissioning of Constrained Devices Using Permissioned Blockchains”. In: ACM International Workshop on IoT Privacy, Trust, and Security, 2016, pp. 29–36.

[133] Hasan, M. G. M. M.; Datta, A.; Rahman, M. A. “Poster abstract: Chained of things: A secure and dependable design of autonomous vehicle services”. In: IEEE/ACM International Conference on Internet-of-Things Design and Implementation, 2018, pp. 298–299.

[134] Hashemi, S. H.; Faghri, F.; Rausch, P.; Campbell, R. H. “World of Empowered IoT Users”. In: IEEE International Conference on Internet-of-Things Design and Implementation, 2016, pp. 13–24.

[135] He, Q.; Guan, N.; Lv, M.; Yi, W. “On the Consensus Mechanisms of Blockchain/DLT for Internet of Things”. In: IEEE International Symposium on Industrial Embedded Systems, 2018, pp. 1–10.

[136] He, S.; Tang, Q.; Wu, C. Q. “Censorship Resistant Decentralized IoT Management Systems”. In: EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2018, pp. 454–459.

[137] Heilman, E.; Kendler, A.; Zohar, A.; Goldberg, S. “Eclipse Attacks on Bitcoin’s Peer-to-Peer Network”. In: USENIX Conference on Security Symposium, 2015, pp. 129–144.

[138] Henningsen, S.; Teunis, D.; Florian, M.; Scheuermann, B. “Eclipsing Ethereum Peers with False Friends”. In: IEEE European Symposium on Security and Privacy Workshops, 2019, pp. 300–309.

[139] Hinckeldeyn, J.; Jochen, K. “(Short Paper) Developing a Smart Storage Container for a Blockchain-Based Supply Chain Application”. In: Crypto Valley Conference on Blockchain Technology, 2018, pp. 97–100.

[140] Hossain, M.; Karim, Y.; Hasan, R. “FIF-IoT: A Forensic Investigation Framework for IoT Using a Public Digital Ledger”. In: IEEE International Congress on Internet of Things, 2018, pp. 33–40.

[141] Huang, J.; Kong, L.; Chen, G.; Wu, M.; Liu, X.; Zeng, P. “Towards Secure Industrial IoT: Blockchain System With Credit-Based Consensus Mechanism”, *IEEE Transactions on Industrial Informatics*, vol. 15–6, Jun 2019, pp. 3680–3689.

[142] Huang, Z.; Su, X.; Zhang, Y.; Shi, C.; Zhang, H.; Xie, L. “A decentralized solution for IoT data trusted exchange based-on blockchain”. In: IEEE International Conference on Computer and Communications, 2017, pp. 1180–1184.

[143] Huckle, S.; Bhattacharya, R.; White, M.; Beloff, N. “Internet of things, blockchain and shared economy applications”, *Procedia Computer Science*, vol. 98, Dec 2016, pp. 461 – 466.

[144] Hudaya, A.; Amin, M.; Ahmad, N. M.; Kannan, S. “Integrating Distributed Pattern Recognition Technique for Event Monitoring within the IoT-Blockchain Network”. In: International Conference on Intelligent and Advanced System, 2018, pp. 1–6.

[145] Huh, S.; Cho, S.; Kim, S. “Managing IoT devices using blockchain platform”. In: International Conference on Advanced Communication Technology, 2017, pp. 464–467.

[146] Hwang, D.; Choi, J.; Kim, K. “Dynamic Access Control Scheme for IoT Devices using Blockchain”. In: International Conference on Information and Communication Technology Convergence, 2018, pp. 713–715.

[147] Hwang, J.; in Choi, M.; Lee, T.; Jeon, S.; Kim, S.; Park, S.; Park, S. “Energy Prosumer Business Model Using Blockchain System to Ensure Transparency and Safety”, *Energy Procedia*, vol. 141, Dec 2017, pp. 194 – 198.

[148] Ibba, S.; Pinna, A.; Seu, M.; Pani, F. E. “CitySense: Blockchain-oriented Smart Cities”. In: International Conference on Agile Software Development Scientific Workshops, 2017, pp. 12:1–12:5.

[149] Imai, S.; Varela, C. A.; Patterson, S. “A Performance Study of Geo-Distributed IoT Data Aggregation for Fog Computing”. In: IEEE/ACM International Conference on Utility and Cloud Computing Companion, 2018, pp. 278–283.

[150] Ioini, N. E.; Pahl, C. “Trustworthy orchestration of container based edge computing using permissioned blockchain”. In: International Conference on Internet of Things: Systems, Management and Security, 2018, pp. 147–154.

[151] Isaja, M.; Soldatos, J. “Distributed ledger technology for decentralization of manufacturing processes”. In: IEEE Industrial Cyber-Physical Systems, 2018, pp. 696–701.

[152] Javaid, U.; Aman, M. N.; Sikdar, B. “BlockPro: Blockchain Based Data Provenance and Integrity for Secure IoT Environments”. In: Workshop on Blockchain-enabled Networked Sensor Systems, 2018, pp. 13–18.

[153] Javaid, U.; Siang, A. K.; Aman, M. N.; Sikdar, B. “Mitigating Iot device based ddos attacks using blockchain”. In: Workshop on Cryptocurrencies and Blockchains for Distributed Systems, 2018, pp. 71–76.

[154] Jeong, J. W.; Kim, B. Y.; Jang, J. W. “Security and Device Control Method for Fog Computer Using Blockchain”. In: International Conference on Information Science and System, 2018, pp. 234–238.

[155] Jerkins, J. A. “Motivating a market or regulatory solution to IoT insecurity with the Mirai botnet code”. In: IEEE Computing and Communication Workshop and Conference, 2017, pp. 1–5.

[156] Jiao, Y.; Wang, P.; Niyato, D.; Xiong, Z. “Social Welfare Maximization Auction in Edge Computing Resource Allocation for Mobile Blockchain”. In: IEEE International Conference on Communications, 2018, pp. 1–6.

[157] Jing, Q.; Vasilakos, A. V.; Wan, J.; Lu, J.; Qiu, D. “Security of the Internet of Things: perspectives and challenges”, *Wireless Networks*, vol. 20–8, Nov 2014, pp. 2481–2501.

[158] Johnson, B.; Laszka, A.; Grossklags, J.; Vasek, M.; Moore, T. “Game-Theoretic Analysis of DDoS Attacks Against Bitcoin Mining Pools”. In: International Conference on Financial Cryptography and Data Security, 2014, pp. 72–86.

[159] Johnson, D.; Menezes, A.; Vanstone, S. “The Elliptic Curve Digital Signature Algorithm (ECDSA)”, *International Journal of Information Security*, vol. 1–1, Aug 2001, pp. 36–63.

[160] Jung, M. Y.; Jang, J. W. “Data management and searching system and method to provide increased security for IoT platform”. In: International Conference on Information and Communication Technology Convergence, 2017, pp. 873–878.

[161] Kak, E.; Orji, R.; Pry, J.; Sofranko, K.; Lomotey, R.; Deters, R. “Privacy Improvement Architecture for IoT”. In: IEEE International Congress on Internet of Things, 2018, pp. 148–155.

