Doctoral theses at NTNU, 2023:44

Befekadu Gezaheng Gebraselase

Networking Impact and Support for Blockchains

Doctoral thesis

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of Philosophiae Doctor
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of Science and Technology

Befekadu Gezaheng Gebraselase

Networking Impact and Support for Blockchains

Thesis for the Degree of Philosophiae Doctor

Trondheim, March 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU

Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

ISBN 978-82-326-6206-7 (printed ver.)
ISBN 978-82-326-5570-0 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2023:44

Printed by NTNU Grafisk senter

Abstract

With the rapid development of information and communication technologies, infrastructures, resources, and networking applications, systems have become complex and heterogeneous. This also creates a massive amount of data exchange, and end devices may produce security, privacy, and performance issues. The research community has proposed blockchain as a disruptive technology to bring about decentralized, secure, transparent, and efficient network operation and management. One of the critical challenges is investigating the capability of the technology so that it can be used in such a complex domain. This thesis aims to examine the impact and support of networking in blockchain while providing insight into the workflow of the technology, investigate its capability in terms of the transaction confirmation time, and explore the applicability of the technology in such domains and possible setbacks that may arise. The research contributions of this Ph.D. work are divided into three parts.

First, we prepared a testbed to study blockchain capabilities. Based on the collected dataset, a comprehensive study of the transaction characteristics of Bitcoin, the first blockchain-based application, was performed. A set of results and finding of the fundamental process were obtained, such as inter-block generation and inter-transaction arrival following an exponential distribution. The transaction fee and size distribution differ from the documentation. Moreover, the block size distribution varies over time, mainly because transaction generation varies over time. Compared to the state of the art, to the best of our knowledge, this work is the first to confirm several hypotheses and assumptions in the literature.

Furthermore, Bitcoin can process 3.3 to 7.2 transactions per seconds, which has increased the number of arrivals waiting for pick up. This has resulted in an increase in transaction fees and affected the transaction confirmation time. This led to a study on understanding and predicting how Bitcoin handles transactions. Thus, a user will know when to request a transaction and miners will know to choose the appropriate mining pool to earn a fair reward. The analysis consists of two parts. The first part is an exploratory data analysis revealing critical characteristics of various fundamental processes for handling Bitcoin transactions. The second part is a predictability analysis intended to provide answers to or insights into several fundamental aspects of transactions handling.

Second, based on the fundamental transaction characteristics, we examined the impact of peer formation strategies and how miners' financial interests affect

the transaction waiting time. We also investigated the impact of the number of peers (peer lists) per node or miner and the end-to-end delay to understand node-to-node communication. A testbed-based study showed that peer selection strategies affect transaction propagation and confirmation time. The study also showed that smaller transactions exhibit longer confirmation times, even with the increasing block size. Moreover, the miner transaction selection strategy impacts the final gain.

Third, blockchain as a network function has been proposed to support the underlying network infrastructure to provide services that satisfy stringent QoS requirements. However, few works in the literature have investigated the suitability of blockchain in networking and the possible setbacks that may arise. To fill this research gap, we conducted a state-of-the-art study on whether blockchain can be adopted in networking and then to what extent possible use cases can be provided. For instance, as a service, blockchain allows tenants and subscribers to manage slice information as necessary without violating the agreement.

In summary, in this thesis, we examine the impact and support of networking for the evolving technology, blockchain, while highlighting to what extent it can be used in complex and heterogeneous systems.

Preface

This dissertation is submitted in partial fulfillment of the requirements for the degree Philosophiae doctor (Ph.D.) at NTNU, Norwegian University of Science and Technology. The Ph.D. thesis work was carried out in the Department of Information Security and Communication Technology (IIK), Trondheim, under the supervision of Professor Bjarne E. Helvik and the co-supervision of Professor Yuming Jiang. This project was part of Trust and Transparency in Digital Society Through Blockchain Technology, funded by the Research Council of Norway.

Acknowledgements

First, I want to thank God for giving me the patience and strength to endure all my challenges.

Numerous people contributed direct or indirect support to this Ph.D. journey. I would like to thank my supervisor, Prof. Bjarne Emil Helvik, for his leap in faith to trust me to join this exciting research topic, for his guidance, support, and reassurance throughout the Ph.D. program, and for some of his life advice. I am also grateful to my co-supervisor, Prof. Yuming Jiang, for his guidance, inspiration, support, and contribution to our work.

I am also very thankful for our administrative staff, especially Mona, Maria, and Pål. Mona and Maria's office is always open to questions about anything. Their office is a Google search engine for any problem and how to address it. Similarly, Pål was so helpful with technical issues. Although some look complex, he has a simple way of managing them.

I have been fortunate enough to have close colleagues who created a balanced office and out-of-office activities. Thanks to Mayank, Ali, Sonu, Charles, Enio, Ergys, Murad, Sruti, Goda, Teda, Tesfaye, and Mistre for exciting discussions. Special appreciation goes to Prof. Poul for introducing Paddle, an exciting game that has become a habit. To my best friends, Sara, Fresew, Yorda, Nahom, and Elsha, you guys are the true definition of friends, and I am so grateful to have you in my life.

A special thank you to my previous mentor, Assistant Professor Tessema M. Mengistu. You played a crucial role in this journey, guiding and supporting me through different challenges on the way and giving remarkable life advice.

Finally, I want to thank my family for their support throughout this challenging journey. My two little sisters (Tizita and Ruth), looking at you both growing fast is special and hard to swallow. Special appreciation to Zeleka, Emushu, Mimi, and Abaye. Life would have been so boring without you all.

