QATAR UNIVERSITY

COLLEGE OF ENGINEERING

IEEE 802.11AX BASED MEDIUM ACCESS DESIGN FOR WIRELESS IOT-BLOCKCHAIN NETWORK

BY

AREZOU ZAMANYZADEH ABYANEH

A Thesis Submitted to

the College of Engineering

in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Electrical Engineering

June 2021

COMMITTEE PAGE

The members of the Committee approve the Thesis of
Arezou Zamanyzadeh Abyaneh defended on 13/04/2021.

Prof. Nizar Zorba
Thesis/Dissertation Supervisor

Prof. Tamer Khattab
Committee Member

Dr. Mohamed Abdallah
Committee Member

Approved:

Khalid Kamal Naji, Dean, College of Engineering

ABSTRACT

ZAMANYZADEH ABYANEH, AREZOU, M., Masters : June : [2021], Masters of Science in Electrical Engineering

Title:_IEEE 802.11ax based Medium Access Design for Wireless IoT-Blockchain Networks

Supervisor of Thesis: Nizar, Zorba.

Blockchain has emerged as a potential solution to security concerns over decentralized networks, and communication is the basic essence of the blockchain network and must be carefully planned while integrating with Internet of Things (IoT). In this work, blockchain nodes are assumed to use wireless channels to communicate among themselves and IoT elements. This work will propose a Medium Access Control (MAC) mechanism for wireless IoT-blockchain system, while addressing transmission latency and throughput. The proposed MAC protocol is based on the widely used IEEE 802.11ax protocol, Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) and works based on M/G/1 queuing model. Blockchain promises higher security in IoT as it exploits all the nodes in the network to verify a new transaction, thus more contributing nodes indicate higher security. This eventually leads to a larger total delay and less throughput as observed in our results, suggesting a tradeoff on the blockchain usage that must be carefully optimized in practical systems.

DEDICATION

To My late Grandfather,

You are the reason I finished this.

ACKNOWLEDGMENTS

I thank God for all the opportunities and trials and most importantly for giving me the strength and courage to complete this thesis.

First and foremost, I offer my sincerest gratitude to my supervisor, Prof. Nizar Zorba. I cannot say thank you enough for his tremendous support and help. Without his encouragement and understanding when the times got rough and guidance from day one, this thesis would not have materialized. It has been a great honor to be one of his students and one simply could not wish for a better supervisor.

In my daily work I have been blessed with a wonderful family and cheerful group of friends, I thank every single one of them for their inseparable support and prayers. My parents, my brother and my uncle deserve special mention for their continuous support not only in past years, but my whole life!

Finally, I would like to thank everybody who was important to the successful realization of this thesis.

TABLE OF CONTENTS

DEDICATION iv

ACKNOWLEDGMENTS v

LIST OF TABLES viii

LIST OF FIGURES ix

Chapter 1: Introduction 1

1.1 Motivation 1

1.2 Thesis Objectives 3

1.3 Thesis Scope 4

1.4 Thesis Publication 4

1.5 Thesis Outline 5

Chapter 2: BACKGROUND 5

2.1 IoT-blockchain 5

2.2 IEEE 802.11 standard 9

2.3 Literature review 13

Chapter 3: Proposed system 17

3.1 Wireless IoT-blockchain System Model 17

3.2 MAC for Wireless IoT-blockchain 19

3.2.1 Transmission model 19

3.2.2 Queue model 25

Chapter 4: SIMULATION, Results, and Discussion 30

4.1 Simulation 30

4.1.1 MAC transmission 30

4.1.2 M/G/1 Queue model 32

4.2 Results and Discussion 34

Chapter 5: ConclusionS And Future work 39

5.1 Conclusions 39

5.2 Future work 40

References 41

Appendix A: Tranmission model Matlab code 49

Appendix B: Queuing model Matlab code 57

LIST OF TABLES

Table 1. Simulation Parameters 31

Table 2. Input parameters for blockchain-Ethereum from Blocksim 32

Table 3. M/G/1 Utilization results 38

LIST OF FIGURES

Figure 1. A visual explanation of how blockchain works 8

Figure 2. Basic access mode of DCF..12

Figure 3. RTS/CTS access mode of DCF 13

Figure 4. Interaction among user node and full nodes..19

Figure 5. The timeline of the proposed MAC protocol 21

Figure 6. M/G/1 waiting system 26

Figure 7. Consideration for the average waiting time...27

Figure 8. Residual service time process..28

Figure 9. Saturation Throughput for a variable number of nodes...............................34

Figure 10. Total average delay for a variable number of nodes 37

Figure 11. Mean system delay for a variable number of nodes..................................38

CHAPTER 1: INTRODUCTION

1.1 Motivation

Since its creation in 2008 by Nakamoto, blockchain technology has attracted interests from both industry and academia [1]. Although blockchain was first introduced in Bitcoin and financial applications, the key characteristic of blockchain network exploits the use of its technology beyond cryptocurrencies. Blockchain features such as decentralization, persistency, audibility, and immutability make it a technology that not only can solve the issues of conventional centralized Internet of Things (IoT) [2], but also brings new potential to the application [3].

Given that most of the IoT devices are generally resource-constrained, low-power and low memory capable, integrating blockchain and IoT faces several readily apparent challenges which are scalability, resource utilization, privacy, and predictability, these must be carefully accounted when designing blockchain for IoTs. Along this thesis, IoT can be considered as a set of Wireless Sensor Networks (WSN) units capable of sensing, actuating, and communicating over internet leveraging IEEE 802.11ax. Blockchain removes any need of third-party involvement in such a system and enables a peer-to-peer connection. Secure communication and tamper-proof storage are the two major solutions that blockchain offers to IoT and Edge Computing [4]. Communication is needed for recording transactions between two parties, coordination, and data sharing. Blockchain then stores the recorded data in distributed ledgers, of which every trusted party can access using the cryptographic keys. Communication plays an essential role in the performance of the peer-to-peer network and specifically for consensus protocol, which is one of the determining factors for adding new transaction and block to the blockchain [5]. Notice that in order to add a new transaction to a block, and chain that block to the main blockchain, the system needs to verify the integrity of the data included in that transaction. This process is executed using consensus mechanism and smart contracts among nodes [3]. Different consensus mechanisms [5] define how the connections between various IoT and blockchain nodes are implemented and provide a variety of choices for architecture and network design.

The original blockchain was designed using wired communication system and working under the assumption that the communication does not have a major effect on the systems performance [6]. This assumption is not suitable or valid for the wireless IoT context, where the channel access and the connection between nodes contribute significantly to the overall system performance, and they must be tackled when designing blockchain for wireless IoT networks.

One of the challenges discussed in the literature is latency, which has many sources in blockchain itself [6]. Combining it with latency in conventional IoT may result in a significant delay and might stance as a bottleneck on the system performance [5]. Taking into account the communication amongst peer nodes happening over wireless channels, a proper Medium Access Control (MAC) design for the blockchain system over wireless IoT can address this issue, while considering metrics like throughput and delay [7] for its evaluation.

We consider a critical infrastructure as the scenario for the application of our proposals. More specifically, a football stadium or a mall can be suitable as examples. The idea is that there are several sensors inside the infrastructure, and those sensors are very important, as they can indicate fire and therefore trigger evacuations. A malicious manipulation or hacking of such sensors would harm the reputation of any organization. As an example, the World Cup 2022 in Qatar would be very interested in a system that alerts of fires and smoke in stadiums but would be very upset if they evacuate a stadium during WC2022 without the need, because someone hacked the sensors. Therefore, the connection to the sensors needs for high security, and it is where Blockchain can support such setup.

While Blockchain is devoted to its usage on a Worldwide scale and in a distributed manner, our considered scenario is more localized into a small area. Moreover, within our scenario, we have a single point of reception of all sensorial measurements, which we call a leading full node, where all data is centralized. We are able to apply Blockchain to such unique scenario, taking advantage of all its benefits, while optimizing its inclusion in the IEEE 802.11ax standard.

1.2 Thesis Objectives

The integration of blockchain and IoT promises a decentralized, scalable and fault resistant system. Communication and transmission parameters pose a critical role in maintaining the performance and security of such Blockchain-IoT system.

In this research our main objectives are:

- To develop a detailed design and analysis for proposed MAC protocol considering different metrics such as throughput and latency.
- To Implement a simulation platform applying the proposed solution to a collection of heterogeneous decentralized communication and computation nodes in a peer-to-peer network.

Research questions we are interested in this context that will be further explored in this thesis are:

- How is the devices' access to the channel in Wireless IoT-blockchain system?
- How does integration of blockchain in IoT affect a wireless system in terms of MAC throughput and delay?
- What is the trade-off between delay and Security?

1.3 Thesis Scope

In this work, we consider multiple participants in the network have peer-to-peer interaction, and without the need for a trusted third party. This model works as a private-permissionless platform where all the participants are allowed to access the chain, and only certain nodes have the authority to add and write to the chain having network consensus. Therefore, decentralized ledger performing under Ethereum [3] is considered. Ethereum uses a hybrid Proof of Work (PoW) and Proof of Stake (PoS) mechanism, which is considered a more efficient substitution for PoW to implement on resource-constraint and delay-intolerant IoT edge devices [8]; whereas the blockchain network utilizes the CSMA/CA protocol for communication among nodes [9]. The IoT network can be considered as a set of WSN units capable of sensing, actuating, and communicating over internet leveraging the IEEE 802.11ax standard [10] and [11]. IEEE 802.11ax proposes new usage schemes for different aggregation mechanisms, better schedulers and the optimized setting of the MAC techniques. This ultimately provides higher throughput by better utilizing the already available features of IEEE 802.11 standard [12]. Therefore, the aim of this work is to propose a novel MAC design based on IEEE 802.11ax standard working on M/G/1 queuing system for wireless IoT-blockchain network and analyze how different elements from IoT and blockchain contribute to the overall system performance, mainly in terms of delay and throughput.

1.4 Thesis Publication

This thesis is based on the previously published papers listed below. I have permission from my co-authors to use my work(s) listed below in my dissertation/thesis.

1. A. Z. Abyaneh, N. Zorba and B. Hamdaoui, "IEEE 802.11ax based Medium

Access Design for Wireless IoT-Blockchain Networks," *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Taipei, Taiwan, 2020.

2. A. Z. Abyaneh, N. Zorba, "IEEE 802.11 based Medium Access Design for Wireless IoT-Blockchain Networks", *Qatar University Annual Research Forum and Exhibition 2020*, 2020.

1.5 Thesis Outline

The rest of this document is organized as follows:

- **Chapter 2**: Provides a background of Blockchain and MAC protocols in IEEE 802.11 standard, in addition to a detailed literature review.
- **Chapter 3**: The system model for wireless IoT-blockchain is presented. The proposed IEEE 802.11ax based MAC protocol is defined for the system with required parameters.
- **Chapter 4**: Performance of the proposed MAC protocol is evaluated based on different metrics and the results are thoroughly discussed.
- **Chapter 5**: The paper is concluded, and some potential future works are stated.

CHAPTER 2: BACKGROUND

This chapter introduces the main concepts used in this work. It starts with a background about IoT-blockchain and IEEE 802.11, and continues with explaining the principles of each system and the different protocols defined in them. Furthermore, it studies the previous work done in the area of using Blockchain in IoT specifically the ones concerning MAC designs.

2.1 IoT-blockchain

The IoT technology is an emerging paradigm that is applied to several aspect of our life: buildings, automotive, manufacturing, etc. There is considerable interest from both academia and industrial in IoT as it allows for smarter, more profitable, and persistent data collection and communication [13]. Such connection and communication require a safe and trusted framework to ensure the security of transfer and storage of data. Current implementation of IoT is based on a centralized, client-server access method which benefits the network by simpler and faster processing of data [14]. However, this benefit comes at the cost of low security and privacy issues. As a third-party has complete access to the data, together with the lack of authentication, that would ultimately increase the risk of malicious attacks [15].

A potential solution is the integration of blockchain into the IoT application domain. Blockchain has the potential to provide a decentralized, secure, auditable and anonymous framework for IoT. Decentralized nature of blockchain removes the control of single entity over IoT data and also implies scalability [13].

The convergence of IoT and blockchain technology offers solutions to significant challenges of the IoT platform by providing the following key advantages [16]:

- *Anonymity.* IoT nodes can communicate with each other and the blockchain, with the use of cryptographic keys, would give the option of hiding their identity and location.
- *Transparency.* Although as desired, the true identity of a IoT entity might be hidden, but each transaction and information present in the system has to be checked, audited, and traced from the source. This feature ensures the transparency and helps build a trust in the system.
- *Decentralization.* This is critical in IoT, as third-party involvement translate into a performance bottleneck for the system. This is true for data communication and storage. In blockchain, validation through smart contract and consensus protocol ensures the data consistency.