[162] Kang, E. S.; Pee, S. J.; Song, J. G.; Jang, J. W. “A Blockchain-Based Energy Trading Platform for Smart Homes in a Microgrid”. In: International Conference on Computer and Communication Systems, 2018, pp. 472–476.

[163] Kapitonov, A.; Berman, I.; Bulatov, V.; Lonshakov, S.; Krupenkin, A. “Robonomics based on blockchain as a principle of creating smart factories”. In: International Conference on Internet of Things: Systems, Management and Security, 2018, pp. 78–85.

[164] Karlsson, K.; Jiang, W.; Wicker, S.; Adams, D.; Ma, E.; van Renesse, R.; Weatherspoon, H. “Vegvisir: A Partition-Tolerant Blockchain for the Internet-of-Things”. In: IEEE International Conference on Distributed Computing Systems, 2018, pp. 1150–1158.

[165] Kataoka, K.; Gangwar, S.; Podili, P. “Trust list: Internet-wide and distributed IoT traffic management using blockchain and SDN”. In: IEEE World Forum on Internet of Things, 2018, pp. 296–301.

[166] Khan, M. A.; Salah, K. “IoT security: Review, blockchain solutions, and open challenges”, *Future Generation Computer Systems*, vol. 82, May 2018, pp. 395 – 411.

[167] Khan, N. “FAST: A MapReduce Consensus for High Performance Blockchains”. In: Workshop on Blockchain-enabled Networked Sensor Systems, 2018, pp. 1–6.

[168] Kim, B. Y.; Choi, S. S.; Jang, J. W. “Data Managing and Service Exchanging on IoT Service Platform Based on Blockchain with Smart Contract and Spatial Data Processing”. In: International Conference on Information Science and System, 2018, pp. 59–63.

[169] Kim, J.-Y.; Moon, S.-M. “Blockchain-based Edge Computing for Deep Neural Network Applications”. In: Workshop on INTelligent Embedded Systems Architectures and Applications, 2018, pp. 53–55.

[170] Kim, S. “Blockchain for a Trust Network Among Intelligent Vehicles”. Elsevier, 2018, *Advances in Computers*, vol. 111, chap. 2, pp. 43 – 68.

[171] King, S.; Nadal, S. “Ppcoin: Peer-to-peer crypto-currency with proof-of-stake”. Source: www.peercoin.net, 26 feb 2019.

[172] Kinkelin, H.; Hauner, V.; Niedermayer, H.; Carle, G. “Trustworthy configuration management for networked devices using distributed ledgers”. In: IEEE/IFIP Network Operations and Management Symposium, 2018, pp. 1–5.

[173] Kouicem, D. E.; Bouabdallah, A.; Lakhlef, H. “Internet of things security: A top-down survey”, *Computer Networks*, vol. 141, Aug 2018, pp. 199 – 221.

[174] Kouzinopoulos, C. S.; Giannoutakis, K. M.; Votis, K.; Tzovaras, D.; Collen, A.; Nijdam, N. A.; Konstantas, D.; Spathoulas, G.; Pandey, P.; Katsikas, S. “Implementing a Forms of Consent Smart Contract on an IoT-based Blockchain to promote user trust”. In: Innovations in Intelligent Systems and Applications, 2018, pp. 1–6.

[175] Kraijak, S.; Tuwanut, P. “A survey on IoT architectures, protocols, applications, security, privacy, real-world implementation and future trends”. In: International Conference on Wireless Communications, Networking and Mobile Computing, 2015, pp. 1–6.

[176] Kramer, J.; van der Werf, J. M.; Stokking, J.; Ruiz, M. “A Blockchain-Based Micro Economy Platform for Distributed Infrastructure Initiatives”. In: IEEE International Conference on Software Architecture, 2018, pp. 11–1109.

[177] Kravitz, D. W. “Transaction Immutability and Reputation Traceability: Blockchain as a Platform for Access Controlled IoT and Human Interactivity”. In: Conference on Privacy, Security and Trust, 2017, pp. 3–309.

[178] Kravitz, D. W.; Cooper, J. “Securing user identity and transactions symbiotically: IoT meets blockchain”. In: Global Internet of Things Summit, 2017, pp. 1–6.

[179] Krishnan, K. N.; Jenu, R.; Joseph, T.; Silpa, M. L. “Blockchain Based Security Framework for IoT Implementations”. In: International CET Conference on Control, Communication, and Computing, 2018, pp. 425–429.

[180] Kshetri, N. “Blockchain’s roles in strengthening cybersecurity and protecting privacy”, *Telecommunications Policy*, vol. 41–10, Nov 2017, pp. 1027 – 1038.

[181] Kshetri, N. “Can Blockchain Strengthen the Internet of Things?”, *IT Professional*, vol. 19–4, Aug 2017, pp. 68–72.

[182] Kshetri, N. “1 Blockchain’s roles in meeting key supply chain management objectives”, *International Journal of Information Management*, vol. 39, Apr 2018, pp. 80 – 89.

[183] Kumar, N. M.; Mallick, P. K. “Blockchain technology for security issues and challenges in IoT”, *Procedia Computer Science*, vol. 132, Sep 2018, pp. 1815 – 1823.

[184] Kundu, A.; Sura, Z.; Sharma, U. “Collaborative and accountable hardware governance using blockchain”. In: IEEE International Conference on Collaboration and Internet Computing, 2018, pp. 114–121.

[185] Kuzmin, A. “Blockchain-based structures for a secure and operate IoT”. In: Internet of Things Business Models, Users, and Networks, 2017, pp. 1–7.

[186] Kwon, Y.; Kim, D.; Son, Y.; Vasserman, E.; Kim, Y. “Be Selfish and Avoid Dilemmas: Fork After Withholding (FAW) Attacks on Bitcoin”. In: ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 195–209.

[187] Lamport, L. “Password authentication with insecure communication”, *Communications of the ACM*, vol. 24–11, Nov 1981, pp. 770–772.

[188] Lao, L.; Dai, X.; Xiao, B.; Guo, S. “G-PBFT: A Location-based and Scalable Consensus Protocol for IoT-Blockchain Applications”. In: IEEE International Parallel and Distributed Processing Symposium, 2020, pp. 664–673.

[189] Laszka, A.; Dubey, A.; Walker, M.; Schmidt, D. “Providing Privacy, Safety, and Security in IoT-based Transactive Energy Systems Using Distributed Ledgers”. In: International Conference on the Internet of Things, 2017, pp. 13:1–13:8.

[190] Lazaroiu, C.; Roscia, M. “Smart district through IoT and Blockchain”. In: IEEE International Conference on Renewable Energy Research and Applications, 2017, pp. 454–461.

[191] Lazaroiu, C.; Roscia, M. “RESCoin to improve Prosumer Side Management into Smart City”. In: International Conference on Renewable Energy Research and Applications, 2018, pp. 1196–1201.

[192] Lazaroiu, G. C.; Roscia, M. “Blockchain and smart metering towards sustainable prosumers”. In: International Symposium on Power Electronics, Electrical Drives, Automation and Motion, 2018, pp. 550–555.

[193] Le, T.; Mutka, M. W. “CapChain: A Privacy Preserving Access Control Framework Based on Blockchain for Pervasive Environments”. In: IEEE International Conference on Smart Computing, 2018, pp. 57–64.