Contents

List of Figures

List of Tables

List of acronyms

VM	Virtual Machine
DLT	Distributed Ledger Technology
RPC	Remote Procedure Call
P2P	Peer to Peer Network
PoW	Proof of Work
PoS	Proof of Stake
DPoS	Deligated Proof of Stake
PBFT	Practical Byzantine fault tolerance
RPCA	Ripple protocol consensus algorithms
DNS	Domain Name System
INV	Inventory
SegWit	Segregated Witness
SAN	A Stochastic Activity Network
HTLCs	Hashed Timelock Contracts
MAST	Merkelized Abstract Syntax Tree
5G	Fifth-Generation Technology
DDOS	Distributed Denial-of-service
IoT	Internet of Things
NAT	Network Address Translation
BGP	Border Gateway Protocol
SDN	Software-defined Networking
NAR	Nonlinear Autoregressive Neural Network
NARX	Nonlinear Autoregressive Network with Exogenous Inputs
ARIMA	AutoRegressive Integrated Moving Average
ARIMAX	Autoregressive Integrated Moving Average with Exogenous input

Part I
Thesis Introduction

Chapter 1

Introduction

Blockchain is a distributed, shared, immutable ledger that stores records of transactions and assets [52]. It is a decentralized decision-based technology that removes the middle man between participating units. These fundamental properties have attracted a vast amount of attention, including in complex systems such as 5G [6, 53], smart grids [9, 51, 75], health [31, 42], and finance [46, 52]. However, as the technology is still in the early stages, it faces significant challenges in addressing scalability, performance, and privacy [30].

Blockchains can be classified as public, private, and consortium based on the level of sharing of the record and the right to access. All of the major types follow a similar workflow, in which we can consider some of the common functionalities to examine the characteristics of the ledger. Blockchains have three key elements. The first is the users who participate in the system to receive the service. The second is the miners who validate and add new records to the primary ledger. As a result, they are rewarded for their effort. The third is the underlying network infrastructure that provides a platform for the participating nodes to push or pull updates. This platform contributes to the ledger's stability and consistency.

The first interaction point of blockchain technology is when a user wants to create a valid transaction. This newly generated transaction is pushed through the peer-to-peer (P2P) network to the participating nodes for validation and confirmation. Due to the continuous process of validating each transaction passing through each node, transaction delay occurs. The delay combines the time it takes to validate and the time it takes to disseminate the transaction. Each node stores new arrivals in the backlog, where it must wait to be picked up and added to the block for confirmation. The amount of time a transaction takes to reach other nodes (the propagation) and, the amount of time it has to wait in the backlog (waiting time), the amount of time to process the transaction (the processing time) affect the service quality. The total amount of time reduces the overall quality of the services provided by the ledger and affects users' interest in using the application. When the number of users increases significantly, more transactions experience long wait times, making it less efficient to match user

demands.

Miners or full nodes are vital elements of the blockchain that participate in adding, validating, and forwarding new updates to neighbors. A miner can be involved in solving the mathematical puzzle in the case of public blockchains, such as Bitcoin, through a high computation effort. Miners choose which transactions to add to the block, and this flexibility gives more incentives to the miner to choose a transaction with a high fee to pick up first. At the same time, miners adopt different transaction selection strategies and in some cases, manually choose peers to improve their chances of getting new transactions faster. In addition, some malicious miners may try to compromise the normal operation through block withholding, DDOS attack, or double-spending. The type of strategies adopted by miners in selecting transactions to add to a block, the P2P formation they choose, and their willingness to be honest affect the overall performance of the technology.

Nodes (full nodes or miners) are interconnected and form a P2P network. A node can be a computer/virtual machine (VM)/server participating in the network for validation and processing. The nodes send and receive messages via the underlying network infrastructure, while the P2P topology is formed at the application layer. This topology is responsible for broadcasting new updates to peers, by which they learn and inform each other about transactions and blocks. In a P2P network, nodes are independent, which removes a central authority and brings distributed and decentralized peers [52]. This property increases the system's robustness from a single point of failure and ensures continuous operation as long as the majority of the nodes are honest and operational. However, the challenge comes from nodes becoming private, making it impossible to collect information. Alternatively, creating an investigation from the network fragment may provide a result that does not reveal the characteristics of the whole network. In addition, insufficient information about the whole network makes it challenging to find simulators to capture the properties adequately. Because of this, there are few or none related works that focus on the impact of end-to-end delay, P2P topology, and arrival intensity on the capability of blockchain. Furthermore, the architecture distributed ledger technology (DLT) makes it hard to measure the performance and dependability of the infrastructure as a system [61].

1.1 Motivation

The capabilities of blockchain in different complex systems such as the IoT, smart grids, and 5G have not been examined. For instance, the public blockchain, Bitcoin, can process 3.3 to 7.2 transactions per second, while Visa can process 2000 transactions per second. In addition, a transaction is confirmed when it is six blocks deep, on average, after an hour. This makes public blockchain less efficient to use in time-sensitive, high throughput demand, and high scalability is needed. To comprehend the technology properly, understanding the capabilities of the blockchain in terms of the delay or transaction confirmation time

is necessary, and exploring how transactions are handled is crucial to see the short-term predictability of the technology. This naturally demands a thorough study of the transaction characteristics of a blockchain. Thus, analytical methods (e.g., queueing theory) may be employed to estimate the performance of the blockchain. It would also provide helpful insight into designing and developing new blockchains.