- *Security*. Blockchain certifies the data entering the network is valid and once the validity is confirmed, it is added to a temper-proof ledger. In other words, it is nearly impossible to delete or modify a transaction once added to the blockchain.

Blockchain can be defined as a data structure where sets of blocks containing transactions are stored and chained to one another in sequential order. Each block contains a hash to the previous block that goes to the origin of genesis block. Genesis block is the first block in the chain and is usually hardcoded into the software. All members in the blockchain network have the same amount of power and resources and they are called nodes. Some of these nodes act as miners or verifiers that decide if the new information coming to the network is valid. They are also responsible for linking the blocks together and hashing blocks and verify any change request. This means a transaction does not belong to the chain until it is processed and accepted by all verifier nodes. Miner or verifier nodes require high computational capability and consume high percentage of storage and power [13].

As shown in Fig.1, the process starts by a user in the network requesting a new transaction. This transaction is broadcasted to all the nodes in the network. Once it is proven valid and safe, the transaction is combined with other transactions to create a new block of data for the distributed ledger. This block is then integrated with the full blockchain network which is temper-proof and permanent, and the transaction is declared complete.

In blockchain the transactions are confirmed by reaching consensus among all the participating nodes in the network. Although the rules of a consensus protocol might vary between different types of blockchain, they all aim to keep the network valid and safe. Consensus protocol makes sure the transactions are validated and are in the correct order, the information within blocks are correct and matches the ledger, and also in some cases are responsible for rewarding the miners or validators for their computational efforts. Various blockchains might use different protocols and algorithm [3].

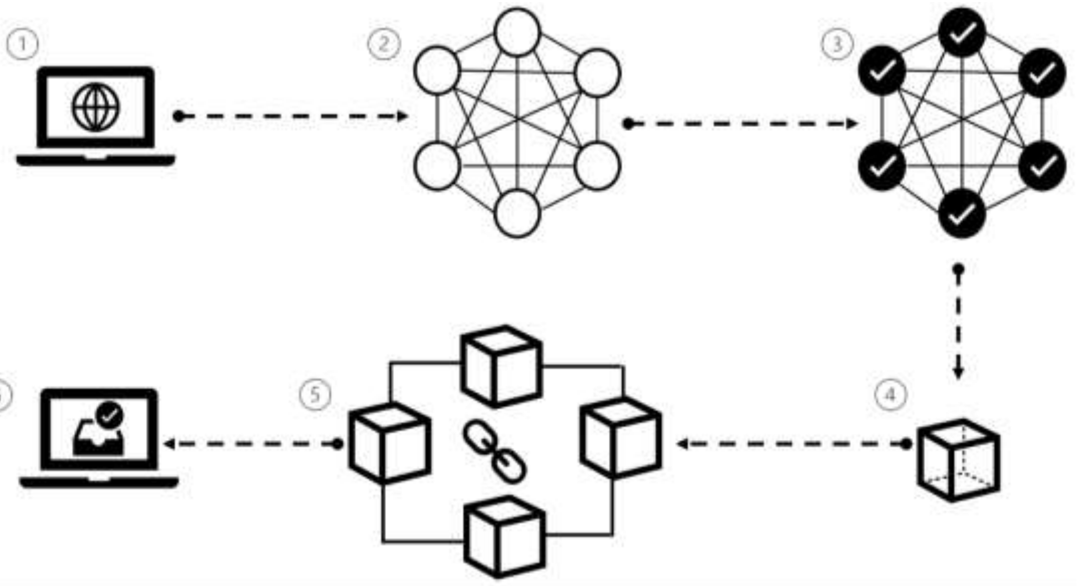

Figure 1. A visual explanation of how blockchain works

The most used consensus method is the Proof of Work (PoW) that is the main techniques behind Bitcoin [1]. Every transaction arriving at the system within a time period of circa ten minutes forms a block and gets validated by trying to solve a PoW puzzle. The puzzles have different level of difficulty and a solution given by a miner to be valid, it must hash to a value less than the current target created by the core algorithm. With the condition that a solution or a block is verified, the miner or verifier who satisfied this condition and broadcasted the block will get rewarded. PoW is generally a safe, secure, and stable scheme, however, mostly requires high computational effort and it is costly and time-consuming to solve [17].

One promising alternative to PoW is Proof of Stake (PoS) consensus scheme [17]. In PoS instead of solving a computationally intensive puzzle, the deciding factor is the nodes deposit size or stake. PoS relies on the amount of time that the miner holds a certain amount of currency which is defined as coin-age. In PoS, the staking nodes or verifiers report to block proposer or leader, which is selected based by the process of locking capital in protocol currency. In this consensus algorithm usually the vote of 2/3 of participating nodes is sufficient to decide on approval or disapproval of a transaction or block [17]. Some of blockchain protocols work on hybrid versions of both PoS and PoW to gain the best of both worlds as it is simpler to implement and the drawbacks of each individual scheme can be eliminated [18].

2.2 IEEE 802.11 standard

The initial version, or so-called IEEE 802.11-1997 was released in 1997 and then many amendments were issued after [19]. The IEEE 802.11 standard offers development of MAC and Physical layer (PHY) specifications for wireless network within local area. The main aim of this standard is to make possible the wireless connectivity for fixed, portable, and moving stations.

Over the years, a series of IEEE standard based on CSMA/CA has been released. The legacy 2 Mbps was the nominal data rate of IEEE 802.11-1997 however this is typically not fast enough and optimal for modern application. The evolution of the standard shows a significant rise in data rate. To support higher throughput, a physical layer changes were specified but the MAC layer was mainly left unchanged. In 802.11b, a physical layer with 2.4 GHz radio band that supported maximum of 11 Mbps was introduced [20]. Another physical layer was specified in 802.11a, where the radio band is extended to the 5GHz band and the maximum physical layer data rate is 54 Mbps per channel [21]. 802.11g defined a new physical layer standard for

WLANs in the same band as 802.11b, 2.4 GHz which make them compatible with each other. The maximum supported physical layer rate in 802.11g is 54 Mbps [22]. Later 802.11a/b/g have improved the data rate by introducing higher modulation and coding schemes, increasing radio band, and the implementation of Multiple Input Multiple Output (MIMO) technologies [23].

802.11n was released to offer 600 Mbps as nominal data rate [24]. This was achieved by deploying combination of techniques including channel width of 40 MHz and increasing the coding rate to 5/6 rather than previously used 3/4 coding rate. 802.11n also introduced some changes in MAC layer with the purpose of reducing overhead size [24]. IEEE 802.11e introduced a MAC layer extension to the 802.11 standard for Quality of Service (QoS) provision, which is back compatible with the physical layers mentioned above, i.e., 802.11a/b/g [25]. The improvement in MAC was introduced by defining Enhanced Distributed Channel Access (EDCA) and Hybrid Controlled Channel Access (HCCA) which differentiate between voice, video, best effort, and background traffic and serve them accordingly [26]. The 802.11ac amendment was released with the sole purpose of increasing the physical rate to 10 times larger than 802.11n. As analysis of 802.11ac proved, further improvement of WLAN throughput requires new channel access rather than old techniques [12]. Similar to the previously mentioned standards that enhance the data rates, 802.11ax introduced a new PHY layer with faster modulation and coding schemes. However, 802.11ax does not widen the radio channel. The higher throughput instead is achieved by more efficient use of spectrum and physical layer link rate goes up to 9.6 Gbps [12]. The major modification of 802.11ax is the adoption of an Orthogonal Frequency-Division Multiplexing Access (OFDMA) method. OFDMA approach has been commonly used in cellular networks, but possibly it is the most important new feature correlated to 802.11ax. This OFDMA is compatible with CSMA/CA mechanism called EDCA or Distributed Coordinating Function (DCF) [12].

IEEE 802.11 MAC regulates channel access using a well-known CSMA/CA protocol. Two modes of operation for CSMA/CA based random access protocols are DCF and Point Coordinating Function (PCF) [27]. We consider the DCF scheme in this thesis, as PCF is obsolete now and not implemented in real devices anymore [12].

In the conventional DCF scheme, the transmission is triggered by a station wishing to send and it starts by first listening to the channel. If the channel is sensed idle for a duration of time equal to distributed inter-frame space (DIFS) then the station starts counting a random backoff. If at any time during the backoff counter the channel is sensed busy the counter is paused and resumes when the channel is sensed idle again. The purpose of random backoff before sending data packet is to mitigate collision. After each successful transmission, the counter resets and if a transmission is lost the counter is doubled [28].

Inter-frame space (IFS) designates the time duration between the packet frames and deciding factor for which IFS to use is determined by type of frame. Some of the IFS implemented in DCF are as following [29]:

- Short Inter-frame Space (SIFS):

This is the shortest inter-frame as is required by a wireless station between receiving a frame and responding to it. Responding frame are Request To Send / Clear To Send (RTS/CTS) or Acknowledgment (ACK) frames. SIFS has the highest priority to ensures an existing transmission is completed.

- DCF Inter-frame Space (DCF):

In conventional DCF, the time interval that a station should wait before it sends its RTS is known as DCF Interframe Spacing (DIFS).

- Extended Interframe Space (EIFS):

This is the longest IFS and is used in case of detecting a corrupted frame.

- Arbitrary Inter-frame Space (AIFS):

Arbitrary inter-frame spacing, the stations are prioritized based on the type of data and access category. Higher priority data types are designated shorter AIFS. This plays a crucial role in scheduling packet transmissions and total delay [12].

DCF has two access mechanisms; basic methods and RTS/CTS method which employs CSMA/CA along with four-way handshaking access [28].

Basic access method uses a two-way handshaking protocol between data and ACK frame, as shown in Fig. 2. RTS/CTS method employs CSMA/CA along with a four-way handshaking mechanism including RTS-CTS-DATA-ACK, as shown in Fig.3. RTS/CTS method is generally applied when the size of the transmitting frame exceeds a certain threshold [28].

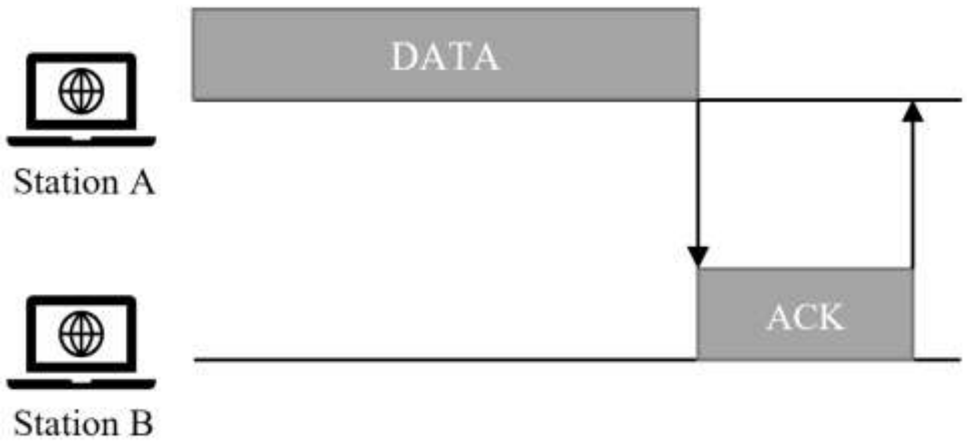

Figure 2. Basic access mode of DCF

In 802.11ax networks, OFDMA is employed on top of the legacy CSMA/CA access mechanisms EDCA or DCF. That implies, the transmitting station has to contend for the channel with other stations in order to send a trigger frame to allocate resources for the associated stations [28].

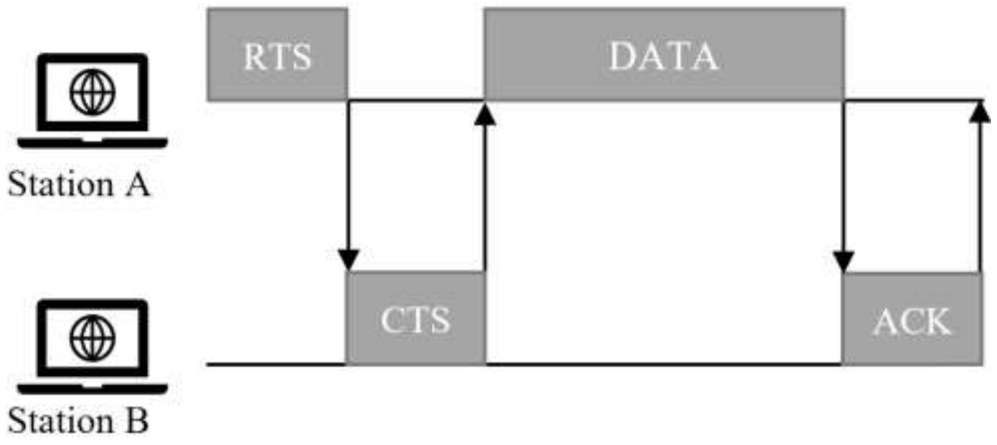

Figure 3. RTS/CTS access mode of DCF

2.3 Literature review

The IoT envisions a completely connected world, where things are able to communicate and exchange data with each other. Many smart applications in a variety of industries are possible now because of this vision. Some of these are smart homes [30], smart cities [31], healthcare [32], etc. The emerging blockchain protocols provide a decentralized domain that is suitable of supporting IoT interactions. In the last decade, there has been studies of convergence of blockchain and IoT, e.g., Internet of Vehicles, Internet of Energy [3], smart homes [33], smart city [34], etc. As has been previously reported in the literature, there are several research issues and challenges in the field of integrating these two technologies [8].