[194] Lee, A. R.; Kim, M. G.; Kim, I. K. “SHAREChain: Healthcare data sharing framework using Blockchain-registry and FHIR”. In: IEEE International Conference on Bioinformatics and Biomedicine, 2019, pp. 1087–1090.

[195] Lee, C.; Nkenyereye, L.; Sung, N.; Song, J. “Towards a Blockchain-enabled IoT Platform using oneM2M Standards”. In: International Conference on Information and Communication Technology Convergence, 2018, pp. 97–102.

[196] Lee, C.; Sung, N.; Nkenyereye, L.; Song, J. “Blockchain Enabled Internet-of-Things Service Platform for Industrial Domain”. In: IEEE International Conference on Industrial Internet, 2018, pp. 177–178.

[197] Lee, J. “BIDaaS: Blockchain Based ID As a Service”, *IEEE Access*, vol. 6, Dec 2018, pp. 2274–2278.

[198] Lei, A.; Cruickshank, H.; Cao, Y.; Asuquo, P.; Ogah, C. P. A.; Sun, Z. “Blockchain-Based Dynamic Key Management for Heterogeneous Intelligent Transportation Systems”, *IEEE Internet of Things Journal*, vol. 4–6, Dec 2017, pp. 1832–1843.

[199] Leiba, O.; Yitzchak, Y.; Bitton, R.; Nadler, A.; Shabtai, A. “Incentivized Delivery Network of IoT Software Updates Based on Trustless Proof-of-Distribution”. In: IEEE European Symposium on Security and Privacy Workshops, 2018, pp. 29–39.

[200] Li, C.; Zhang, L. “A Blockchain Based New Secure Multi-Layer Network Model for Internet of Things”. In: IEEE International Congress on Internet of Things, 2017, pp. 33–41.

[201] Li, D.; Peng, W.; Deng, W.; Gai, F. “A Blockchain-Based Authentication and Security Mechanism for IoT”. In: International Conference on Computer Communication and Networks, 2018, pp. 1–6.

[202] Li, S. “Application of Blockchain Technology in Smart City Infrastructure”. In: IEEE International Conference on Smart Internet of Things, 2018, pp. 276–2766.

[203] Li, Z.; Barenji, A. V.; Huang, G. Q. “Toward a blockchain cloud manufacturing system as a peer to peer distributed network platform”, *Robotics and Computer-Integrated Manufacturing*, vol. 54, Dec 2018, pp. 133 – 144.

[204] Li, Z.; Kang, J.; Yu, R.; Ye, D.; Deng, Q.; Zhang, Y. “Consortium blockchain for secure energy trading in industrial internet of things”, *IEEE Transactions on Industrial Informatics*, vol. 14–8, Aug 2018, pp. 3690–3700.

[205] Liang, X.; Zhao, J.; Shetty, S.; Li, D. “Towards data assurance and resilience in IoT using blockchain”. In: IEEE Military Communications Conference, 2017, pp. 261–266.

[206] Lin, J.; Shen, Z.; Miao, C. “Using Blockchain Technology to Build Trust in Sharing LoRaWAN IoT”. In: International Conference on Crowd Science and Engineering, 2017, pp. 38–43.

[207] Lin, J.; Shen, Z.; Zhang, A.; Chai, Y. “Blockchain and IoT Based Food Traceability for Smart Agriculture”. In: International Conference on Crowd Science and Engineering, 2018, pp. 3:1–3:6.

[208] Litecoin. “Litecoin - Open source P2P digital currency”. Source: https://litecoin.org/, 26 feb 2019.

[209] Liu, B.; Yu, X. L.; Chen, S.; Xu, X.; Zhu, L. “Blockchain Based Data Integrity Service Framework for IoT Data”. In: IEEE International Conference on Web Services, 2017, pp. 468–475.

[210] Liu, D.; Alahmadi, A.; Ni, J.; Lin, X.; Shen, X. “Anonymous Reputation System for IIoT-Enabled Retail Marketing Atop PoS Blockchain”, *IEEE Transactions on Industrial Informatics*, vol. 15–6, Jun 2019, pp. 3527–3537.

[211] Liu, Z.; Seo, H. “IoT-NUMS: Evaluating NUMS Elliptic Curve Cryptography for IoT Platforms”, *IEEE Transactions on Information Forensics and Security*, vol. 14–3, Mar 2019, pp. 720–729.

[212] Lombardi, F.; Aniello, L.; Angelis, S. D.; Margheri, A.; Sassone, V. “A blockchain-based infrastructure for reliable and cost-effective IoT-aided smart grids”. In: Living in the Internet of Things: Cybersecurity of the IoT, 2018, pp. 1–6.

[213] Lunardi, R. C.; Zorzo, A. F. “Estruturando diferentes aplicações com blockchain.” SBC Horizontes, ISSN 2175-9235, Source: http://horizontes.sbc.org.br/index.php/2020/12/estruturando-diferentes-aplicacoes-com-blockchain/>, 15 feb 2021.

[214] Lunardi, R. C.; Alharby, M.; Nunes, H. C.; Dong, C.; Zorzo, A. F.; van Moorsel, A. “Context-based consensus for appendable-block blockchains”. In: IEEE International Conference on Blockchain, 2020, pp. 401–408.

[215] Lunardi, R. C.; Michelin, R. A.; Neu, C. V.; Nunes, H. C.; Zorzo, A. F.; Kanhere, S. S. “Impact of Consensus on Appendable-Block Blockchain for IoT”. In: 2EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2019, pp. 228–237.

[216] Lunardi, R. C.; Michelin, R. A.; Neu, C. V.; Zorzo, A. F. “Distributed access control on IoT ledger-based architecture”. In: IEEE/IFIP Network Operations and Management Symposium, 2018, pp. 1–7.

[217] Lunardi, R. C.; Nunes, H. C.; Branco, V.; Lippert, B.; Neu, C. V.; Zorzo, A. F. “Performance and Cost Evaluation of Smart Contracts in Collaborative Health Care Environments”. In: International Conference for Internet Technology and Secured Transactions, 2019a, pp. 1–6.

[218] Lundqvist, T.; de Blanche, A.; Andersson, H. R. H. “Thing-to-thing electricity micro payments using blockchain technology”. In: Global Internet of Things Summit, 2017, pp. 1–6.

[219] Luong, N. C.; Xiong, Z.; Wang, P.; Niyato, D. “Optimal Auction for Edge Computing Resource Management in Mobile Blockchain Networks: A Deep Learning Approach”. In: IEEE International Conference on Communications, 2018, pp. 1–6.

[220] Machado, C.; Fröhlich, A. A. M. “IoT Data Integrity Verification for Cyber-Physical Systems Using Blockchain”. In: IEEE International Symposium on Real-Time Distributed Computing, 2018, pp. 83–90.

[221] Maitra, S.; Yanambaka, V. P.; Abdelgawad, A.; Puthal, D.; Yelamarthi, K. “Proof-of-Authentication Consensus Algorithm: Blockchain-based IoT Implementation”. In: IEEE World Forum on Internet of Things, 2020, pp. 1–2.

[222] Makhdoom, I.; Abolhasan, M.; Abbas, H.; Ni, W. “Blockchain’s adoption in IoT: The challenges, and a way forward”, *Journal of Network and Computer Applications*, vol. 125, Jan 2019, pp. 251 – 279.