The structural design of a blockchain makes it have a fixed block size that limits the number of transactions to be added. Due to the limited capacity by design, the number of transactions that the system can handle is also limited. This necessitates a strategy for a miner to select transactions in forming blocks. A bigger block size causes a longer block propagation delay that may affect the acceptance of the block by the network. For instance, if two blocks are generated with a short inter-generation time, the smaller block size has a higher chance of reaching more nodes. In most cases, adding more transactions in a block also increases the total financial gain. Although it is natural for miners to prioritize higher-fee transactions to gain financially, such a strategy may cause a long delay in transaction confirmation for lower-fee transactions. Thus, such financial gain-oriented strategies may reduce the overall quality of the services provided by the ledger. Understanding the impact of block size limitations and miners' incentives on the overall performance is vital, because they are crucial indicators of the ledgers' stability.

The P2P network is responsible for providing a communication channel between the participating nodes. It also plays a significant role in providing the quality of service that matches user demand. The nodes in the P2P network are autonomous, distributed geographically, and in some cases, hidden from the outside world through NAT and a firewall. Therefore, it is difficult to collect traffic and transaction-related information to analyze the technology. This creates a challenge in finding a proper measurement dataset to explore the transaction characteristics, the impact of peer-to-peer formation strategies, the effect of end-to-end delay, and how the fork occurrence impacts the quality of the service. These network-related parameters have a direct or indirect effect on the performance of the technology. Nevertheless, there are few or no investigations in the current literature.

Blockchains are considered an alternative technology for addressing privacy, security, and enhanced performance in networking. These considerations lack state-of-the-art studies on the capability of blockchain, such as public blockchain, as a technology that faces challenges in providing higher throughput and latency, which makes it difficult to apply this technology in time-sensitive and stringent requirement domains. However, insight into the research on how blockchain can contribute to networking in information management, bookkeeping, and secondary mechanisms to provide security through keeping logs and records could be highlighted.

1.2 Thesis Structure

This thesis is structured in two parts. The first part (Part I) provides an introduction and overview of the collected papers presented in Part II.

Part I contains six chapters. Following the current introduction chapter, the remaining chapters are organized as follows: Chapter 2 gives a brief background on blockchain and critical concepts. Chapter 3 discusses the current state of the art in examining the performance of blockchain. Chapter 4 highlights the research goals and the methodology. Chapter 5 presents the contributions of the thesis. Finally, Chapter 6 summarizes the thesis conclusions and recommends future research directions.

Chapter 2

Background

This chapter reviews the background of the main characteristics and features of the networking impact of and support for blockchain. Section 2.1 gives an overview of blockchain and its fundamental elements. Next, Section 2.2 provides a highlight on different types of blockchain. Section 2.3 presents the most common consensus protocols and comparisons. Following that, Section 2.4 provides a short discussion. Section 2.5 presents a detailed description of the Bitcoin workflow and interactions. Section 2.6 highlights how Bitcoin nodes perform a peer-to-peer discovery and transaction exchange.

2.1 Blockchain

The Blockchain system architecture allows the involved parties to communicate and exchange in a peer-to-peer (P2P) network through which distributed decisions are performed by the majority rather than by a centralized authority. As the word states, a blockchain is a chain of blocks (records). Each block has a pointer to the previous block (previous hash), nonce, and transaction list, as illustrated in Figure 2.1. The blocks contain the previous block's cryptographic hash, which makes it hard to temper or reverse the current transaction. As the nodes are distributed, consensus protocol used to achieve constancy and stability.

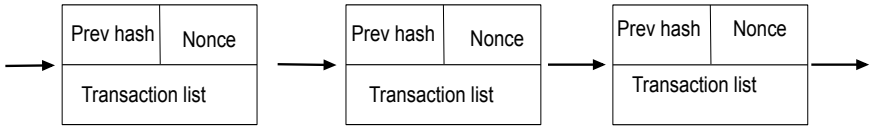


Figure 2.1: Blocks forming a chain

2.1.1 Key elements

Blockchain has key elements that allow the technology to provide a secure and consistent ledger. The elements include the following:

- *User:* The first interaction point of blockchain technology starts from a user's or customer's interest in making a transaction. A valid transaction created by a user propagates through a peer-to-peer network; that is, peers forward new arrivals to their neighbors.
- *Miners:* The peer or peers that maintain and update the chain of blocks are called miners [68]. They are the vital element of the technology that validates, adds, and creates block and transaction information.
- *Peer:* A virtual machine, pool, or node can act as a peer that provides the intended service. A peer can act as a client when receiving new updates from its neighbors and as a server when pushing new updates to its neighbors. In this work, a node refers to the full Nodes that verify all of the rules of Bitcoin.
- *Peer-to-peer network:* Peers form a logical peer to peer network. This logical network is used to push and pull new updates, and when these peers connect to each other, they forms a mesh network.