Beside blockchain, there are other distributed ledger technologies which are considered to be an alternative for blockchain specifically in IoT application, namely IOTA technology. IOTA changes the structure of conventional distributed ledger form and has different consensus protocols than blockchain. Given these modifications, it is assumed that IOTA will have some advantages over blockchain specially in terms of transaction fee and scalability. However, up until the date of this work, IOTA still has some limitations and requires more secure and private communication link infrastructure and clearer implementation of authorization and data access control [35].

Implementation of blockchain on lightweight, low-power and memory-constraints IoT devices is challenging. Previous studies have covered different challenges in access mechanism [16], security and thread models [36], activity privacy [37], availability and accountability [38], etc. One important rising problem would be related to communication costs and tradeoffs present. Although this issue is critical and is considered as deciding factor, A closer look to the literature, however, reveals few previous studies in this field. This section explores some of the work done in this area.

Communication traffic for blockchain data synchronization of IoT devices is analyzed in [39]. Further, costs of communication among lightweight IoT devices and a set of blockchain nodes is investigated in [6]. The proposed network considers connection between IoT blockchain nodes via a wireless base station. This work focuses on the radio link layer aggregation of data in this system and proposes a periodic update on blockchain nodes instead of constant synchronization. The proposed aggregation scheme works based on the channel quality, the offered rate, bandwidth requirements, and the statistics of updates of the useful data structures. The scheme proposed in this paper aggregates the information in order to reduce the communication cost and it worked under the assumption that the application domain is delay-tolerant. Blockchain nodes store the information then periodically transmit them to the subsequent aggregation point. The approach works on Proof of Inclusion scheme and blockchain nodes are required to send a proof by means of the state tree, and upon approval are allowed to replace the local copy of the client. In this situations

IoT clients only receive the block header which is the minimum amount of information required. Although the authors in [6] claim the proposed aggregation scheme can be applied to any account-based blockchain, they focus on Ethereum protocol. Given this system model, the protocol from communication costs and security level was evaluated. The result shows a lower communication cost, at the expense of higher information delay, or availability of information, at the application layer. That being said, the work presented in [6], focuses on synchronization delay and failed to consider the propagation of block to the rest of the network, to keep the database replications consistent and have a fully synchronized network.

As mentioned previously, a crucial feature of blockchain and consensus mechanism is broadcasting new transaction and blocks. In case of wireless link, this transmission can be affected by the MAC protocols. A MAC design is the proper mean to organize the devices' access to the channel, and to enable their implementation in commercial systems.

The topic of blockchain-enabled wireless IoT has also been explored further in [40], motivated by the common assumption that in practical blockchain, the communication between nodes is without any channel, throughput, and latency constraints. However, in order to have proper design of a IoT-blockchain system, channel quality and conditions, different interferences and network topology should be considered. Taking these factors into account, the impact of wireless communication on the overall IoT-blockchain system is studied. Two key parameters in blockchain: blockchain transaction throughput and communication throughput. The authors further analyzed the perfomance of the system by spatio-temporal domain Poisson point process (PPP) modeling and signal to interference plus noise ratio (SINR) metric. Also, the work is expanded in terms of security analysis and studied the effect of three attacks (eclipse attack, random link attack, and random node attack) on the proposed system. Using the simulation analysis, the design is valid under these attacks and performs satisfactorily.

The effect of traditional CSMA/CA on the performance of Wireless Blockchain Network (WBN) is tackled in [41]. In this work, authors studied different key perfomance parameters of WBN such as transaction validation delay, transaction per second and transaction loss probability. In order to compare to conventional protocol and highlight the effect of communication delay in the system, they studied the aforementioned parameters with and without the contribution of CSMA/CA latency. Their design considers the effect of queuing and transmission delay while Direct Acyclic Graph (DAG) is assumed as its blockchain architecture. As a conclusion, they claimed the perfomance of DAG-based blockchain is constrained when considering the contribution of wireless network due to limited transmission capability. They further investigate the role of CSMA/CA om system security and showed this limited transmission capability can have crucial impact on security, since computational power of nodes are limited, and this has a direct impact on consensus method.

A new Block Access Control (BAC) approach to organize block mining and transmitting with the aim of addressing the forking issue is presented in [42]. Authors do an analysis of transaction throughput in WBN working on BAC to highlight the effect of CSMA/CA on wireless blockchain performance. By leveraging Markov chain modelling high transaction throughput was achieved while addressing forking and double spending problems that come with involvement of blockchain.

CHAPTER 3: PROPOSED SYSTEM

In this section the proposed Wireless IoT-blockchain System Model, together with MAC and queue designs for the system are explained and justified. In order to investigate the perfomance of the system, throughput and delays have been formulated.

3.1 Wireless IoT-blockchain System Model

Blockchain and IoT offer several opportunities and solutions that complement each other and they both can bring various advantages to the scenario. In order to achieve a successful IoT-blockchain integration there is a necessity to form connections among various entities and devices contributing to the network.

The blockchain network relies upon a peer-to-peer approach for propagation mechanism, where nodes transfer transactions to their neighbor/s. In this work, we consider two types of nodes: Full nodes and User nodes. User nodes are resource-constrained IoT devices that generate raw data, referring to as the unsigned and unverified transactions. Such basic nodes usually do not have enough capability to be integrated with blockchain functions. The sensory data are formed into transaction packets and are wirelessly transmitted to the full nodes, which on the other hand have enough power and capacity to validate and place the transaction into the blockchain ledger. Such use case stems from heterogeneous systems, with a variety of IoT devices, ranging from low to high complexity and processing capabilities. Such formation of clusters would comprehend all kind of devices in practical scenarios.

Consensus protocol and smart contracts are two major players for determining how the new transactions can form a block, and then be added to the blockchain. This work employs hybrid PoW/PoS-based blockchain as the base consensus mechanism for Ethereum. Such consensus mechanism allows for a single chain in the entire system even the threat of forking exists [3]. Forks are called the situations where there are multiple copies of the blockchain state and that means the full nodes are no longer able to reach a unique consensus, as there are multiple views of the network [3]. Forking is mainly due to access rates and delays, and it is a serious issue in blockchain network security. Forking analysis and solutions are out of the scope of this thesis.

The main process of converting a raw data that originated from a user node (an IoT device) into a confirmed block through a full node (blockchain node) is denoted as follows [43]:

(i) A user node acquires new unsigned transactions.

(ii) User node sends the transaction to a leading full node from blockchain network. The leading full node will be designated following Ethereum protocol.

(iii) The leading full node claims the transactions and uses a private key to sign them.

(iv) The leading full node broadcasts the transactions employing CSMA/CA to all blockchain nodes to confirm their legitimacy.

(v) If the majority of the full nodes validate the transactions following the hybrid PoW/PoS consensus, the transactions will then be added to a block, and if not, they will be declined. In this work we focus on the initial validation process since it takes higher priority when defining the security of the full system [14].

Fig. 4 presents the architecture of communication among user nodes and full nodes and indicates the leading full nodes. The time taken to reach a consensus mainly depends on transmission delay throughout communication, in addition to the verification delay form the blockchain core function. By definition, each full node will independently participate in the validation process of a transaction. In blockchain, a greater number of nodes available to verify the transaction leads to a more secure blockchain. However, at the same time, the consensus and verification progress have to be executed for larger number of nodes, which translates into more delay and less throughput. This trade-off is a major concern considered when shifting from a centralized architecture to a peer-to-peer based network. Its characterization and analysis, showing the effect of all parameters in it, are two main objectives of this thesis.

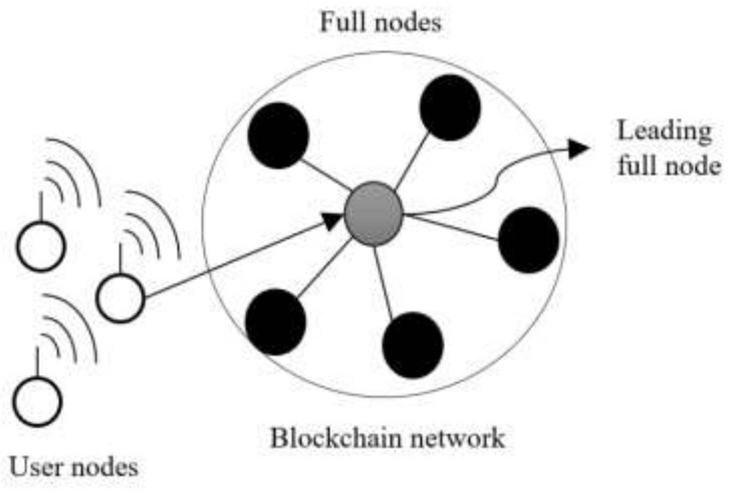

Figure 4. Interaction among user node and full nodes

3.2 MAC for Wireless IoT-blockchain

3.2.1 Transmission model

In order to analyze how the framework of broadcast mechanism works in wireless IoT-blockchain environment (the previously mentioned step (iv)) and its effect on the overall performance, a reliable broadcast transmission compatible with IEEE 802.11ax protocol is proposed. The primary MAC technique of IEEE 802.11 is the DCF, which uses CSMA/CA scheme as its core. CSMA/CA is a contention-based protocol, which listens to the channel before starting the transmission to avoid collision. The basic access mechanism defined in [28] and, with corresponding settings for the wireless network, was adopted as a primary benchmark for the proposed model. The proposed IEEE 802.11ax MAC protocol works on top of the legacy CSMA/CA and the contention-based channel access mechanism, the EDCA [44].

Fig. 5 illustrates the timeline for proposed protocol showing the broadcast transmission among the leading full node and the rest of full nodes. The setup is controlled by the leading node that has a collection of transactions of which it wishes to add to the blockchain. In this case, it will send the transaction through a wireless channel as a broadcast packet. By definition, in CSMA/CA method, in order to start the transmission, the leading full node has to contend for the channel with other nodes to transmit the trigger frame. The trigger frame is the special control frame for IEEE 802.11ax that allocates the resources, and specifies the common parameters of the upcoming transmission, such as duration and guard interval. In our proposed design all the aforementioned parameters are assumed to be the same for all transmission flows. Due to broadcast capacity constraints in IEEE 802.11ax standard, each node can broadcast a limited number of transactions each time [41]. Therefore, after a Short Interframe Space (SIFS), the channel will be dedicated for sending the designated packet and full nodes starts broadcasting. By definition of IEEE 802.11, the SIFS is utilized for high priority actions, such as transmission of the packet or acknowledgment (ACK) frames, where these transmissions can start once the SIFS has elapsed. Upon successful transmission, and after another SIFS, the full nodes start the validation process using the consensus protocol, and each node sends back ACK frames confirming the validity and safety of the transaction, immediately after another SIFS. The ACK frame in this case consists of the conventional information plus the validation results of the transaction. However, requiring all nodes which have successfully received and confirmed the transaction to reply back would be a serious challenge. This is due to the fact that all nodes will reply back at the same time, resulting in a collision at the leading full node. To solve this issue, adding an extension to the MAC header that allows the nodes to send their confirmation in a sequential order can be a potential approach. Nevertheless, this is not the optimal approach, but it guarantees no collisions in the medium nor confirmation from all full nodes in the blockchain. Optimizing this step will be tackled in future work.