[223] Makhdoom, I.; Tofigh, F.; Zhou, I.; Abolhasan, M.; Lipman, J. “PLEDGE: A Proof-of-Honesty based Consensus Protocol for Blockchain-based IoT Systems”. In: IEEE International Conference on Blockchain and Cryptocurrency, 2020, pp. 1–3.

[224] Malik, S.; Dedeoglu, V.; Kanhere, S. S.; Jurdak, R. “TrustChain: Trust Management in Blockchain and IoT Supported Supply Chains”. In: IEEE International Conference on Blockchain, 2019, pp. 184–193.

[225] Mavridou, A.; Laszka, A. “Designing Secure Ethereum Smart Contracts: A Finite State Machine Based Approach”. In: International Conference on Financial Cryptography and Data Security, 2018, pp. 523–540.

[226] Mazières, D. “The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus”. Source: https://www.stellar.org/papers/stellar-consensus-protocol, 26 feb 2019.

[227] Melo, Jr, W. S.; Bessani, A.; Carmo, L. F. R. C. “How Blockchains Can Help Legal Metrology”. In: Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers, 2017, pp. 5:1–5:2.

[228] Meloni, A.; Madanapalli, S.; Divakaran, S. K.; Browdy, S. F.; Paranthaman, A.; Jasti, A.; Krishna, N.; Kumar, D. “Exploiting the IoT Potential of Blockchain in the IEEE P1931.1 ROOF Standard”, *IEEE Communications Standards Magazine*, vol. 2–3, Sep 2018, pp. 38–44.

[229] Mendez Mena, D. M.; Yang, B. “Blockchain-Based Whitelisting for Consumer IoT Devices and Home Networks”. In: SIG Conference on Information Technology Education, 2018, pp. 7–12.

[230] Mermer, G. B.; Zeydan, E.; Arslan, S. S. “An overview of blockchain technologies: Principles, opportunities and challenges”. In: Signal Processing and Communications Applications Conference, 2018, pp. 1–4.

[231] Mettler, M. “Blockchain technology in healthcare: The revolution starts here”. In: IEEE International Conference on e-Health Networking, Applications and Services, 2016, pp. 1–3.

[232] Michelin, R. A.; Dorri, A.; Steger, M.; Lunardi, R. C.; Kanhere, S. S.; Jurdak, R.; Zorzo, A. F. “SpeedyChain: A Framework for Decoupling Data from Blockchain for Smart Cities”. In: EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2018, pp. 145–154.

[233] Miller, D. “Blockchain and the Internet of Things in the Industrial Sector”, *IT Professional*, vol. 20–3, May 2018, pp. 15–18.

[234] Minoli, D.; Occhiogrosso, B. “Blockchain mechanisms for IoT security”, *Internet of Things*, vol. 1-2, Sep 2018, pp. 1 – 13.

[235] Missier, P.; Bajoudah, S.; Capossele, A.; Gaglione, A.; Nati, M. “Mind My Value: A Decentralized Infrastructure for Fair and Trusted IoT Data Trading”. In: International Conference on the Internet of Things, 2017, pp. 15:1–15:8.

[236] Mo, B.; Su, K.; Wei, S.; Liu, C.; Guo, J. “A Solution for Internet of Things based on Blockchain Technology”. In: IEEE International Conference on Service Operations and Logistics, and Informatics, 2018, pp. 112–117.

[237] Mohan, S.; Asplund, M.; Bloom, G.; Sadeghi, A.; Ibrahim, A.; Salajageh, N.; Griffioen, P.; Sinipoli, B. “Special Session: The Future of IoT Security”. In: International Conference on Embedded Software, 2018, pp. 1–7.

[238] Mohanta, B. K.; Panda, S. S.; Jena, D. “An Overview of Smart Contract and Use Cases in Blockchain Technology”. In: International Conference on Computing, Communication and Networking Technologies, 2018, pp. 1–4.

[239] Mozaffari-Kermani, M.; Azarderakhsh, R.; Aghaie, A. “Fault Detection Architectures for Post-Quantum Cryptographic Stateless Hash-Based Secure Signatures Benchmarked on ASIC”, *ACM Transactions on Embedded Computing Systems*, vol. 16–2, Dec 2016, pp. 1–19.

[240] Mukhopadhyay, U.; Skjellum, A.; Hambolu, O.; Oakley, J.; Yu, L.; Brooks, R. “A brief survey of Cryptocurrency systems”. In: Conference on Privacy, Security and Trust, 2016, pp. 745–752.

[241] Mylrea, M.; Gourisetti, S. N. G. “Blockchain for Supply Chain Cybersecurity, Optimization and Compliance”. In: Resilience Week, 2018, pp. 70–76.

[242] Nakamoto, S. “Bitcoin: A peer-to-peer electronic cash system”. Source: https://bitcoin.org/bitcoin.pdf, 26 feb 2019.

[243] Neu, C.; Trebien, E.; Bertoglio, D.; Lunardi, R.; Zorzo, A. “Extração e gerenciamento de incidentes em SIEM”. In: Escola Regional de Redes de Computadores, 2019, pp. 190–195.

[244] Neu, C.; Trebien, E.; Bertoglio, D.; Lunardi, R.; Zorzo, A. “Gerenciamento de incidentes em siem seguindo itil”, *Revista Eletrônica Argentina-Brasil de Tecnologias da Informação e da Comunicação*, vol. 3–1, Mar 2020, pp. 1–10.

[245] Neu, C. V.; Tatsch, C. G.; Lunardi, R. C.; Michelin, R. A.; Orozco, A. M. S.; Zorzo, A. F. “Lightweight IPS for port scan in OpenFlow SDN networks”. In: IEEE/IFIP Network Operations and Management Symposium Workshops, 2018, pp. 1–6.

[246] Ngamsuriyaroj, S.; Likittheerameth, T.; Kahutson, A.; Pathummasut, T. “Package Delivery System Based on Blockchain Infrastructure”. In: ICT International Student Project Conference, 2018, pp. 1–6.

[247] Niya, S. R.; Jha, S. S.; Bocek, T.; Stiller, B. “Design and implementation of an automated and decentralized pollution monitoring system with blockchains, smart contracts, and LoRaWAN”. In: IEEE/IFIP Network Operations and Management Symposium, 2018, pp. 1–4.

[248] Noor, S.; Yang, W.; Guo, M.; van Dam, K. H.; Wang, X. “Energy Demand Side Management within micro-grid networks enhanced by blockchain”, *Applied Energy*, vol. 228, Oct 2018, pp. 1385 – 1398.

[249] Norta, A.; Fernandez, C.; Hickmott, S. “Commercial Property Tokenizing With Smart Contracts”. In: International Joint Conference on Neural Networks, 2018, pp. 1–8.

[250] Novo, O. “Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT”, *IEEE Internet of Things Journal*, vol. 5–2, Apr 2018, pp. 1184–1195.

[251] Nunes, H. C.; Lunardi, R. C.; Zorzo, A. F.; Michelin, R. A.; Kanhere, S. S. “Context-based Smart Contracts For Appendable-block Blockchains”. In: IEEE International Conference on Blockchain and Cryptocurrency, 2020, pp. 1–9.