2.1.2 Workflow

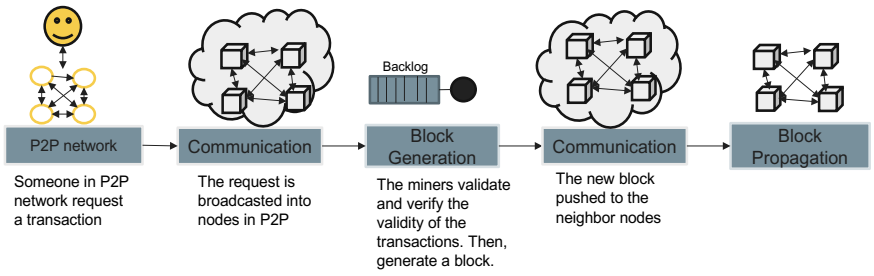


Figure 2.2: Blockchain process flow

Fig. 2.2 illustrates the workflow of the transaction arrival, block formation, propagation, and validation in a blockchain. Users generate a transaction for confirmation. This transaction propagates to the other nodes in the network for validation and confirmation. When full nodes receive new arrivals, the nodes check the transaction's validity. If the transaction is valid, the node stores it in its backlog (the memory pool), waiting for confirmation. If the transaction is invalid, then the node ignores the transaction. When a block generation event happens, the nodes choose unconfirmed transactions in the backlog to pack into a new transaction block. This newly generated block is pushed to other nodes,

and this information is sent to all the nodes. At each node, the validity of the newly generated block is checked. If the validity is confirmed with consensus, the updated blockchain is accepted, and the new block transactions are validated. The validated transactions are removed from the memory pool at each full node, which repeats process above.

2.2 Blockchain Categories

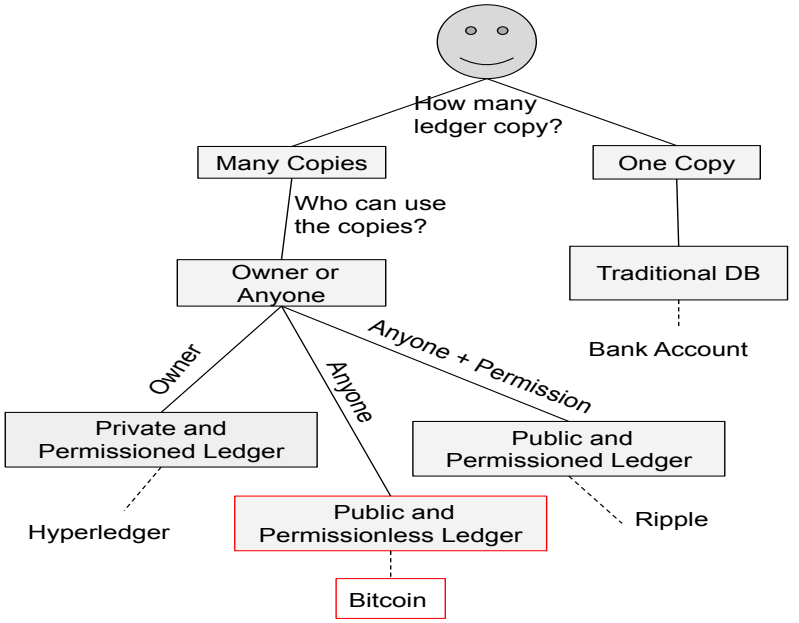


Figure 2.3: Blockchain types

This section covers three types of blockchains: public, private, and consortium [5], as illustrated in Fig. 2.3. The classification of blockchains is based on their characteristics. This implies that different kinds of blockchains have similar process flows.

2.2.1 Public blockchain

Public blockchains are available for any users or nodes to join the network [52]. Based on the level of access restrictions, public blockchains can be divided into two types: permission and permissionless ledgers. The permissionless blockchain allows new nodes to join the network without access restrictions. It provides equal rights to access the blockchain, create new blocks of data, and validate blocks of data. However, a permission-based ledger requires nodes to satisfy the

rules or justify the honesty to participate in the consensus protocol. Permissionless blockchains are more exciting types to study, and they have the property of a fully decentralized and distributed ledger. Most permissionless blockchains are used to exchange and mine cryptocurrencies, such as Bitcoin and Ethereum.

2.2.2 Private blockchain

Private organizations or communities control private blockchains. The central authority determines who can be a node in a private blockchain. The central authority does not necessarily grant all nodes equal rights to perform functions. Access control can be implemented in different ways. It can be an independent authorization system or a set of rules to meet before joining. It is easy to manage the consensus and membership services in private blockchains since all the nodes in the network are well-known. Such alignments enable private blockchain owners or communities to plug and play functions. These properties make private blockchains more suitable for developing applications for many purposes. Developing different forms for different uses allows for the enhancement of pure and easy access. However, this also makes private blockchain unfit for decentralized decision-making [5].

2.2.3 Consortium blockchain

Public blockchains have longer validation times for new updates, while private blockchains are vulnerable to fraud and bad actors. Consortium blockchains have a group of nodes or leaders that make decisions for the whole network rather than a single entity in a private blockchain. Thus, this type of blockchain is suitable for collaboration between different companies or organizations. The most common consortium blockchains are Quorum, Hyperledger, and Corda [48, 65]

2.3 Consensus protocol

In blockchain technology, nodes push and pull updates through the P2P network. All participating nodes receive a notification if the updates add new records or amendments. Although organizations have implemented their own version of consensus algorithms, the primary goal of consensus algorithms is to provide nodes to communicate and offer a validated set to add to the ledger. The most common consensus algorithms are Proof of Work (POW), Proof of Stake (POS), Delegated Proof of Stake (DPOS), Practical Byzantine Fault Tolerance (PBFT), and Ripple. In this thesis, most of the works focus on public permissionless blockchains. Therefore, we provided the core concept of the PoW. The interested reader may find additional information in [19, 48, 64, 65].

2.3.1 Proof of work (POW)

Bitcoin blockchain uses the consensus algorithm called PoW, requires at least 51% control of the whole network to enable manipulation of the network. That is practically not feasible depending on the number of connected nodes and because the attacker will have to attack all nodes simultaneously. PoW gives nodes with high resource use and computation power a bigger chance to solve a mathematical puzzle. Most of the common consensus protocols are presented in Table 2.1. This table provides a comparison of the protocols that includes their advantages.