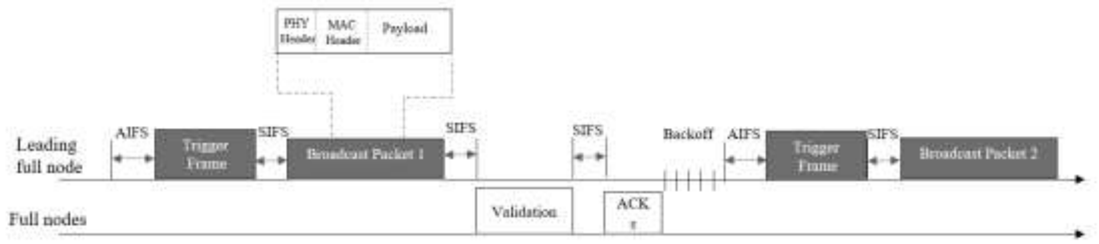

Figure 5. The timeline of the proposed MAC protocol

After successfully receiving all the ACK frames, the leading full node waits for a period of time equal to the Arbitrary Inter Frame Spaces (AIFS), to assure that the channel is idle in order to start a new transmission. For efficiency reasons, a discrete-time backoff is employed before the AIFS period, and then the trigger frame will activate to start the next transmission.

Beside the verification delay discussed in the previous section, there are many delays that are caused by communication and the MAC mechanism. Mainly, they are the carrier-sense delay, backoff delay, transmission delay, propagation delay, processing delay and queuing delay. Following, these delays are investigated within the wireless IoT-blockchain setup context. The carrier sense delay occurs due to the contention window size when the node is sensing if the channel is idle for transmission. Within the proposed design the Bianchi conditions are considered, which entitles having Saturated mode that means there is always a packet ready for broadcast [28]. Queuing delay will be studied in the next section.

Considering the above assumptions, leading full node after waiting in queue can immediately start broadcasting after acquiring the ideal channel. The tackled scenario has a small coverage through the IEEE 802.11ax specifications, and therefore, the propagation delay is insignificant. Furthermore, the processing delay is highly dependent on computational power of devices in addition to the efficiency of the in-network data processing algorithms [45], which are beyond the scope of this thesis. Thus, the total time it takes to broadcast a transaction until the transaction is confirmed and ACK is received consists mainly of carrier-sense delay, back-off delay, and verification delay. Since the leading full node requires a confirmation from all the other full nodes for each transaction packet, the verification process is repeated for each one of the N full nodes in the scenario. Moreover, there will be an ACK packet for each N, which plays a role in the total delay as well.

For the purpose of obtaining the average time that the channel is busy due to a successful transmission in this proposed protocol, some background on DCF and packet transmission probability is required. In Bianchi model, the behavior of a single station using a Markov model is studied and the stationary probability, τ, is obtained [28]. τ is the probability that the station transmits a packet in a randomly chosen generic slot time. This probability depends on back-off time, where DCF adopts an exponential back-off scheme. This means at each transmission, the back-off time is uniformly chosen from the range of $(0, w-1)$. The w defines the contention window and at the first transmission its value is equal to minimum contention window, CW_{min}. In this system, this value is doubled after each transmission until maximum value which is defined as $CW_{max} = 2^m\, CW_{min}$, considering m as maximum back-off stage [45]. In Bianchi model, $w = CW_{min}$. Considering all transmission happen when the back-off counter reaches zero [28], regardless of back-off stage, that gives us the τ as

$$\tau = \frac{2(1-2p)}{(1-2p)(w+1)pw(1-(2p)^m)} \tag{1}$$

And as it can be observed, τ is dependent on the probability that transmitted packets collide, p, which is unknown. However, by definition, p is the probability that a packet encounters a collision in transmission which means, at least one of the $n-1$ remaining stations is transmitting at that time. Taking into account τ as the packet transmission probability [28], p is defined as

$$p = 1 - (1-\tau)^{n-1} \tag{2}$$

One important performance metric for the MAC protocol in the wireless IoT-blockchain environment is throughput. Let Ω be the normalized throughput which is defined as the successfully transmitted radio link frames per time unit. As previously mentioned, we consider Bianchi conditions, therefore when computing throughput, it is assumed to have a fixed number of contending nodes in saturation mode, n. And we analyzed the system throughput considering at least there is one transmission in the chosen time slot with probability P_t given as [28]

$$P_t = 1 - (1-\tau)^n \tag{3}$$

Throughput is highly dependent on the probability of a successfully transmitted packet, P_s , as well as the probability that a transmitted packet collides, P_c. Considering that exactly one station transmits on the channel, P_s can be obtained as [28]

$$P_s = \frac{n\tau(1-\tau)^{n-1}}{P_t} = \frac{n\tau(1-\tau)^{n-1}}{1-(1-\tau)^n} \tag{4}$$

And P_c can be defined as

$$P_c = 1 - P_s \tag{5}$$

Knowing that P_s and P_c have a direct relation to the contention window size and number of nodes, allowing to express the throughput Ω as the ratio,

$$\Omega = \frac{P_s P_t E[P]}{(1 - P_t)\sigma + P_s P_t T_s + P_t P_c T_c} \tag{6}$$

where E[P] is average broadcast packet length containing transactions, data originated from user node. Therefore, the average payload size successfully transmitted in a slot time is given by $P_s P_t E[P]$, with $P_s P_t$ as the probability of a successfully transmitted packet in a slot time. In Eqn. (6), $1 - P_t$ represents the probability of a time slot being empty, and $P_t P_c$ shows the probability that there is a collision in the chosen time slot.

Moreover in Eqn. (6), T_s is average time that the channel is captured with a successful transmission, and T_c is average time that the channel is captured by stations which collide. σ is the duration of an empty contention time slot during transmission. In calculation of throughput, $E[P]$, T_s, T_c and σ are expressed with the same unit.

Considering the proposed system of wireless IoT-blockchain is fully managed by basic access mechanism, the T_s can be obtained as

$$\begin{aligned} T_s = AIFS + E[TF] + SIFS + E[P] + PHY_{header} + MAC_{header} + \\ + SIFS + N_f T_b + SIFS + N_f ACKs + \delta \end{aligned} \tag{7}$$

where $E[TF]$ is defined as the trigger frame size and δ denotes the free-space propagation delay. ACKs are the acknowledgment frames containing verification message as well. PHY_{header} and MAC_{header} are the size of the data fields added at the beginning of a payload packet in order to turn it into a transmission. The delay experienced throughout the verification process, T_b, can be expressed as

$$T_b = \frac{E[P]}{E[T]} T_v \tag{8}$$

where $E[T]$ is the average transaction length defined in the blockchain protocol, and T_v is the average time taken by each node to verify a transaction. N_t signifies total number of full nodes and N_f is defined as total number of full nodes $-$ leading full node. Moreover, if there is a collision occurring in the transmission process the delay experienced, T_c , can be expressed as

$$T_c = AIFS + E[TF] + SIFS + E[P] + PHY_{header} + MAC_{header} + \delta \qquad (9)$$

Other than the time for the transmission itself, network experiences another delay defined as backoff delay, T_{Bo}, the time that a station chooses to wait before accessing the channel under busy channel.

$$T_{Bo} = (1 - P_t)\sigma + P_s P_t T_s + P_t P_c T_c \left(\frac{w-1}{2}\right) \qquad (10)$$

Summing all the delays, the total delay experienced throughout the MAC process, is denoted as T_D and can be represented as

$$T_D = T_{Bo} + T_s + T_C \qquad (11)$$

It is important to highlight that the number of nodes plays a significant role in performance of the system, as increasing number of nodes will increase the security of the system. That imply higher number of verifiers to validate the legitimacy of the transactions and as a result a decrease in the probability of a malicious data within the blockchain. However, as observed in Eqn. (6) and Eqn. (7), the throughput and delay are highly affected by the number of nodes.

3.2.2 Queue model

This work is based on M/G/1 queuing system, shown in Fig.6, where the packets containing the transactions follow Poisson arrival process, which means interarrival times are exponentially distributed [47]. In order to have an accurate design, we consider generally distributed service time, which practically means it can be any distribution. This way there we have the option of changing the formulation for service time which, can be done by changing the consensus protocol and properties of blockchain. As explained in previous section, there is one server, leading full node, serving the packets in the system. In this queuing system, it is assumed there are infinite number of waiting packets. The proposed model works in the First In First Out (FIFO) mode [48], which selects the packet at the head of the queue as the next packet to service and the packet enters service immediately. FIFO allocates packets forwarded along the same path to be transmitted in the same order as they arrived at the source. FIFO also provides a sense of fairness because the serviced packet is the one which has been waiting the longer and this may affect the waiting time. Wireless IoT-blockchain network can greatly benefit from this property if we consider the packets all have same priority.

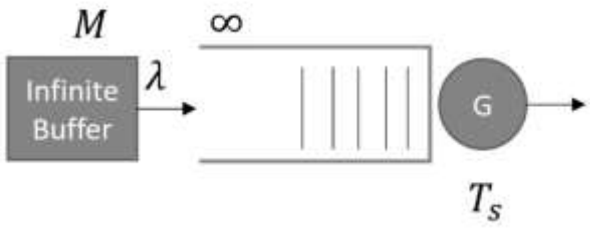

Figure 6. M/G/1 waiting system

In M/G/1 queuing system, the packets arrive according to a Poisson process with rate λ. And the utilization ρ and the traffic intensity can be obtained as

$$\rho = \lambda E[T_s] \tag{12}$$

In order for the system to run under stability condition, the $\rho < 1$ should be ensured.

In this analysis, we want to find average time spent in the queue, $E[T_q]$, and average queue length, $E[N_q]$. Although the complete distribution of these performance parameters can be obtained based on embedded Markov chain, in this thesis we choose the residual life approach since mean values are sufficient for our purpose. Under FIFO conditions, newly arrived packet, shown in Fig. 7, has to wait until the packet that is currently in the transmission is completely transmitted and all the packets before it, in the queue are also served. In this case, the residual service time, $E[R]$, is service time of the customer being served and $E\left[N_q^a\right]$ is the number of packets encountered in queue by arriving packet. Therefore, for average service time for each packet can be formulated as

$$E\left[T_q\right] = E\left[N_q^a\right]E[T_D] + E[R] \tag{13}$$

where $E[T_D]$ is the service time corresponding to the total delay experienced throughout the MAC process, as we obtained in previous section.

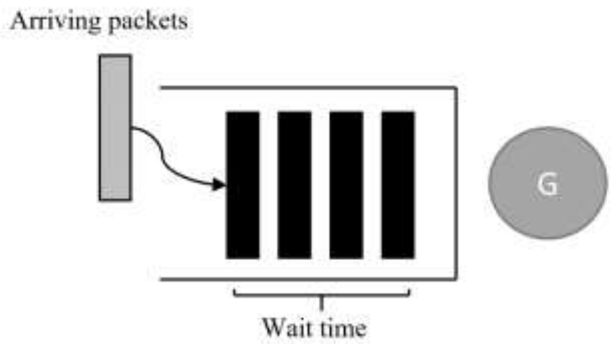

Figure 7. Consideration for the average waiting time

Since The arrival process of the M/G/1 queue is a Poisson process by assumption, the Poisson Arrivals See Time Averages (PASTA) property holds [48]. This means the number of packets in the system seen by the typical arrival has the same distribution as the number of customers in system in steady state, or $E\left[N_q\right] = E\left[N_q^a\right]$. Therefore, applying Little's Law [46] gives,

$$E\left[N_q\right] = \lambda E\left[T_q\right] \tag{14}$$

Using Eqn. (13) and Eqn. (14), we can reformulate it as

$$E[T_q] = \lambda E[N_q]E[T_D] + E[R] = \rho E[T_q] + E[R] = \frac{E[R]}{(1-\rho)} \qquad (15)$$

that depends on the residual service time, $E[R]$. Fig. 8 shows residual service process over time. The average value of the sawtooth curve can be calculated by dividing the sum of the areas of the triangles by the length of the interval.