[252] O’Hara, K. “Smart contracts - dumb idea”, *IEEE Internet Computing*, vol. 21–2, Mar 2017, pp. 97–101.

[253] Ongaro, D.; Ousterhout, J. “In search of an understandable consensus algorithm”. In: Conference on USENIX Technical Conference, 2014, pp. 305–320.

[254] Ouaddah, A.; Elkalam, A. A.; Ouahman, A. A. “Harnessing the Power of Blockchain Technology to Solve IoT Security & Privacy Issues”. In: International Conference on Internet of Things, Data and Cloud Computing, 2017, pp. 7:1–7:10.

[255] Paavolainen, S.; Nikander, P. “Security and Privacy Challenges and Potential Solutions for DLT based IoT Systems”. In: Global Internet of Things Summit, 2018, pp. 1–6.

[256] Pahl, C.; Ioini, N. E.; Helmer, S.; Lee, B. “An architecture pattern for trusted orchestration in IoT edge clouds”. In: International Conference on Fog and Mobile Edge Computing, 2018, pp. 63–70.

[257] Pan, J.; Yang, Z. “Cybersecurity Challenges and Opportunities in the New "Edge Computing + IoT" World”. In: ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization, 2018, pp. 29–32.

[258] Papadis, N.; Borst, S.; Walid, A.; Grissa, M.; Tassiulas, L. “Stochastic models and wide-area network measurements for blockchain design and analysis”. In: IEEE Conference on Computer Communications, 2018, pp. 2546–2554.

[259] Park, J.; Kim, K. “TM-Coin: Trustworthy management of TCB measurements in IoT”. In: IEEE International Conference on Pervasive Computing and Communications Workshops, 2017, pp. 654–659.

[260] Patil, M. A.; Karule, P. T. “Design and implementation of keccak hash function for cryptography”. In: International Conference on Communications and Signal Processing, 2015, pp. 0875–0878.

[261] Peck, M. E. “Blockchains: How they work and why they’ll change the world”, *IEEE Spectrum*, vol. 54–10, Oct 2017, pp. 26–35.

[262] Petzoldt, A.; Bulygin, S.; Buchmann, J. “Fast Verification for Improved Versions of the UOV and Rainbow Signature Schemes”. In: International Workshop on Post-Quantum Cryptography, 2013, pp. 188–202.

[263] Pinno, O. J. A.; Gregio, A. R. A.; Bona, L. C. E. D. “ControlChain: Blockchain as a Central Enabler for Access Control Authorizations in the IoT”. In: IEEE Global Communications Conference, 2017, pp. 1–6.

[264] Plaza, C.; Gil, J.; de Chezelles, F.; Strang, K. A. “Distributed Solar Self-Consumption and Blockchain Solar Energy Exchanges on the Public Grid Within an Energy Community”. In: IEEE International Conference on Environment and Electrical Engineering, 2018, pp. 1–4.

[265] Ploennigs, J.; Cohn, J.; Stanford-Clark, A. “The Future of IoT”, *IEEE Internet of Things Magazine*, vol. 1–1, Sep 2018, pp. 28–33.

[266] Polyzos, G. C.; Fotiou, N. “Blockchain-Assisted Information Distribution for the Internet of Things”. In: IEEE International Conference on Information Reuse and Integration, 2017, pp. 75–78.

[267] Psaras, I. “Decentralised Edge-Computing and IoT Through Distributed Trust”. In: International Conference on Mobile Systems, Applications, and Services, 2018, pp. 505–507.

[268] Pustišek, M.; Kos, A. “Approaches to Front-End IoT Application Development for the Ethereum Blockchain”, *Procedia Computer Science*, vol. 129, Sep 2018, pp. 410 – 419.

[269] Pustišek, M.; Kos, A.; Sedlar, U. “Blockchain Based Autonomous Selection of Electric Vehicle Charging Station”. In: International Conference on Identification, Information and Knowledge in the Internet of Things, 2016, pp. 217–222.

[270] Puthal, D.; Mohanty, S. P. “Proof of Authentication: IoT-Friendly Blockchains”, *IEEE Potentials*, vol. 38–1, Jan 2019, pp. 26–29.

[271] Qian, Y.; Jiang, Y.; Chen, J.; Zhang, Y.; Song, J.; Zhou, M.; Pustišek, M. “Towards decentralized IoT security enhancement: A blockchain approach”, *Computers & Electrical Engineering*, vol. 72, Nov 2018, pp. 266 – 273.

[272] Qiu, Z.; Hao, J.; Guo, Y.; Zhang, Y. “Dual Vote Confirmation based Consensus Design for Blockchain integrated IoT”. In: IEEE/IFIP Network Operations and Management Symposium, 2020, pp. 1–7.

[273] Raghav; Andola, N.; Venkatesan, S.; Verma, S. “PoEWAL: A lightweight consensus mechanism for blockchain in IoT”, *Pervasive and Mobile Computing*, vol. 69, Nov 2020, pp. 101291:1–101291:12.

[274] Rahim, K.; Tahir, H.; Ikram, N. “Sensor Based PUF IoT Authentication Model for a Smart Home with Private Blockchain”. In: International Conference on Applied and Engineering Mathematics, 2018, pp. 102–108.

[275] Rahman, M. A.; Hassanain, E.; Rashid, M. M.; Barnes, S. J.; Hossain, M. S. “Spatial blockchain-based secure mass screening framework for children with dyslexia”, *IEEE Access*, vol. 6, Oct 2018, pp. 61876–61885.

[276] Rahman, M. A.; Hossain, M. S.; Loukas, G.; Hassanain, E.; Rahman, S. S.; Alhamid, M. F.; Guizani, M. “Blockchain-Based Mobile Edge Computing Framework for Secure Therapy Applications”, *IEEE Access*, vol. 6, Nov 2018, pp. 72469–72478.

[277] Rahulamathavan, Y.; Phan, R. C. .; Rajarajan, M.; Misra, S.; Kondoz, A. “Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption”. In:

IEEE International Conference on Advanced Networks and Telecommunications Systems, 2017, pp. 1–6.

[278] Rawat, D. B.; Alshaikhi, A. “Leveraging Distributed Blockchain-based Scheme for Wireless Network Virtualization with Security and QoS Constraints”. In: International Conference on Computing, Networking and Communications, 2018, pp. 332–336.

[279] Regnath, E.; Steinhorst, S. “SmaCoNat: Smart Contracts in Natural Language”. In: Forum on Specification Design Languages, 2018, pp. 5–16.

[280] Reyna, A.; Martín, C.; Chen, J.; Soler, E.; Díaz, M. “On blockchain and its integration with IoT. Challenges and opportunities”, *Future Generation Computer Systems*, vol. 88, Nov 2018, pp. 173 – 190.

[281] Rifi, N.; Rachkidi, E.; Agoulmine, N.; Taher, N. C. “Towards using blockchain technology for IoT data access protection”. In: IEEE International Conference on Ubiquitous Wireless Broadband, 2017, pp. 1–5.

[282] Rivest, R. L.; Shamir, A.; Adleman, L. “A method for obtaining digital signatures and public-key cryptosystems”, *Commununications of the ACM*, vol. 21–2, Feb 1978, pp. 120–126.