Table 2.1: Comparison of the consensus algorithms

Cases	POW [26]	PBFT [64]	POS [65]	DPOS [48]	Ripple [19]
Limitations	Energy Consumption	Scalability	Unbalanced Distribution	Decentralization	Highly Centralized
Energy Efficient	No	Yes	Yes	Yes	Yes
Permission	No	No	No	Yes	No
Adversary Tolerance	51%	33%	Unknown	Less than 20%	20%
Throughput	3.3–8.7	10–20	7	Unknown	1500

2.4 Discussion

Sections 2.2 and 2.3 described the different categories of blockchains and their consensus protocols. They also provided an example of each type. For instance, Bitcoin is a permissionless public ledger that uses PoW as a consensus protocol. Bitcoin is the first application to use blockchain to provide a secure and distributed ledger. Based on this fact, this thesis focuses on performing deployment, analysis, and modeling based on Bitcoin. The word Bitcoin and blockchain are used interchangeably beginning in the next section, 2.5; no matter the term used, the emphasis is on the Bitcoin blockchain.

2.5 Bitcoin

This section covers the workflow of a full Bitcoin node and the interactions between nodes. Additionally, it points out key events that may provide valuable information about the system. In addition, some of the well-known limitations are listed.

2.5.1 Introduction to Bitcoin

Bitcoin has taken the world by storm. It is a distributed ledger technology that allows information to be distributed. It enables data not to be centralized or controlled by a single party. It was invented around 2008 by an unknown author or group of people named Satoshi Nakamoto [52], and the currency was first used in 2009. The years 2010-2013 are the crucial years that made Bitcoin

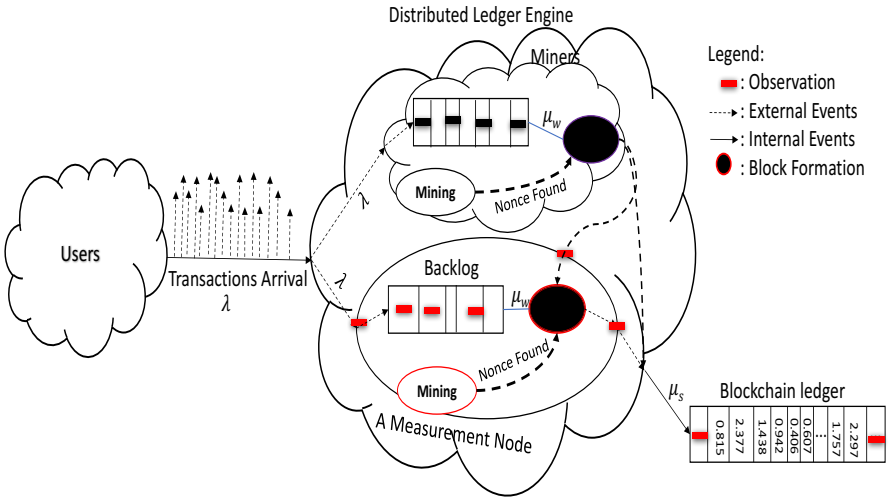


Figure 2.4: An illustration of the work flow of Bitcoin

famous. Since it is the first application to use blockchain technology, it also introduced blockchain to academia and industry.

As section 2.1.2 discusses the process flow of a blockchain, this part provides details of the process flow in Bitcoin. Fig. 2.4 illustrates the workflow of block formation and propagation in Bitcoin. The users, nodes, or pools generate transactions with an intensity of $\lambda(t)$. The receiving nodes store the arrivals in the memory pool for confirmation, while the nodes mine continuously to find the mathematical puzzle to create blocks. When a node finds the puzzle, it includes transactions in the block and forwards the block to the network with a possible propagation delay, μ_s. The block generation intensity (μ_w) can differ among nodes, generally in proportion to the nodes power to mine. A newly generated block is validated by solving a computationally intensive problem using a cryptographic hash algorithm. The winner is awarded by the network, and it continues to find a nonce to create a new block for the blockchain.

At the receiving end, the new block may be accepted by other nodes based on the timestamp, the number of confirmations, and the chain's length. The arrival, propagation, and block generation intensities provide crucial information about the state of the ledger.

2.5.2 Block-related attributes

Understanding Bitcoin also requires exploring what is occurring from the ledger and traffic perspective. Some of the critical attributes of the Bitcoin block and transactions are described below:

- Block size: The block size in Bitcoin extends from 0 to 2.5 MB. The legacy

nodes support the block size until 1 MB. However, the recently updated nodes support Sigwit, which extends the block size by 1 MB.

- The number of transactions: Each generated block contains transactions from a minimum of 0 to a maximum of 4250 transactions.
- Miner: A valid block has a miner that generated and pushed the block. Sometimes, this miner can be a single node, but usually is one of the mining pools.
- Transaction size: A user or customer generates transactions for confirmation. The size of this transaction ranges from 100 bytes to 30 kilobytes.
- Transaction fee: A miner's main motive for processing a transaction is the fee attached to it. It ranges from 0.0000012 to 0.01 BTC.