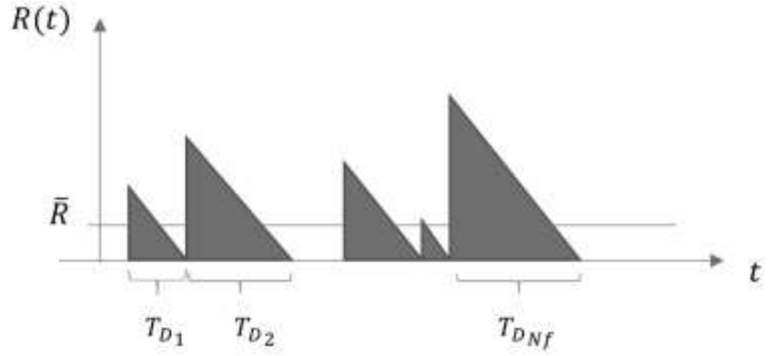

Figure 8. Residual service time process

The number of the triangles, N_f, is determined by the arrival rate λ; so we will have $\lambda t = N_f$ packets arriving. So,

$$E[R] = \frac{1}{t}\int_0^t R(t')dt' = \frac{1}{t}\sum_{i=1}^{Nf}\frac{1}{2}T_{s_i}^2 = \frac{N_f}{t}\frac{1}{N_f}\frac{1}{2}T_{D_i}^2 = \lambda\frac{1}{2}E[T_D^2] \qquad (16)$$

Substituting this in Eqn. (15) will give Pollaczek-Khinchin (PK) mean formula [47] for the waiting time in the queue as,

$$E[T_q] = \frac{E[R]}{1-\rho} = \frac{\lambda E[T_D^2]}{2(1-\rho)} = \frac{1+CV[T_D^2]}{2}\frac{\rho}{(1-\rho)}\,E[T_D] \qquad (17)$$

And from the mean waiting time one immediately gets system time, $E[T_A]$, as

$$\begin{aligned} E[T_A] &= E[T_D] + E[T_q] = E[T_D] + \frac{\lambda E[T_D^2]}{2(1-\rho)} \\ &= 1 + \frac{1+CV[T_D^2]}{2}\frac{\rho}{(1-\rho)}\,E[T_D] \qquad (18) \end{aligned}$$

By applying Little's law to Eqn. (17) and Eqn. (18), we can obtain the corresponding formulae for the average number of packets in the queue, $E\left[N_q\right]$, and in the system, $E[N]$, as follows

$$E\left[N_q\right] = \lambda E\left[T_q\right] = \frac{\lambda^2 E[T_D^2]}{2(1-\rho)} = \frac{1 + CV[T_D^2]}{2} \frac{\rho^2}{(1-\rho)} \tag{19}$$

$$E[N] = \lambda E[T_A] = \frac{\lambda^2 E[T_D^2]}{2(1-\rho)} + \rho = \frac{1 + CV[T_D^2]}{2} \frac{\rho^2}{(1-\rho)} + \rho \tag{20}$$

CHAPTER 4: SIMULATION, RESULTS, AND DISCUSSION

In this section a simulation model is designed and implemented based on the analysis done in chapter 3, that shows the mathematical analysis of the MAC design along with queuing model. The purpose of the simulation model is to evaluate the performance of proposed MAC algorithm and observe throughput and different delays in the IoT-blockchain under different network setups. The main parts of simulations are implemented using MATLAB.

4.1 Simulation

4.1.1 MAC transmission

Simulation parameters following the IEEE 802.11ax and Ethereum standards are listed in Table 1. IEEE 802.11ax setting is based on best-effort access categories and uses the DCF default parameters [12]. The frame sizes are those defined by the 802.11 MAC specifications and the PHY header is that defined for the frequency hopping spread spectrum (FHSS) PHY, while the User Datagram Protocol (UDP) flow is used as the transport layer protocol [29]. The transaction length, which refers to the unsigned and unverified transaction data is obtained from the Ethereum yellow paper [49].

Table 1. Simulation Parameters

Parameters	Value
Number of nodes	5-50
Packet Payload	[64,512,1024] bits
MAC Header	272 bits
PHY Header	128 bits
ACK length	112 bits + PHY header
Transaction Length	64 bits
Air Propagation Delay	1 µs
Min. Contention window	31
Stage Number	3
SIFS	16 µs
AIFS	43 µs
Slot Time	9 µs

As discussed earlier, backoff window has an impact on carrier-sense delay which can directly affect the probability of collision. The other parameters related to blockchain such as transaction verification time were implemented following the values from Blocksim [50] and summarized in Table 2. Blocksim presents a discrete-event simulator for Bitcoin and Ethereum, that allows changing the conditions and analyzing the blockchain performance with different settings. For the targeted purpose of this thesis, the settings, and parameters for the Ethereum model are deployed.

Our designed model is based on an IoT environment, and a private blockchain such that the number of nodes playing a role in the verification process is finite. In simulation model the number of nodes is considered to be 5 to 50 nodes, where the leading full node has a new transaction and is willing to broadcast it to the network following the proposed algorithm for channel access. Retransmissions are not considered in this work.

Table 2. Input parameters for blockchain-Ethereum from Blocksim

	Distribution	Location
Block validation delay	Log-Normal	0.229 s
Transaction validation delay	Log-Normal	0.004 s
Time between blocks	Normal	15.79 s

4.1.2 M/G/1 Queue model

The algorithm for M/G/1 FIFO is shown in Algorithm 1. The queue model has two event process, arrival of new packets and departure of packets. Simulation of FIFO is quite simple since the service time only needs to be generated according to necessary distribution at the point when packet enters service process.

In order to model the M/G/1 queue, we used the MATLAB version of VBASim [51]. MATLAB version of VBASim offers a collection of MATLAB Scripts, Functions and Class M-files that aid in developing discrete-event simulations and in this case, queuing model simulation.

As explained in previous section, we consider Poisson arrival process and a uniformly distributed service time with fixed mean values. There is a single server and although the number of nodes does not have an effect in queuing time, as shown in Eqn. (17), it has a large contribution in service time calculated and simulated in transmission section.

We choose to work the Event-based Simulation of the M/G/1 queue mode in VBASim. In this simulation we follow inter-arrival rate of Ethereum [6], which is assumed to be $\lambda = 0.1\, s^{-1}$. And the simulation and computation values of service time is taken from previous section. This model will provide Average wait time, Average number of packets in queue, Number of packets remaining in queue and server utilization.

Algorithm 1 M/G/1 FIFO

```
Input: $T$ −simulation time, $X_t$ −state of the systme at each time,
$L$ −interarrival time
$S$ −service time //uniform distribution,
$\lambda$ − arrival rate //exponential distribution
01: $X_t \leftarrow 0$
02: while $t < T$ do
03:     if Arrival at time t, then
04:         $X_t \leftarrow X_t + 1$
05:         $t + S$ // departure time of the packet
06:         $t + L$ // arrival time of the next packet
07:     endif
08:     if departure at time t, then
09:         $X_t \leftarrow X_t - 1$
10:         $t + S$
11:     endif
12: end while
```

4.2 Results and Discussion

Fig. 9 shows the total throughput for an increasing number of full nodes on several different payload and number of transactions. The throughput in the simulation is defined as the total packet that get through the channel successfully by the payload size over the simulation time. As expected from the mathematical model mentioned in previous section, with growth in the number of nodes, the throughput decreases. This indicates a lower percentage of successful transmission and lower number of packets are received by the full nodes. Furthermore, increasing number of nodes leads to increasing number of consecutive ideal slots between two broadcast transmissions, causing a decrease in throughput. Additionally, it can be observed that throughput tends to rise with the growth of payload size. Increasing the payload size implies increasing the amount of data to send, which increases throughput.

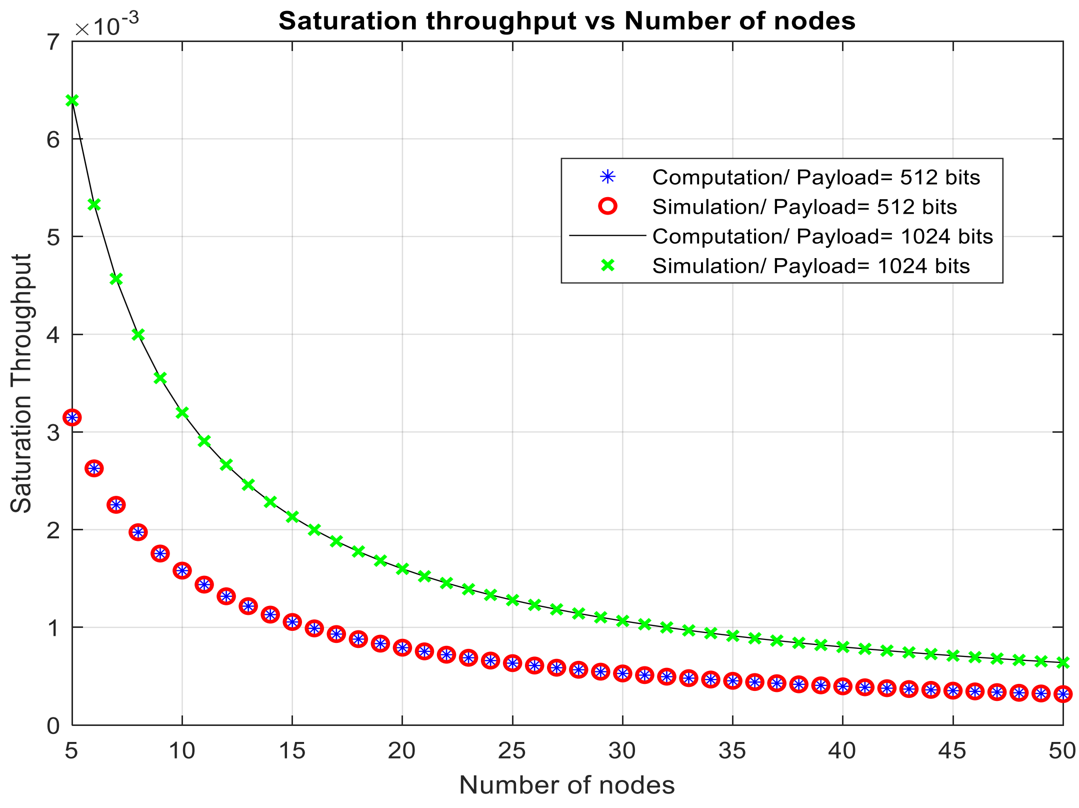

Figure 9. Saturation Throughput for a variable number of nodes

However, this change is not very obvious due to the large verification delay in T_s and consequently contributing to overall simulation time.

Fig. 10 highlights the relationship between the number of full nodes contributing to the network and the time taken for a node to successfully broadcast different number of transactions and receive back the confirmation message. The average delay shown is recorded in microsecond.

It can be noted the impact our proposed protocol made to the network is significant in terms of delay analysis. It can be observed that as the number of nodes increases from 5 to 50, a significant increase in total average delay is introduced, which is mostly due to validation time. This delay is more noticeable in higher number of nodes, that is where the difference between computation and simulation analysis can be observed as well. This difference is mostly due to the effect of backoff counter. As explained in previous section a backoff mechanism is implemented before starting the transmission. In computation, the contention window is equal to minimum value of contention window defined in the model. In simulation since there has to be a backoff counter implemented, the initial backoff time is equal to min. contention window and has uniform backoff at next stages. Note that the simulation stops counting while the channel is sensed busy. On the other hand, as previously mentioned, increasing number of nodes increases the security and immunity of the network to malicious attack. Consequently, a trade-off is presented when a conventional blockchain is integrated with an IoT environment. This trade-off must be addressed carefully while designing wireless IoT-blockchain setup. And since in our system, we did consider 0.004s as the transaction verification delay, we can say that the huge delay is mostly due to blockchain. This is further confirmed by comparing to the conventional IEEE 802.11ax MAC protocol without the blockchain impact to the proposed system. We observed the total throughput decreases by 12% in average when considering the effect of validation time. There has to be further modification to MAC layer design to compensate for this shortcoming. One recommendation to solve this issue is to adjust DCF/EDCA parameters such as contention window. Increasing the window size, tends to decrease the probability of collision and in result gives a higher throughput. However, increasing the contention window is acceptable to some extent due to the possible impact imposed on the delivery delay of the packet.

On the other hand, the average delay is increased by 21% when the validation process delay is included in total transmission delay. Possible solution for this issue can be moving to PoS consensus mechanism entirely and remove the time that is introduced by solving the puzzle in PoW from the verification process. However, this might not come without expenses as it possibly will affect the security of the IoT-blockchain network. From MAC design side, one possible solution would be modifying the acknowledgment process. As in the current model, we assume each node should send back an ACK along with verification message at the end of each transmission.