[283] S. B. Dhaou, N. L. “Analysing the transformational Effect of Emerging Technologies on Smart Cities: Blockchain and IoT”. In: International Conference on Smart and Sustainable Technologies, 2018, pp. 1–10.

[284] Sadique, K. M.; Rahmani, R.; Johannesson, P. “Towards Security on Internet of Things: Applications and Challenges in Technology”, *Procedia Computer Science*, vol. 141, Sep 2018, pp. 199 – 206.

[285] Saghiri, A. M.; Vahdati, M.; Gholizadeh, K.; Meybodi, M. R.; Dehghan, M.; Rashidi, H. “A framework for cognitive Internet of Things based on blockchain”. In: International Conference on Web Research, 2018, pp. 138–143.

[286] Sagirlar, G.; Carminati, B.; Ferrari, E. “AutoBotCatcher: Blockchain-Based P2P Botnet Detection for the Internet of Things”. In: IEEE International Conference on Collaboration and Internet Computing, 2018, pp. 1–8.

[287] Sakakibara, Y.; Nakamura, K.; Matsutani, H. “An FPGA NIC Based Hardware Caching for Blockchain”. In: International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies, 2017, pp. 1:1–1:6.

[288] Salahuddin, M. A.; Al-Fuqaha, A.; Guizani, M.; Shuaib, K.; Sallabi, F. “Softwarization of Internet of Things Infrastructure for Secure and Smart Healthcare”, *Computer*, vol. 50–7, Jul 2017, pp. 74–79.

[289] Samaniego, M.; Deters, R. “Blockchain as a Service for IoT”. In: IEEE International Conference on Internet of Things and IEEE Green Computing and Communications and IEEE Cyber, Physical and Social Computing and IEEE Smart Data, 2016, pp. 433–436.

[290] Samaniego, M.; Deters, R. “Using Blockchain to Push Software-Defined IoT Components Onto Edge Hosts”. In: International Conference on Big Data and Advanced Wireless Technologies, 2016, pp. 58:1–58:9.

[291] Samaniego, M.; Deters, R. “Internet of Smart Things - IoST: Using Blockchain and CLIPS to Make Things Autonomous”. In: IEEE International Conference on Cognitive Computing, 2017, pp. 9–16.

[292] Samaniego, M.; Deters, R. “Zero-Trust Hierarchical Management in IoT”. In: IEEE International Congress on Internet of Things, 2018, pp. 88–95.

[293] Samaniego, M.; Espana, C.; Deters, R. “Smart Virtualization for IoT”. In: IEEE International Conference on Smart Cloud, 2018, pp. 125–128.

[294] Sankaran, S.; Sanju, S.; Achuthan, K. “Towards Realistic Energy Profiling of Blockchains for Securing Internet of Things”. In: IEEE International Conference on Distributed Computing Systems, 2018, pp. 1454–1459.

[295] Sanseverino, E. R.; Silvestre, M. L. D.; Gallo, P.; Zizzo, G.; Ippolito, M. “The Blockchain in Microgrids for Transacting Energy and Attributing Losses”. In: IEEE International Conference on Internet of Things and IEEE Green Computing and Communications and IEEE Cyber, Physical and Social Computing and IEEE Smart Data, 2017, pp. 925–930.

[296] Saraf, C.; Sabadra, S. “Blockchain platforms: A compendium”. In: IEEE International Conference on Innovative Research and Development, 2018, pp. 1–6.

[297] Saravanan, M.; Shubha, R.; Marks, A. M.; Iyer, V. “SMEAD: A secured mobile enabled assisting device for diabetics monitoring”. In: IEEE International Conference on Advanced Networks and Telecommunications Systems, 2017, pp. 1–6.

[298] Schwartz, D.; Youngs, N.; Britto, A. “The ripple protocol consensus algorithm”. Source: https://ripple.com, 26 feb 2019.

[299] Search, I. “Iota search - transactions overview”. Source: https://iotasear.ch/live-transactions, 26 jan 2021.

[300] Seitz, A.; Henze, D.; Miehle, D.; Bruegge, B.; Nickles, J.; Sauer, M. “Fog Computing as Enabler for Blockchain-Based IIoT App Marketplaces - A Case Study”. In: International

Conference on Internet of Things: Systems, Management and Security, 2018, pp. 182–188.

[301] Shae, Z.; Tsai, J. J. P. “On the Design of a Blockchain Platform for Clinical Trial and Precision Medicine”. In: IEEE International Conference on Distributed Computing Systems, 2017, pp. 1972–1980.

[302] Shafagh, H.; Burkhalter, L.; Hithnawi, A.; Duquennoy, S. “Towards Blockchain-based Auditable Storage and Sharing of IoT Data”. In: Cloud Computing Security Workshop, 2017, pp. 45–50.

[303] Shala, B.; Wacht, P.; Trick, U.; Lehmann, A.; Shala, B.; Ghita, B.; Shiaeles, S. “Ensuring trustworthiness for P2P-based M2M applications”. In: Internet Technologies and Applications, 2017, pp. 58–63.

[304] Shandilya, A.; Gupta, H.; Khatri, S. K. “Role and Aplications of IoT in Online Transactions using Blockchain Technology”. In: International Conference on Advances in Computing and Communication Engineering, 2018, pp. 465–470.

[305] Sharma, P. K.; Chen, M.; Park, J. H. “A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT”, *IEEE Access*, vol. 6, Sep 2018, pp. 115–124.

[306] Sharma, P. K.; Park, J. H. “Blockchain based hybrid network architecture for the smart city”, *Future Generation Computer Systems*, vol. 86, Sep 2018, pp. 650 – 655.

[307] Sharma, P. K.; Rathore, S.; Jeong, Y.; Park, J. H. “SoftEdgeNet: SDN Based Energy-Efficient Distributed Network Architecture for Edge Computing”, *IEEE Communications Magazine*, vol. 56–12, Dec 2018, pp. 104–111.

[308] Sharma, P. K.; Singh, S.; Jeong, Y. S.; Park, J. H. “DistBlockNet: A Distributed Blockchains-Based Secure SDN Architecture for IoT Networks”, *IEEE Communications Magazine*, vol. 55–9, Sep 2017, pp. 78–85.

[309] Sharma, V.; You, I.; Palmieri, F.; Jayakody, D. N. K.; Li, J. “Secure and Energy-Efficient Handover in Fog Networks Using Blockchain-Based DMM”, *IEEE Communications Magazine*, vol. 56–5, May 2018, pp. 22–31.

[310] Shrestha, A. K.; Vassileva, J. “Towards Decentralized Data Storage in General Cloud Platform for Meta-products”. In: International Conference on Big Data and Advanced Wireless Technologies, 2016, pp. 60:1–60:7.

[311] Singh, A.; Kumar, D.; Hötzel, J. “IoT Based information and communication system for enhancing underground mines safety and productivity: Genesis, taxonomy and open issues”, *Ad Hoc Networks*, vol. 78, Sep 2018, pp. 115 – 129.

[312] Singh, M.; Kim, S. “Trust Bit: Reward-based intelligent vehicle commination using blockchain paper”. In: IEEE World Forum on Internet of Things, 2018, pp. 62–67.

[313] Singh, M.; Singh, A.; Kim, S. “Blockchain: A game changer for securing IoT data”. In: IEEE World Forum on Internet of Things, 2018, pp. 51–55.