2.5.3 Mining pool

Miners play a crucial role in validating and generating blocks in the Bitcoin system. They are the vital element of the overall system performance. However, as the difficulty of solving the mathematical puzzle increases every 2016 block, independent miners struggle to solve the puzzle with an average interval of 10 minutes. This has forced miners to collaborate to form a team to solve the puzzle through a combined computational effort, a mining pool. Furthermore, due to the structure of PoW, the time between each two consecutive blocks in Bitcoin is exponentially distributed. The number of blocks found per time period is a Poisson process. Thus, the rate parameter is defined by the ratio of the PoW difficulty to the overall mining power present in the network. Therefore, the individual miners expect to face a high variance in payouts, depending on their overall mining power share.

In such cases, the miners are forced to join a pool that guarantees a fair reward for the amount of computational effort put into solving the mathematical puzzle. However, the pool affect the platform's performance, because as more nodes join a pool, there will be only a few mining pools dominating the overall activity. For instance, a few mining pools produce more than 50% of a valid block. These mining pools can set up an independent strategy to increase their financial gain, which affects transactions with smaller fees.

2.6 P2P Communication

Bitcoin nodes form a peer-to-peer network that runs on top of the underlying network infrastructure. Fig. 2.5 illustrates the relationship between blockchains and the network infrastructure. The amount of traffic generated by the Bitcoin system increases with the number of users and applications integrating the Bitcoin services. Additionally, the Bitcoin nodes have to synchronize the ledger's current state to provide a consistent database. Adding up all these different kinds of traffic, the amount of traffic generated by blockchain technologies in a

P2P network is greater than 5 GB per day. However, the network infrastructure is responsible for making sure all the updates are propagated. Based on these facts, the following two subsections address the impact of networking on Bitcoin and a blockchain in networking.

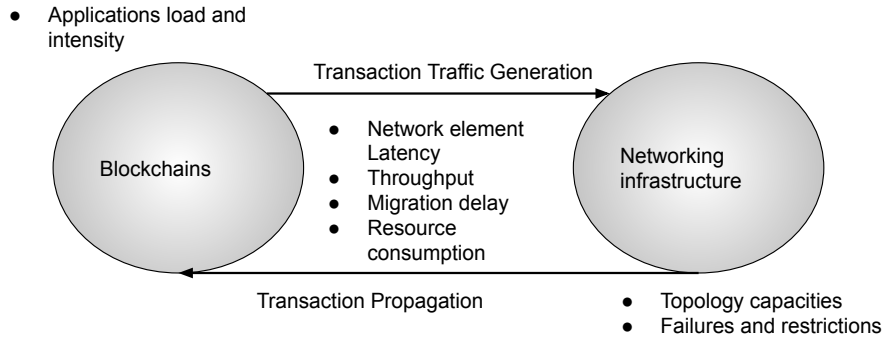


Figure 2.5: Relationship between blockchain and networking

2.6.1 Peer to peer (P2P) discovery

The participating nodes form a peer-to-peer network, where the transactions and blocks are broadcasted to reach all the nodes in the network. By default, nodes have seed peer addresses, and these addresses are essential for the new node to find the first neighbor nodes, using it as a DNS seed. The new node sends the addr message by including its IP address to the neighbor node, and the neighbor node sends the addr message back while including its IP address. From this point on, the new node can request a getaddr message to receive all the neighbor node list according to the connected node. The new node will discover the eight neighbor nodes as a peer list chosen based on low latency. Fig. 2.6 illustrates address propagation and discovery in Bitcoin with only two nodes.

2.6.2 Legacy relay protocol

When a node receives an INV message from neighbor nodes, if the node has the block already, then no extra effort is required. However, if the node has not received the new hash for the block, the node sends a getdata message to collect all the missing blocks. If a node has been offline for a few minutes or a month, it starts by sending a getblocks to previously connected nodes, and it receives an INV response and starts to download the missing block. In this way, the block validated by nodes can be propagated to the connected peers, which is called legacy relaying. This is based on the fact that this technique does not consider the bandwidth limitation and the probability of propagating a transaction that may reside in the memory pool (backlog). The research community is working

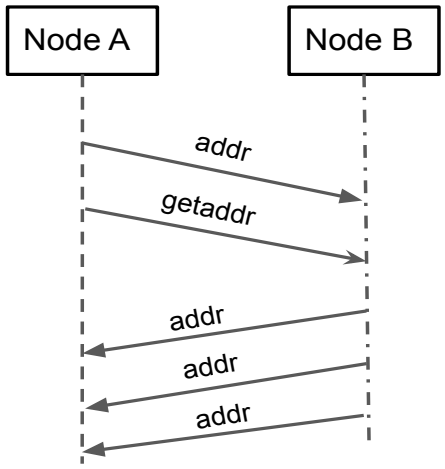


Figure 2.6: Address propagation and discovery

on improving legacy relaying to compact relay. Please refer to the following reference for the compact relay's detailed workflow if the reader is interested [11].

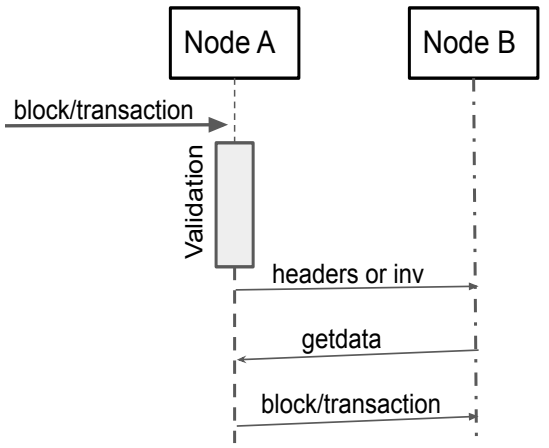


Figure 2.7: Legacy relay protocol

2.7 Bitcoin Limitations

2.7.1 Latency

Bitcoin nodes form a peer-to-peer overlay network. This is a distributed system that guarantees security through the cryptographic hash and the PoW [48]. For the ledger to remain consistent, it requires the newly generated block must reach the available nodes before the new block is generated. Thus, the Bitcoin research community recommended that the inter-block generation time be 10 minutes [52]. However, as the number of nodes increases, it takes longer for most nodes to receive it, resulting in higher latency [30].