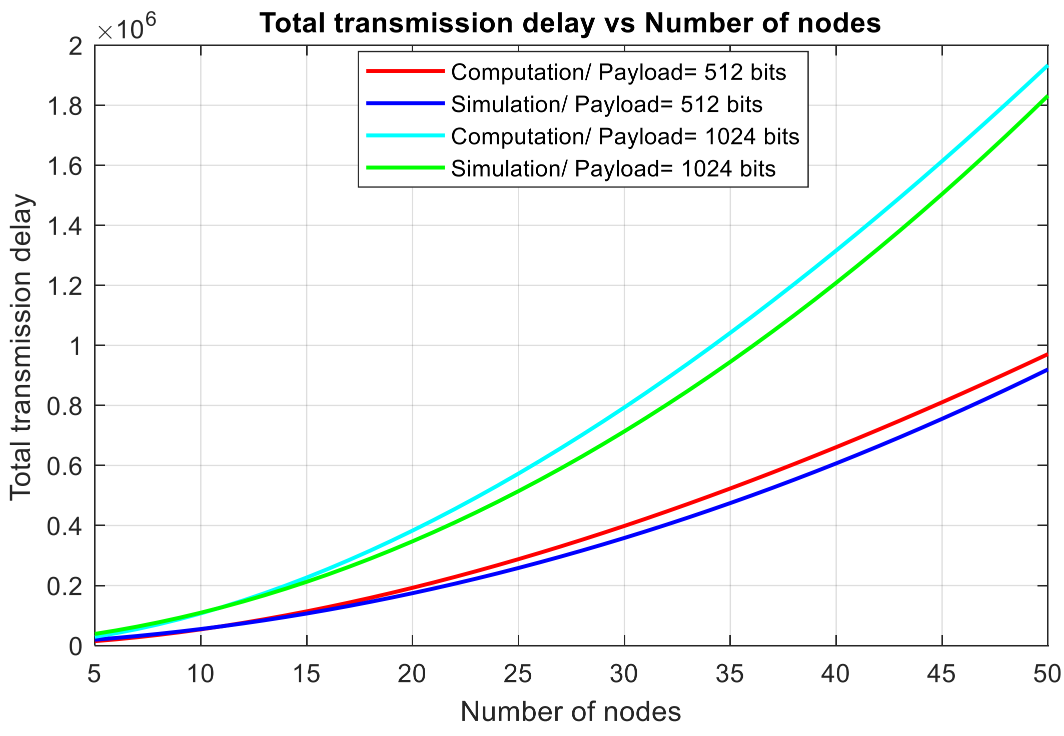

Figure 10. Total average delay for a variable number of nodes

As expected from queue modeling in previous section, as the average service time increases, the average time spent in the queue also increases. Moreover, as mentioned before, service time in this model is the time that the leading full node starts broadcasting the packet until the acknowledgements are successfully received, which is related to the number of nodes or validators. This is verified using simulation and the obtained result is shown in Fig. 11, where the system delay (the sum of queuing delay and transmission delay) is simulated versus different number of nodes. Notice that the system delay increases with increasing number of nodes and following similar behavior, the increase has higher rate with growth of number of validators. Moreover, the utilizations which is a measure for productivity of the server and following queuing theory, the higher the average utilization level, the longer the wait times. Utilization results are summarized in table 3.

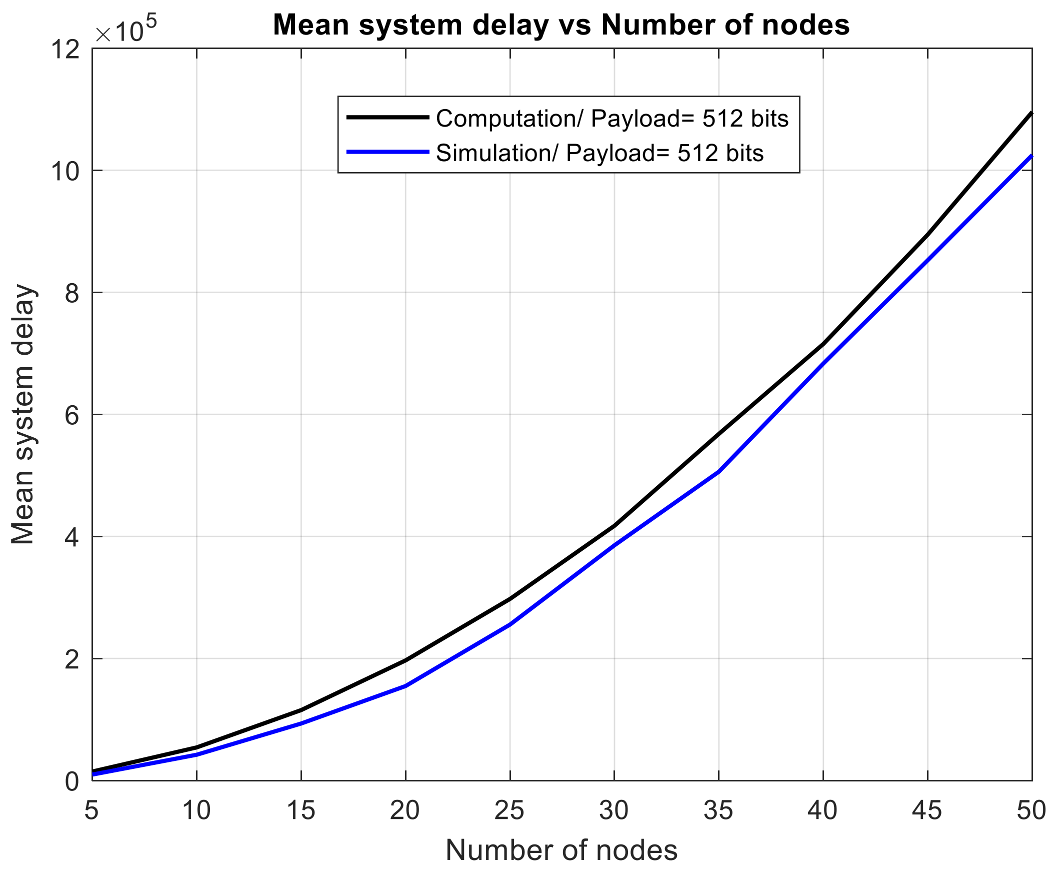

Figure 11. Mean system delay for a variable number of nodes

Table 3. M/G/1 Utilization results

Number of nodes	Simulation	Computation
5	0.001356	0.002451
10	0.005401	0.008986
15	0.012134	0.018993
20	0.021561	0.032098
25	0.033665	0.047993
30	0.048477	0.066448
35	0.065946	0.088942
40	0.086151	0.110222
45	0.10902	0.135159
50	0.134573	0.161935

CHAPTER 5: CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

The integration of blockchain and IoT promises a decentralized, scalable and fault resistant system. However, it is evident to recognize the gap between blockchain and IoT due to some basic characteristics accompanied with a blockchain, such as the need for an intensive computation effort, a high hardware cost, storage capacity and significant power consumption. On the other hand, conventional inexpensive and resource-constraint IoT devices have limited capability for blockchain application. Therefore, a proper design for an efficient integration of blockchain and IoT is required.

In our proposed work, a MAC transmission scheme based on IEEE 802.11ax is modeled aiming to define the transmission among nodes in wireless IoT-blockchain setup. The scheme is based on CSMA/CA basic access mechanism and it is implemented on top of Ethereum-based blockchain. The queuing is considered to work according to M/G/1 FIFO mechanism. The mathematical equations for total delay and throughput were obtained taking into account the effect of blockchain and used to investigate the effect of blockchain on CSMA/CA mechanism. This is achieved by building a MATLAB model in order to simulate the variation of delay and throughput with respect to different number of nodes. Blockchain promises higher security in IoT as it exploits all the nodes in the network to verify a new transaction before adding it to the chain, thus more contributing nodes indicate higher security. This eventually leads to a larger total delay and less throughput as observed from the presented simulation results. It introduces a trade-off that is highly considered as a major concern while integrating IoT network and blockchain technology.

5.2 Future work

This work can be further extended in future to analyze and design a wireless IoT-blockchain network with reliable transmission and communication between nodes. In addition, working on retransmission scenario for such a MAC design is an open research problem, which will be investigated further. Retransmission can occur due to several sources in such systems, which can be viewed from the communication perspective or the blockchain itself.

Another future work could be working on improving the considered parameters; delay, and throughput in order to have a more practical implementation of blockchain for IoT devices that are delay intolerant. In order to make more acceptable or to bring nearer a standard, one recommendation is to move the system to work according to PoS consensus mechanism instead of hybrid PoW/PoS, which will eliminate the puzzle solving time in PoW. Nevertheless, this does not come without any cost, as current implementation of PoS is proven to be less secure than PoW. However, this can be improved with smart contract involvement.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," Bitcoin, 2008.

[2] A. Radwan and H. C. Chi, "Multi-Objective Optimization of Green Small Cell Allocation for IoT Applications in Smart City," *IEEE Access,* vol. 8, pp. 101903 - 101914, 2020.

[3] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access,* vol. 4, pp. 2292 - 2303, 2016.

[4] B. Hamdaoui, N. Zorba and A. Rayes, "Participatory IoT Networks-on-Demand for Safe, Reliable and Responsive Urban Cities," *IEEE Blockchain Technical Briefs,* January 2019.

[5] E. Ragnoli, J. D. Sheehan and E. Ragnoli, "On the Design of Co-operating Blockchains for IoT," in *2020 3rd International Conference on Information and Computer Technologies (ICICT),* San Jose, 2020.

[6] P. Danzi, A. Kalør, Č. Stefanović and P. Popovski, "Delay and Communication Tradeoffs for Blockchain Systems With Lightweight IoT Clients," *IEEE Internet of Things Journal,* vol. 6, no. 2, pp. 2354 - 2365, 2019.

[7] A. A. Brincat, A. Lombardo, G. Morabito and S. Quattropani, "On the use of Blockchain technologies in WiFi networks," *Computer Networks,* vol. 162, 2019.

[8] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras and H. Janicke, "Blockchain Technologies for the Internet of Things: Research Issues and Challenges," *IEEE Internet of Things Journal* , vol. 6, no. 2, pp. 2188 - 2204, 2018.

[9] J. Kang, Z. Xiong, D. Niyato, P. Wang, D. Ye and D. I. Kim, "Incentivizing

Consensus Propagation in Proof-of-Stake Based Consortium Blockchain Networks," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 157 - 160, 2018.

[10] R. Parsamehr, A. Esfahani, G. Mantas, A. Radwan, S. Mumtaz, J. Rodriguez and J.-F. Martínez-Ortega, "A Novel Intrusion Detection and Prevention Scheme for Network Coding-Enabled Mobile Small Cells," *IEEE Transactions on Computational Social Systems* , vol. 6, no. 6, pp. 1467 - 1477, 2019.

[11] M. S. Afaqui, E. Garcia-villegas and E. Lopez-Aguilera, "IEEE 802.11ax: Challenges and Requirements for Future High Efficiency WiFi," *IEEE Wireless Communications* , vol. 24, no. 3, pp. 130 - 137, 2016.

[12] E. Khorov, A. Kiryanov, A. Lyakhov and G. Bianchi, "A Tutorial on IEEE 802.11ax High Efficiency WLANs," *IEEE Communications Surveys & Tutorials,* vol. 21, no. 1, pp. 197 - 216, 2018.

[13] A. Reyna, C. Martín, J. Chen, E. Soler and M. Díaz, "On blockchain and its integration with IoT. Challenges and opportunities," *Future Generation Computer Systems*, vol. 88, pp. 173-190, 2018.

[14] F. Restuccia, S. D'Oro, S. S. Kanhere, T. Melodia and S. K. Das, "Blockchain for the Internet of Things: Present and Future," *IEEE Internet of Things Journal,* vol. 1, no. 1, 2018.

[15] D. Minoli and B. Occhiogrosso, "Blockchain mechanisms for IoT security," *Internet of Things*, Vols. 1-2, pp. 1-13, 2018.

[16] O. Novo, "Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184 - 1195, 2018.

[17] S. Leonardos, D. Reijsbergen and G. Piliouras, "Weighted Voting on the Blockchain: Improving Consensus in Proof of Stake Protocols," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2019.

[18] Y. Huang, Y. Zeng, F. Ye and Y. Yang, "Incentive Assignment in PoW and PoS Hybrid Blockchain in Pervasive Edge Environments," in *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*, 2020.

[19] IEEE 802.11 WG, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Std. 802.11, 1999.

[20] IEEE 802.11 WG, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz," IEEE Std. 802.11b, 1999.

[21] IEEE 802.11 WG, "Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications: high-speed physical layer in the 5 GHz band," IEEE Std. 802a, 1999.

[22] IEEE 802.11 WG, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 4: Further Higher Data Rate Extension in the," IEEE Std. 802.11g, 2003.

[23] R. Karmakar, S. Chattopadhyay and S. Chakraborty, "Impact of IEEE 802.11n/ac PHY/MAC high throughput enhancements on transport and application protocols—A survey," *IEEE Communications Surveys & Tutorials,* vol. 19, no. 4, pp. 2050 - 2091, 2017.

[24] F. M. Abinader, E. Almeida, S. Choudhury, V. Sousa, A. Cavalcante, F. Chaves, E. Tuomaala, R. Vieira and K. Doppler, "Performance Evaluation of IEEE 802.11n WLAN in Dense Deployment Scenarios," in *2014 IEEE 80th Vehicular*

Technology Conference (VTC2014-Fall), 2014.

[25] IEEE 802.11 WG, "Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications: medium access control (MAC) quality of service (QoS)," IEEE 802.11e D8.0, 2004.