[314] Singla, A.; Bertino, E. “Blockchain-Based PKI Solutions for IoT”. In: IEEE International Conference on Collaboration and Internet Computing, 2018, pp. 9–15.

[315] Singla, K.; Bose, J.; Katariya, S. “Machine Learning for Secure Device Personalization Using Blockchain”. In: International Conference on Advances in Computing, Communications and Informatics, 2018, pp. 67–73.

[316] Sok, K.; Colin, J. N.; Po, K. “Blockchain and Internet of Things Opportunities and Challenges”. In: International Symposium on Information and Communication Technology, 2018, pp. 150–154.

[317] Solat, S. “Rdv: An alternative to proof-of-work and a real decentralized consensus for blockchain”. In: Workshop on Blockchain-enabled Networked Sensor Systems, 2018, pp. 25–31.

[318] Song, J. C.; Demir, M. A.; Prevost, J. J.; Rad, P. “Blockchain Design for Trusted Decentralized IoT Networks”. In: Conference on System of Systems Engineering, 2018, pp. 169–174.

[319] Spathoulas, G.; Collen, A.; Pandey, P.; Nijdam, N. A.; Katsikas, S.; Kouzinopoulos, C. S.; Moussa, M. B.; Giannoutakis, K. M.; Votis, K.; Tzovaras, D. “Towards Reliable Integrity in Blacklisting: Facing Malicious IPs in GHOST Smart Contracts”. In: Innovations in Intelligent Systems and Applications, 2018, pp. 1–8.

[320] Stanciu, A. “Blockchain Based Distributed Control System for Edge Computing”. In: International Conference on Control Systems and Computer Science, 2017, pp. 667–671.

[321] Steichen, M.; Hommes, S.; State, R. “ChainGuard — A firewall for blockchain applications using SDN with OpenFlow”. In: Principles, Systems and Applications of IP Telecommunications, 2017, pp. 1–8.

[322] Stellar. “Stelar - get started with blockchain”. Source: https://www.stellar.org/, 26 feb 2019.

[323] Suankaewmanee, K.; Hoang, D. T.; Niyato, D.; Sawadsitang, S.; Wang, P.; Han, Z. “Performance Analysis and Application of Mobile Blockchain”. In: International Conference on Computing, Networking and Communications, 2018, pp. 642–646.

[324] Suchaad, S. A.; Mashiko, K.; Ismail, N. B.; Abidin, M. H. Z. “Blockchain Use in Home Automation for Children Incentives in Parental Control”. In: International Conference on Machine Learning and Machine Intelligence, 2018, pp. 50–53.

[325] Sukhwani, H.; Martínez, J. M.; Chang, X.; Trivedi, K. S.; Rindos, A. “Performance Modeling of PBFT Consensus Process for Permissioned Blockchain Network (Hyperledger Fabric)”. In: IEEE Symposium on Reliable Distributed Systems, 2017, pp. 253–255.

[326] Tapas, N.; Merlino, G.; Longo, F. “Blockchain-Based IoT-Cloud Authorization and Delegation”. In: IEEE International Conference on Smart Computing, 2018, pp. 411–416.

[327] Teslya, N.; Ryabchikov, I. “Blockchain-based platform architecture for industrial IoT”. In: Conference of Open Innovations Association, 2017, pp. 321–329.

[328] Teslya, N.; Ryabchikov, I. “Blockchain Platforms Overview for Industrial IoT Purposes”. In: Conference of Open Innovations Association, 2018, pp. 250–256.

[329] Theodorou, S.; Sklavos, N. “Blockchain-Based Security and Privacy in Smart Cities”. Elsevier, 2019, chap. 3, pp. 21 – 37.

[330] Tian, F. “A supply chain traceability system for food safety based on HACCP, blockchain & Internet of things”. In: International Conference on Service Systems and Service Management, 2017, pp. 1–6.

[331] Truong, C. N.; Schimpe, M.; Bürger, U.; Hesse, H. C.; Jossen, A. “Multi-Use of Stationary Battery Storage Systems with Blockchain Based Markets”, *Energy Procedia*, vol. 155, Nov 2018, pp. 3 – 16.

[332] Tselios, C.; Politis, I.; Kotsopoulos, S. “Enhancing SDN security for IoT-related deployments through blockchain”. In: IEEE Conference on Network Function Virtualization and Software Defined Networks, 2017, pp. 303–308.

[333] Uddin, M. A.; Stranieri, A.; Gondal, I.; Balasubramanian, V. “Continuous patient monitoring with a patient centric agent: A block architecture”, *IEEE Access*, vol. 6, Jun 2018, pp. 32700–32726.

[334] Urien, P. “Towards secure elements for trusted transactions in blockchain and blochchain IoT (BIoT) Platforms. Invited paper”. In: International Conference on Mobile and Secure Services, 2018, pp. 1–5.

[335] Vahdati, M.; HamlAbadi, K. G.; Saghiri, A. M.; Rashidi, H. “A Self-Organized Framework for Insurance Based on Internet of Things and Blockchain”. In: IEEE International Conference on Future Internet of Things and Cloud, 2018, pp. 169–175.

[336] Vairam, P. K.; Mitra, G.; Rebeiro, C.; Ramamurthy, B.; Veezhinathan, K. “ApproxBC: Blockchain Design Alternatives for Approximation-Tolerant Resource-Constrained Applications”, *IEEE Communications Standards Magazine*, vol. 2–3, Sep 2018, pp. 45–51.

[337] Varshney, G.; Gupta, H. “A security framework for IOT devices against wireless threats”. In: International Conference on Telecommunication and Networks, 2017, pp. 1–6.

[338] Vasek, M.; Thornton, M.; Moore, T. “Empirical Analysis of Denial-of-Service Attacks in the Bitcoin Ecosystem”. In: International Conference on Financial Cryptography and Data Security, 2014, pp. 57–71.

[339] Vasin, P. “Blackcoin’s proof-of-stake protocol v2”. Source: https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf, 26 feb 2019.

[340] Villari, M.; Galletta, A.; Celesti, A.; Carnevale, L.; Fazio, M. “Osmotic Computing: Software Defined Membranes meet Private/Federated Blockchains”. In: IEEE Symposium on Computers and Communications, 2018, pp. 01292–01297.

[341] Vitale, F.; Janzen, I.; McGrenere, J. “Hoarding and Minimalism: Tendencies in Digital Data Preservation”. In: Conference on Human Factors in Computing Systems, 2018, pp. 1–12.

[342] Walker, M. A.; Dubey, A.; Laszka, A.; Schmidt, D. C. “PlaTIBART: A Platform for Transactive IoT Blockchain Applications with Repeatable Testing”. In: Workshop on Middleware and Applications for the Internet of Things, 2017, pp. 17–22.

[343] Wang, J.; Li, M.; He, Y.; Li, H.; Xiao, K.; Wang, C. “A Blockchain Based Privacy-Preserving Incentive Mechanism in Crowdsensing Applications”, *IEEE Access*, vol. 6, Dec 2018, pp. 17545–17556.

[344] Wang, S.; Yuan, Y.; Wang, X.; Li, J.; Qin, R.; Wang, F. “An Overview of Smart Contract: Architecture, Applications, and Future Trends”. In: IEEE Intelligent Vehicles Symposium, 2018, pp. 108–113.