2.7.2 Throughput

By using blockchain technology, Bitcoin displays a promising future of trust-free transactions, a permissionless distributed ledger, and pseudonym trading. However, Bitcoin provides a low throughput compared with a centralized system such as Visa and Twitter. Bitcoin can process 3.3 to 7.2 transactions per second [19, 26], which is 1000 times fewer than Visa and Twitter. This is due to the limit of block size, and the block interval makes the number of transactions processed lower. Most researcher claims increasing the block size and reducing the block generation interval may increase the throughput. Decker and Wattenhofer [24] first observed that block size is directly proportional to network delay and that a network delay in the blockchain network leads to increased soft forks. Other studies [28, 30, 62] investigated the trade-off between throughput and provable security to improve the transaction processing speed by shortening the block interval or increasing the block size.

2.7.3 Migration delay and resource consumption

The amount of storage required to install a full Bitcoin node is increasing each year. A full node requires a half-terabyte of space in the current state, which is a massive amount of space for an independent node or miner to prepare to use the full functionality of the Bitcoin. Other than that, depending on the internet provider, a single full node requires 5 GB per day, which requires a large bandwidth. Furthermore, a full node must download a half-terabyte of data to validate and add a new block to the ledger. The number of resources wasted and the migration delay also increase each day as long as the main ledger grows. Simultaneously, the underlying network infrastructure provides a mechanism to download and synchronize between neighboring nodes.

2.8 Impact of Networking in Bitcoin

Bitcoin nodes form a peer-to-peer network that consists of eight neighbor nodes per node. This does not mean a node has to have eight operate; instead, it is the number of nodes the node connects to at a time to push and pull new

updates. The type of topology that may form affects the propagation delay seen by the transactions and blocks. The block is generated with an average time of 10 minutes, as stated by the documentation, although it is around 8.5 minutes. In [8], the authors showed that increasing the participating nodes' rates in the relay network reduces the block propagation delay. Similarly, in [54] the authors showed the impact of the relay network on the orphan block rate; a shorter block propagation time decreases the orphan block rate. Bitcoin can control double-spending by considering transactions permanently confirmed when the block containing the transactions is six blocks deep in the longest accepted blockchain. However, this can be sensitive to peer-to-peer network latency and the impact of soft and hard forks. In [67], it is evident that the time spent on block convergence is proportionally increased with the extension of network latency; moreover, six-blocks confirmation convention of blockchain is affected by sizeable peer-to-peer network latencies. Additionally, a node leaving and rejoining the Bitcoin P2P network also affects the transactions and block propagation times. In [50], the authors showed that the impact on the network of leaving and joining has a noticeable effect on the system's workflow. Thus, the P2P topology, the continuity of nodes being alive, and the end-to-end delay significantly affect the platform's overall performance.

2.9 Blockchain for Networking

Blockchain provides a mechanism for distributing information securely without the involvement of a third-party. Such a security advantage and guarantee of the integrity without the probability of modification of the distributed data attracted attention from the networking world. For instance, providing security in a vehicle network [32], 5G [53, 73], and border gateway protocol (BGP)[41], protecting the privacy of the communication data in the cognitive cellular network [40], wireless mobile network [58], and mobile communications [66], and reducing scalability issues in the internet of things [18].

2.10 Methodology Background

Bitcoin has become more complex and is evolving more rapidly with new proposals and functionalities. However, it is essential to perform an analysis of the previous versions. This led to an increasing need for tools and techniques that assist in understanding the behavior of the system. Modeling then provides a framework for gathering, organizing, assessing, and understanding information about blockchain technology.

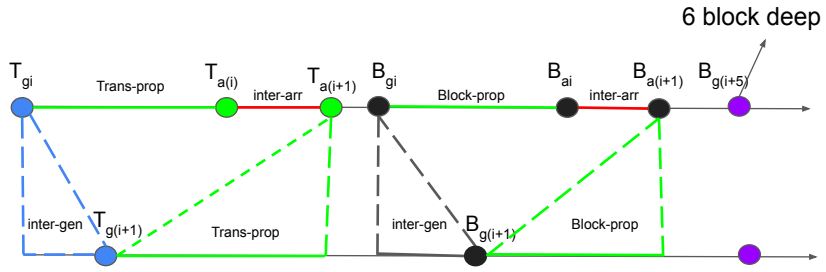


Figure 2.8: Time sequence showing transaction and block generation events

2.10.1 Queueing Concepts

Characteristics of Queueing Systems

The three key elements of blockchain are the user, the miners (nodes), and the P2P network. These three elements can be translated into the queuing system as the customer (user), the server (miners or nodes), and the protocols (P2P network).

System capacity In modeling, considering the capacity of the system is essential. There is a limit to the number of transactions in the wait line. An arriving transaction that finds the system full does not enter but is rejected by the system. However, other cases may have an infinite capacity.