[26] D. Gao, J. Cai and K. Ngi Ngan, "Admission control in IEEE 802.11e wireless LANs," *IEEE Network* , vol. 19, no. 4, pp. 6 - 13, 2005.

[27] P. Chatzimisios, V. Vitsas and A. C. Boucouvalas, "Throughput and delay analysis of IEEE 802.11 protocol," in *Proceedings 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications,* Liverpool, 2002.

[28] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications,* vol. 18, no. 3, pp. 535 - 547, 2000.

[29] L. Dai and X. Sun, "A Unified Analysis of IEEE 802.11 DCF Networks: Stability, Throughput, and Delay," *IEEE Transactions on Mobile Computing,* vol. 12, no. 8, pp. 1558 - 1572, 2012.

[30] T. Qiu, N. Chen, K. Li, M. Atiquzzaman and W. Zhao, "How Can Heterogeneous Internet of Things Build Our Future: A Survey," *IEEE Communications Surveys & Tutorials,* vol. 20, no. 3, pp. 2011 - 2027, 2018.

[31] A. Elbery, H. S. Hassanein, N. Zorba and H. A. Rakha, "IoT-Based Crowd Management Framework for Departure Control and Navigation," *IEEE Transactions on Vehicular Technology,* vol. 70, no. 1, pp. 95 - 106, 2020.

[32] L. Catarinucci, D. de Donno, L. Mainetti, L. Palano, L. Patrono, M. Laura Stefanizzi and L. Tarricone, "An IoT-Aware Architecture for Smart Healthcare Systems," *IEEE Internet of Things Journal* , vol. 2, no. 6, pp. 515-526, 2015.

[33] A. Dorri, S. Kanhere, R. Jurdak and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Kona, 2017.

[34] B. Hamdaoui, M. Alkalbani, A. Rayes and N. Zorba, "IoTShare: A Blockchain-Enabled IoT Resource Sharing On-Demand Protocol for Smart City Situation-Awareness Applications," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10548 - 10561, 2020.

[35] R. Nakanishi, Y. Zhang, M. Sasabe and . S. Kasahara, "IOTA-Based Access Control Framework for the Internet of Things," in *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, Paris, 2020.

[36] A. Dorri, S. S. Kanhere and R. Jurdak, "Towards an Optimized BlockChain for IoT," in *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, Pittsburgh, 2017.

[37] A. Dorri, C. Roulin, R. Jurdak and S. S. Kanhere, "On the Activity Privacy of Blockchain for IoT," in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*, Osnabrueck, 2020.

[38] A. Boudguiga, N. Bouzerna, L. Granboulan, A. Olivereau, F. Quesnel, A. Roger and R. Sirdey, "Towards Better Availability and Accountability for IoT Updates by Means of a Blockchain," in *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, Paris, 2017.

[39] P. Danzi, A. E. Kalor, C. Stefanovic and P. Popovski, "Analysis of the Communication Traffic for Blockchain Synchronization of IoT Devices," in *2018*

IEEE International Conference on Communications (ICC), Kansas City, 2018.

[40] Y. Sun, L. Zhang, G. Feng, B. Yang, B. Cao and M. A. Imran, "Blockchain-Enabled Wireless Internet of Things: Performance Analysis and Optimal Communication Node Deployment," *IEEE Internet of Things Journal,* vol. 6, no. 3, pp. 5791 - 5802, 2019.

[41] B. Cao, M. Li, L. Zhang, Y. Li and M. Peng, "How Does CSMA/CA Affect the Performance and Security in Wireless Blockchain Networks," *IEEE Transactions on Industrial Informatics,* vol. 16, no. 6, pp. 4270 - 4280, 2019.

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

[43] A. ZamanyZadeh Abyaneh, N. Zorba and B. Hamdaoui, "IEEE 802.11ax based Medium Access Design for Wireless IoT-Blockchain Networks," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, Taipei, Taiwan, 2020.

[44] E. Khorov, V. Loginov and A. Lyakhov, "Several EDCA parameter sets for improving channel access in IEEE 802.11ax networks," in *2016 International Symposium on Wireless Communication Systems (ISWCS)*, 2016.

[45] M. Alkalbani, B. Hamdaoui, N. Zorba and A. Rayes, "A Blockchain-Based IoT Networks-on-Demand Protocol for Responsive Smart City Applications," in *2019 IEEE Global Communications Conference (GLOBECOM),* 2019.

[46] E. Datsika, A. Antonopoulos, N. Zorba and C. Verikoukis, "Adaptive Cooperative Network Coding Based MAC Protocol for Device-to-Device Communication," in *2015 IEEE International Conference on Communications*

(ICC), London, 2015.

[47] S. K. Bose, An introduction to queuing systems, New York: Springer Science+Business Media, LLC , 2002.

[48] S. Teymori and W. Zhuang, "Queue Analysis for Wireless Packet Data Traffic," in *NETWORKING 2005. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, Berlin, Springer, 2005.

[49] G. Wood, "Ethereum: A secure decentralised generalised transaction," *Ethereum project yellow paper*, vol. 151, pp. 1-32, 2014.

[50] C. Faria and M. Correia, "BlockSim: Blockchain Simulator," in *IEEE International Conference on Blockchain (Blockchain)*, Atlanta, 2019.

[51] B. L. Nelson, "Simulation Programming with VBASim," in *Foundations and Methods of Stochastic Simulation*, Bostton, Springer, 2012.

APPENDIX A: TRANMISSION MODEL MATLAB CODE

1. Throughput Analysis

```
% Numerical calculations
clear; clc; close all;
% Set default parameters
SIFS = 28; AIFS = 128; slot_time = 50; prop_delay = 1;Trigger_Frame=16;Tv=4000;% us
payload512 = 512; MAC_header = 272; PHY_header = 128; ack = 112; % bits
packet = MAC_header + PHY_header + payload512; % bits
ACK = 112 + PHY_header; % bits
%Ts = (Packet + SIFS + ACK + AIFS + 2 * prop_delay) / slot_time;
%Tc = (Packet + AIFS + prop_delay) / slot_time;
      transaction_size=64;

      %% payload512 = 512;
      W = 32;
      m = 3;
      Nc=1; %the average #of collision before tranmitting a frame= average #of retransmission
      Nt=51; %total number of full nodes
      Nf= Nt-1; %total number of full nodes-leading full node
      Tb1= ((payload512/transaction_size)*Tv);
      % Computation payload=512
      throughputcom512 = [];
      delaycom512=[];
      for n = 5 : 1 : Nf % Start from 5 stations condition
      Ts = (AIFS+Trigger_Frame + SIFS +packet+ n* Tb1+ n*ACK +SIFS + 2 * prop_delay) / slot_time; %average time that the channel is captured with a successful transmission
      Tc = (packet + AIFS+ Trigger_Frame + SIFS + prop_delay) / slot_time; %average time that the channel is captured by stations which collide
          fun = @(p) (p-1+(1-2*(1-2*p)/((1-2*p)*(W+1)+ p*W*(1-(2*p)^m)))^(n-1));
          % P is the probability that transmitted packet collide
          P = fzero(fun,[0,1]);
          % tau is probability that a station transmits in a generic slot time
          tau = 2*(1-2*P)/((1-2*P)*(W+1)+ P*W*(1-(2*P)^m));
          % Ptr is that in a slot time there is at least one transmission
          Ptr = 1 - (1 - tau) ^ n;
          % Ps is the probability that a transmission is successful
          Ps = n * tau * (1 - tau) ^ (n - 1) / Ptr;
          % ETX is the number of consecutive idle slots between two consecutive transmissions on the channel
          E_Idle = 1 / Ptr - 1;
          % Throughput = Ps * E[P] / (ETX + Ps * Ts +
  (1 - Ps) * Tc)
          throughputcom512 = [throughputcom512, Ps*(payload512/slot_time)/(E_Idle+Ps*Ts+(1-Ps)*Tc)];
      end
      plot(5 : 1 : 50,throughputcom512,'b*','LineWidth',0.5);
      xlabel('Number of Stations');
      ylabel('Total throughput');
      title('Total throughput vs Number of different stations in basic case');
      hold on;
      grid on;
      %simulation payload=512
      sim_time = 0;
      delaysim512=[];
      throughputsim512=[];
      for n = 5 : 1 : Nf
          suc_pkt = 0;
          collision_pkt = 0;
          station_stage = zeros(1,n);
          next_tran_time = zeros(1,n);

          for i = 1:n % initial backoff time
              station_stage(i) = 0;
              next_tran_time(i) = AIFS + floor(W * rand) * slot_time;
          end

          while suc_pkt < 100000
              tran_time =  min(next_tran_time);
              no_tran = sum(tran_time == next_tran_time);

              if no_tran == 1 % successful transition
                  suc_pkt = suc_pkt + 1;
                  for i = 1:n
                      if next_tran_time(i) == tran_time % uniform backoff at stage 0
                          if suc_pkt == 100000
                              sim_time = next_tran_time(i)+Trigger_Frame+packet+SIFS+n*ACK+n*AIFS+ n * Tb1;
                              %(AIFS+Trigger_Frame + SIFS +Packet+ n* Tb+ ACK +SIFS + 2 * prop_delay) / slot_time;
                          end
                          station_stage(i) = 0;
                          next_tran_time(i) = next_tran_time(i)+Trigger_Frame+packet+n * Tb1+SIFS+n*ACK+AIFS+floor(W*rand)*slot_time;
                      else % stop counting while the channel is busy
                          next_tran_time(i) = next_tran_time(i) +Trigger_Frame+ packet + SIFS+ n*ACK+n * Tb1 + AIFS;
                      end
                  end

              else % collision
                  collision_pkt = collision_pkt+1;
                  for i = 1:n
                      if next_tran_time(i) == tran_time
```

```
% uniform backoff at next stage
                        if station_stage(i) < m
                            station_stage(i) = station_stage(i) +1;
                        end
                        next_tran_time(i)=next_tran_time(i) + Trigger_Frame+packet + AIFS+ floor(2^station_stage(i)*W*rand)*slot_time;
                    else
                        next_tran_time(i)=next_tran_time(i) +Trigger_Frame+ packet + AIFS;
                    end
                end
            end
        end
        throughputsim512 = [throughputsim512 suc_pkt*(payload512)/sim_time];

    end
    plot(5 : 1 : 50,throughputsim512,'ro','LineWidth',1.5);
    hold on;
    title('Saturation throughput: Matlab simulation')
    xlabel('Number of Stations');
    ylabel('Saturation Throughput');
    grid on;


    %% payload1024 = 1024;
    payload1024 = 8000;
    Tb5= ((payload1024/transaction_size)*Tv);
    % Computation payload=1024
    throughputcom1024 = [];
    delaycom1024=[];

    for n = 5 : 1 : Nf % Start from 5 stations condition
    Ts = (AIFS+Trigger_Frame + SIFS +packet+ n* Tb5+ n*ACK +SIFS + 2 * prop_delay) / slot_time; %average time that the channel is captured with a successful transmission
    Tc = (packet + AIFS+ Trigger_Frame + SIFS + prop_delay) / slot_time; %average time that the channel is captured by stations which collide
        fun = @(p) (p-1+(1-2*(1-2*p)/((1-2*p)*(W+1)+ p*W*(1-(2*p)^m)))^(n-1));
        % P is the probability that transmitted packet collide
        P = fzero(fun,[0,1]);
        % tau is probability that a station transmits in a generic slot time
        tau = 2*(1-2*P)/((1-2*P)*(W+1)+ P*W*(1-(2*P)^m));
        % Ptr is that in a slot time there is at least one transmission
        Ptr = 1 - (1 - tau) ^ n;
        % Ps is the probability that a transmission is successful
        Ps = n * tau * (1 - tau) ^ (n - 1) / Ptr;
        % ETX is the number of consecutive idle slots between two consecutive transmissions on the channel
        E_Idle = 1 / Ptr - 1;
        % Throughput = Ps * E[P] / (ETX + Ps * Ts + (1 - Ps) * Tc)
        throughputcom1024 = [throughputcom1024, Ps*(payload1024*2/slot_time)/(E_Idle+Ps*Ts+(1-Ps)*Tc)];
    end
    plot(5 : 1 : 50,throughputcom1024,'k','LineWidth',0.5);
    xlabel('Number of Stations');
    ylabel('Total throughput');
    title('Total throughput vs Number of different stations in basic case');
    hold on;
    grid on;
    %simulation payload=1024
    sim_time = 0;
    delaysim1024=[];
    throughputsim1024=[];
    for n = 5 : 1 : Nf
        suc_pkt = 0;
        collision_pkt = 0;
        station_stage = zeros(1,n);
        next_tran_time = zeros(1,n);

        for i = 1:n % initial backoff time
            station_stage(i) = 0;
            next_tran_time(i) = AIFS + floor(W * rand) * slot_time;
        end

        while suc_pkt < 100000
            tran_time =  min(next_tran_time);
            no_tran = sum(tran_time == next_tran_time);

            if no_tran == 1 % successful transition
                suc_pkt = suc_pkt + 1;
                for i = 1:n
                    if next_tran_time(i) == tran_time % uniform backoff at stage 0
                        if suc_pkt == 100000
                            sim_time = next_tran_time(i)+Trigger_Frame+packet+SIFS+n*ACK+n*AIFS+ n * Tb5;
                            %(AIFS+Trigger_Frame + SIFS +Packet+ n* Tb+ ACK +SIFS + 2 * prop_delay) / slot_time;
                        end
                        station_stage(i) = 0;
                        next_tran_time(i) = next_tran_time(i)+Trigger_Frame+packet+n * Tb5+SIFS+n*ACK+AIFS+floor(W*rand)*slot_time;
                    else % stop counting while the channel is busy
                        next_tran_time(i) = next_tran_time(i) +Trigger_Frame+ packet + SIFS+ n*ACK+n * Tb5 + AIFS;
                    end
                end

            else % collision
                collision_pkt = collision_pkt+1;
                for i = 1:n
                    if next_tran_time(i) == tran_time % uniform backoff at next stage
                        if station_stage(i) < m
                            station_stage(i) = station_stage(i) +1;
                        end
                        next_tran_time(i)=next_tran_time(i) +Trigger_Frame+ packet + AIFS+ floor(2^station_stage(i)*W*rand)*slot_time;
                    else
                        next_tran_time(i)=next_tran_time(i) +Trigger_Frame+ packet + AIFS;
                    end
                end
            end
        end
        throughputsim1024 = [throughputsim1024 suc_pkt*(payload1024*2)/sim_time];

end
plot(5 : 1 : 50,throughputsim1024,'gx','LineWidth',1.5);
    hold on;
    title('Saturation throughput vs Number of nodes')
    xlabel('Number of nodes');
    ylabel('Saturation Throughput');
    grid on;

    legend('Computation/ Payload= 512 bits ', 'Simulation/ Payload= 512 bits', 'Computation/ Payload= 1024 bits ', 'Simulation/ Payload= 1024 bits');
```