[345] Wang, Z.; Dong, X.; Li, Y.; Fang, L.; Chen, P. “IoT Security Model and Performance Evaluation: A Blockchain Approach”. In: International Conference on Network Infrastructure and Digital Content, 2018, pp. 260–264.

[346] Waves. “Waves - get started with blockchain”. Source: https://wavesplatform.com/, 26 feb 2019.

[347] Weaver, N. “Risks of cryptocurrencies”, *Commununications of the ACM*, vol. 61–6, Jun 2018, pp. 20–24.

[348] Wen, B.; Luo, Z.; Wen, Y. “Evidence and Trust: IoT Collaborative Security Mechanism”. In: International Conference on Information Science and Technology, 2018, pp. 98–9.

[349] Wessling, F.; Ehmke, C.; Hesenius, M.; Gruhn, V. “How Much Blockchain Do You Need? Towards a Concept for Building Hybrid DApp Architectures”. In: IEEE/ACM International Workshop on Emerging Trends in Software Engineering for Blockchain, 2018, pp. 44–47.

[350] Wibowo, S.; Hw, E. P. “Blockchain Implementation Assessment Framework, Case Study of IoT LPWA Licensing in Indonesia”. In: International Conference on ICT for Smart Society, 2018, pp. 1–5.

[351] Wright, C.; Serguieva, A. “Sustainable blockchain-enabled services: Smart contracts”. In: IEEE International Conference on Big Data, 2017, pp. 4255–4264.

[352] Wu, B.; Li, Q.; Xu, K.; Li, R.; Liu, Z. “SmartRetro: Blockchain-Based Incentives for Distributed IoT Retrospective Detection”. In: IEEE International Conference on Mobile Ad Hoc and Sensor Systems, 2018, pp. 308–316.

[353] Wu, L.; Du, X.; Wang, W.; Lin, B. “An Out-of-band Authentication Scheme for Internet of Things Using Blockchain Technology”. In: International Conference on Computing, Networking and Communications, 2018, pp. 769–773.

[354] Wu, X.; Duan, B.; Yan, Y.; Zhong, Y. “M2M Blockchain: The Case of Demand Side Management of Smart Grid”. In: IEEE International Conference on Parallel and Distributed Systems, 2017, pp. 810–813.

[355] Xie, C.; Sun, Y.; Luo, H. “Secured Data Storage Scheme Based on Block Chain for Agricultural Products Tracking”. In: International Conference on Big Data Computing and Communications, 2017, pp. 45–50.

[356] Xiong, Z.; Zhang, Y.; Niyato, D.; Wang, P.; Han, Z. “When mobile blockchain meets edge computing”, *IEEE Communications Magazine*, vol. 56–8, Aug 2018, pp. 33–39.

[357] Xu, D.; Xiao, L.; Sun, L.; Lei, M. “Game theoretic study on blockchain based secure edge networks”. In: IEEE/CIC International Conference on Communications in China, 2017, pp. 1–5.

[358] Xu, L.; Chen, L.; Gao, Z.; Xu, S.; Shi, W. “EPBC: Efficient Public Blockchain Client for Lightweight Users”. In: Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers, 2017, pp. 1:1–1:6.

[359] Yang, H.; Zheng, H.; Zhang, J.; Wu, Y.; Lee, Y.; Ji, Y. “Blockchain-based trusted authentication in cloud radio over fiber network for 5G”. In: International Conference on Optical Communications and Networks, 2017, pp. 1–3.

[360] Yang, J.; Lu, Z.; Wu, J. “Smart-toy-edge-computing-oriented data exchange based on blockchain”, *Journal of Systems Architecture*, vol. 87, Jun 2018, pp. 36 – 48.

[361] Yang, Z.; Zheng, K.; Yang, K.; Leung, V. C. M. “A blockchain-based reputation system for data credibility assessment in vehicular networks”. In: IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications, 2017, pp. 1–5.

[362] Yeow, K.; Gani, A.; Ahmad, R. W.; Rodrigues, J. J. P. C.; Ko, K. “Decentralized Consensus for Edge-Centric Internet of Things: A Review, Taxonomy, and Research Issues”, *IEEE Access*, vol. 6, Dec 2018, pp. 1513–1524.

[363] Yin, H.; Guo, D.; Wang, K.; Jiang, Z.; Lyu, Y.; Xing, J. “Hyperconnected Network: A Decentralized Trusted Computing and Networking Paradigm”, *IEEE Network*, vol. 32–1, Jan 2018, pp. 112–117.

[364] Yu, B.; Wright, J.; Nepal, S.; Zhu, L.; Liu, J.; Ranjan, R. “IoTChain: Establishing Trust in the Internet of Things Ecosystem Using Blockchain”, *IEEE Cloud Computing*, vol. 5–4, Jul 2018, pp. 12–23.

[365] Yu, F. R.; Liu, J.; He, Y.; Si, P.; Zhang, Y. “Virtualization for Distributed Ledger Technology (vDLT)”, *IEEE Access*, vol. 6, Apr 2018, pp. 25019–25028.

[366] Zhang, Y.; Wen, J. “An IoT electric business model based on the protocol of bitcoin”. In: International Conference on Intelligence in Next Generation Networks, 2015, pp. 184–191.

[367] Zhao, Y.; Yu, Y.; Li, Y.; Han, G.; Du, X. “Machine learning based privacy-preserving fair data trading in big data market”, *Information Sciences*, vol. 478, Apr 2019, pp. 449 – 460.

[368] Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends”. In: IEEE International Congress on Big Data, 2017, pp. 557–564.

[369] Zhou, L.; Wang, L.; Sun, Y.; Lv, P. “BeeKeeper: A Blockchain-Based IoT System With Secure Storage and Homomorphic Computation”, *IEEE Access*, vol. 6, Dec 2018, pp. 43472–43488.

[370] Zhu, X.; Badr, Y.; Pacheco, J.; Hariri, S. “Autonomic Identity Framework for the Internet of Things”. In: International Conference on Cloud and Autonomic Computing, 2017, pp. 69–79.

[371] Zorzo, A. F.; Nunes, H. C.; Lunardi, R. C.; Michelin, R. A.; Kanhere, S. S. “Dependable IoT Using Blockchain-Based Technology”. In: Latin-American Symposium on Dependable Computing, 2018, pp. 1–9.

[372] Zouari, J.; Hamdi, M.; Kom, T. “Privacy Preserving Profile Matching Protocol for Human-Centric Social Internet of Things”. In: IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2018, pp. 181–186.

[373] Özyilmaz, K. R.; Doğan, M.; Yurdakul, A. “IDMoB: IoT Data Marketplace on Blockchain”. In: Crypto Valley Conference on Blockchain Technology, 2018, pp. 11–19.

[374] Özyılmaz, K. R.; Yurdakul, A. “Work-in-progress: integrating low-power IoT devices to a blockchain-based infrastructure”. In: International Conference on Embedded Software, 2017, pp. 1–2.

Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Graduação
Av. Ipiranga, 6681 - Prédio 1 - 3º. andar
Porto Alegre - RS - Brasil
Fone: (51) 3320-3500 - Fax: (51) 3339-1564
E-mail: prograd@pucrs.br
Site: www.pucrs.br