2. Delay Analysis

```
clear; clc; close all;
% Set default parameters
SIFS = 28; AIFS = 128; slot_time = 50; prop_delay = 1;Trigger_Frame=16;Tv=4000;% us
payload512 = 512; MAC_header = 272; PHY_header = 128; ack = 112; % bits
packet = MAC_header + PHY_header + payload512; % bits
ACK = 112 + PHY_header; % bits
transaction_size=64;

W = 32;
    m = 3;
    Nc=1; %the average #of collision before tranmitting a frame= average #of retransmission
    Nt=51; %total number of full nodes
    Nf= Nt-1; %total number of full nodes-leading full node
    Tb5= ((payload512/transaction_size)*Tv);
    %%
    % Computation payload=512
    delaycom512=[];
    for n = 5 : 1 : Nf % Start from 5 stations condition
    Ts = (AIFS+Trigger_Frame + SIFS +packet+ n* Tb5+ n*ACK +SIFS + 2 * prop_delay) / slot_time; %average time that the channel is captured with a successful transmission
    Tc = (packet + AIFS+ Trigger_Frame + SIFS + prop_delay) / slot_time; %average time that the channel is captured by stations which collide
        fun = @(p) (p-1+(1-2*(1-2*p)/((1-2*p)*(W+1)+ p*W*(1-(2*p)^m)))^(n-1));
        % P is the probability that transmitted packet collide
        P = fzero(fun,[0,1]);
        % tau is probability that a station transmits in a generic slot time
        tau = 2*(1-2*P)/((1-2*P)*(W+1)+ P*W*(1-(2*P)^m));
        % Ptr is that in a slot time there is at least one transmission
        Ptr = 1 - (1 - tau) ^ n;
        % Ps is the probability that a transmission is successful
        Ps = n * tau * (1 - tau) ^ (n - 1) / Ptr;
        % ETX is the number of consecutive idle slots between two consecutive transmissions on the channel
        E_Idle = 1 / Ptr - 1;
        delaycom512= [delaycom512, n*(E_Idle+Ps*Ts+(1-Ps)*Tc)]
    end
    plot(5 : 1 : 50,delaycom512,'r','LineWidth',1.5);
    xlabel('Number of Stations');
    ylabel('Total delay');
    title('Total delay vs Number of different stations');
    grid on;
    hold on;
    sim_time= 0;
    delaysim512=[];

    for n = 5 : 1 : Nf
        suc_pkt = 0;
        collision_pkt = 0;
        station_stage = zeros(1,n);
        next_tran_time = zeros(1,n);

        for i = 1:n % initial backoff time
            station_stage(i) = 0;
            next_tran_time(i) = AIFS + floor(W * rand) * slot_time;
        end

        while suc_pkt < 100000
            tran_time =  min(next_tran_time);
            no_tran = sum(tran_time == next_tran_time);

                if no_tran == 1 % successful transition
                    suc_pkt = suc_pkt + 1;
                for i = 1:n
                    if next_tran_time(i) == tran_time % uniform backoff at stage 0
                        if suc_pkt == 100000
                            sim_time = next_tran_time(i)+packet+SIFS+ACK+n*AIFS+ n*Tb5;
                            %(AIFS+Trigger_Frame + SIFS +Packet+ n* Tb+ ACK +SIFS + 2 * prop_delay) / slot_time;
                        end
                        station_stage(i) = 0;
                        next_tran_time(i) = next_tran_time(i)+Trigger_Frame+packet+n * Tb5+SIFS+n*ACK+AIFS+floor(W*rand)*slot_time;
                    else % stop counting while the channel is busy
                        next_tran_time(i) = next_tran_time(i) +Trigger_Frame+ packet + SIFS+ n*ACK+n * Tb5 + AIFS;
                    end
                end

            else % collision
                collision_pkt = collision_pkt+1;
                for i = 1:n
                    if next_tran_time(i) == tran_time % uniform backoff at next stage
                        if station_stage(i) < m
                            station_stage(i) = station_stage(i) +1;
                        end
                        next_tran_time(i)=next_tran_time(i) + packet + AIFS+ floor(2^station_stage(i)*W*rand)*slot_time;
                    else
                        next_tran_time(i)=next_tran_time(i) + packet + AIFS;
                    end
                end
            end
        end
        delaysim512= [delaysim512, n*sim_time];

    end
    plot(5 : 1 : 50,delaysim512,'b','LineWidth',1.5);
    xlabel('Number of Stations');
    ylabel('Total delay');
    title('Total delay vs Number of different stations');
    grid on;
    hold on;
    %%
    payload1024=1024
    Tb1= ((payload1024/transaction_size)*Tv);
    %%
    delaycom1024=[];
    for n = 5 : 1 : Nf % Start from 5 stations condition
    Ts = (AIFS+Trigger_Frame + SIFS +packet+ n* Tb1+ n*ACK +SIFS + 2 * prop_delay) / slot_time; %average time that the channel is captured with a successful
```

```
transmission
    Tc = (packet + AIFS+ Trigger_Frame + SIFS + prop_delay) / slot_time; %average time that the channel is captured by stations which collide
        fun = @(p) (p-1+(1-2*(1-2*p)/((1-2*p)*(W+1)+ p*W*(1-(2*p)^m)))^(n-1));
        % P is the probability that transmitted packet collide
        P = fzero(fun,[0,1]);
        % tau is probability that a station transmits in a generic slot time
        tau = 2*(1-2*P)/((1-2*P)*(W+1)+ P*W*(1-(2*P)^m));
        % Ptr is that in a slot time there is at least one transmission
        Ptr = 1 - (1 - tau) ^ n;
        % Ps is the probability that a transmission is successful
        Ps = n * tau * (1 - tau) ^ (n - 1) / Ptr;
        % ETX is the number of consecutive idle slots between two consecutive transmissions on the channel
        E_Idle = 1 / Ptr - 1;
        delaycom1024= [delaycom1024, n*(E_Idle+Ps*Ts+(1-Ps)*Tc)]
    end
    plot(5 : 1 : 50,delaycom1024,'c','LineWidth',1.5);
    xlabel('Number of Stations');
    ylabel('Total delay');
    title('Total delay vs Number of different stations');
    grid on;
    hold on;
    sim_time= 0;
    delaysim1024=[];

    for n = 5 : 1 : Nf
        suc_pkt = 0;
        collision_pkt = 0;
        station_stage = zeros(1,n);
        next_tran_time = zeros(1,n);

        for i = 1:n % initial backoff time
            station_stage(i) = 0;
            next_tran_time(i) = AIFS + floor(W * rand) * slot_time;
        end

        while suc_pkt < 100000
            tran_time =  min(next_tran_time);
            no_tran = sum(tran_time == next_tran_time);

            if no_tran == 1 % successful transition
                suc_pkt = suc_pkt + 1;
                for i = 1:n
                    if next_tran_time(i) == tran_time % uniform backoff at stage 0
                        if suc_pkt == 100000
                            sim_time = next_tran_time (i)+packet+SIFS+ACK+n*AIFS+ n * Tb1;
                            %(AIFS+Trigger_Frame + SIFS +Packet+ n* Tb+ ACK +SIFS + 2 * prop_delay) / slot_time;
                        end
                        station_stage(i) = 0;
                        next_tran_time(i) = next_tran_time(i)+Trigger_Frame+packet+n * Tb1+SIFS+n*ACK+AIFS+floor(W*rand)*slot_time;
                    else % stop counting while the channel is busy
                        next_tran_time(i) = next_tran_time(i) +Trigger_Frame+ packet + SIFS+ n*ACK+n * Tb1 + AIFS;
                    end
                end

            else % collision
                collision_pkt = collision_pkt+1;
                for i = 1:n
                    if next_tran_time(i) == tran_time % uniform backoff at next stage
                        if station_stage(i) < m
                            station_stage(i) = station_stage(i) +1;
                        end
                        next_tran_time(i)=next_tran_time(i) + packet + AIFS+ floor(2^station_stage(i)*W*rand)*slot_time;
                    else
                        next_tran_time(i)=next_tran_time(i) + packet + AIFS;
                    end
                end
            end
        end
        delaysim1024= [delaysim1024, n*sim_time];

    end
    plot(5 : 1 : 50,delaysim1024,'g','LineWidth',1.5);
    xlabel('Number of nodes');
    ylabel('Total transmission delay');
    title('Total transmission delay vs Number of nodes');
    grid on;
    hold on;
    legend('Computation/ Payload= 512 bits ', 'Simulation/ Payload= 512 bits', 'Computation/ Payload= 1024 bits ', 'Simulation/ Payload= 1024 bits');

```

APPENDIX B: QUEUING MODEL MATLAB CODE

```
lambda=0.1;
g=poissrnd(lambda);
X=20000.*rand(1,g);
S=14700.*rand(1,g)
t=0:0.01:25;
F=zeros(size(t));
```

```
for ii=2:length(t)
for jj=1:g
    if t(ii)>X(jj) && t(ii-1)<X(jj)
        F(ii-1:ii+4)=1;
    end
end
end
subplot(3,1,1);
plot(t,F)
ylim([0 3])
M=zeros(1,length(t));
for ii=1:g
F=t>X(ii) & t<X(ii)+S(ii);
M=M+F;
end
subplot(3,1,2)
plot(t,M)
[d p]=sort(X);
flag=zeros(g,1);
P=[d' S(p)' flag flag];
P(1,3)=P(1,1);
P(1,4)=1;
cn=P(1,1)+P(1,2);
for ii=2:g
a=find(P(:,1)<cn);
b=find(P(:,4)==0);
c=intersect(a,b);
if isempty(c)
    P(b(1),3)=P(b(1),1);
    cn=P(b(1),1)+P(b(1),2);
    P(b(1),4)=1;
else
    P(c(1),3)=cn;
    cn=cn+P(c(1),2);
    P(c(1),4)=1;
end
end
M=zeros(1,length(t));

for ii=1:g
F=t>P(ii,3) & t<P(ii,3)+P(ii,2);
T=-t+P(ii,3)+P(ii,2);
M=M+F.*T;
F=zeros(1,length(t));
end
subplot(3,1,3)
plot(t,M)

p=lambda.*mean(P(:,2));
Es2=var(P(:,2))+mean(P(:,2)).^2;
Ew=lambda.^2.*Es2./(2.*(1-p))
EA=lambda.^2.*Es2./(2.*(1-p))+mean(P(:,2))
EN=lambda.*mean(P(:,2))+Ew
```