Arrival process The arrival process is usually characterized by the inter-arrival times of successive transactions. For instance, Fig. 2.8 illustrates a time sequence of transaction and block generation, the transaction and block inter-arrival events. Two transactions T_{gi} and $T_{g(i+1)}$ are generated at different timestamps; although they arrive at some node in the network, their inter-arrival time is the difference $T_{a(i+1)} - T_{ai}$. Similarly, the transactions' inter-generation time is the difference $T_{g(i+1)} - T_{gi}$.

Queue discipline Queue discipline refers to the logical ordering of transactions in a backlog and determines which transaction will be chosen for service when a block generation event happens. In Bitcoin, miners have the full right to determine and choose transactions according to their financial incentives. The default strategy is to use a fee per byte. However, miners prefer to use fee-based transaction selection strategies most of the time. These two queue disciplines are considered in this thesis.

Service process In Bitcoin, the service times are when a node solves the mathematical puzzle and creates a block. This new block will be pushed to the network. The block inter-generation time is the time between two consecutive block generations ($B_{g(i+1)} - B_{gi}$). Similarly, when these two blocks arrive at

the node in the network, their inter-arrival time is the difference $B_{a(i+1)} - B_{ai}$. This is demonstrated by the example in Fig. 2.8.

Protocols The protocol, in this case, block or transaction relay protocols (addressed in subsection 2.6.2), provides rules for the different entities in a system or network to cooperate. For instance, Fig. 2.8 illustrates that after a transaction is generated, it is pushed to the neighbor nodes. The time it takes to decimate the transaction into the network is the transaction propagation delay ($T_{ai} - T_{gi}$). Similarly, the time it takes for relay protocols to propagate a block in the network is the block propagation delay. It is the difference between the block generation and arrival times ($B_{ai} - B_{gi}$).

Queuing notation

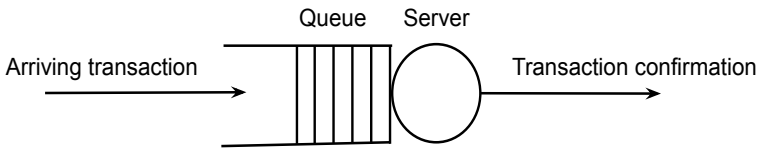


Figure 2.9: Queueing concept

Fig. 2.9 shows transactions arriving at a Bitcoin node (the server), waiting in the queue (the memory pool), processed by the node, and added to the chain (was confirmed). Kendall notation provides a common representation of queueing models and uses five parameters. It is written as $P_a/P_s/n/k/d$, where P_a denotes the probability distribution of the interarrival time, P_b the probability distribution of the service time, n the number of servers in the system, k the maximum number of customers allowed in the system, and d the queue discipline. For instance, in this thesis, we considered $M(t)/M^N/1$, where the transactions' arrival follows an inhomogeneous Poison process to the system with infinite buffer capacity. The service time is distributed exponentially, which removes N transactions at a time.

Chapter 3

State of the Art

This chapter covers the state of the art from two different perspectives. The first perspective is on-chain and off-chain proposals to reduce the wait time and increase the throughput. These proposals are discussed in considerable detail. Second is the state of art in defining models to examine blockchain capability. The focus is on the impact and support of the underlying network infrastructure in the evolving technology, blockchain, as well as how it has been modeled. As a technology, it has some unsolved open challenges, which are listed.

3.1 Scalability Proposals

The Bitcoin community proposed on-chain [15] and off-chain [16] methods to increase throughput. On-chain or first-layer techniques focus on improving the block size in terms of hard forks and soft forks [17]. The former increases the block size by modifying the core codebase, which may lead to incompatibility between versions. The latter (Soft-fork(SegWit)) is a methodology that increases the block size by a virtual volume of 4 MB, but the actual size extends to 2 MB [15]. This technique separates the signature data from Bitcoin transactions. SegWit is one of the methods proposed and implemented in the Bitcoin system to increase the throughput and reduce the wait time. It stores transaction signatures in a separate Merkle tree, which prevents unintended transaction malleability. Moreover, it further enables advanced second-layer protocols, such as the Lightning Network [20], MAST [13], and atomic swaps [14].

One of the second layer or off-chain solutions is the Lightning Network [20]. It is a proposed implementation of hashed timelock contracts (HTLCs) with bi-directional payment channels, which create multiple payment channels between participating entities [12]. The participating nodes need to lock funds on a multi-sig account published in the main network and are allowed to perform transactions with the locked fund as long as the channel is open. A direct payment channel is not required to transfer funds to a party. Intermediate nodes between participants can route the transaction, acquiring a minimal transaction

fee. HTLCs [21] are used to lock the fund based on the expiration time and the propagation of a hashed secret through the routing path. Using channels, the participants can make or receive payments from each other. The transactions are processed differently than the on-chain standard. This method opens a channel for the parties to conduct the process until they reach an agreement. The result is updated on the main blockchain when the two parties open and closes the channel.

Other than on-chain and off-chain proposals, proposals such as sharding and compact rely on reducing traffic usage. Sharding is a traditional technology first proposed in the database field primarily for the optimization of large commercial databases [74]. This method divides the data of an extensive database into several segments and then stores them on different servers to reduce the pressure of the centralized server. Doing so improves the search performance and enlarges the storage capacity of the entire database system. The primary goal is to split a blockchain network into separate shards that contain their own data, different from other shards. The other method proposed by the research community is to perform block compression. Compact block relay [74] and TXilm [27] are some of the recent works that attempt to reduce some redundant data of a block that has been stored in the receiver's memory pool. Txilm [27] is a protocol that compresses the size of the transactions inside a block and reduces the bandwidth utilization of the network. The block carries a short hash of the transaction ID instead of the complete transaction information. The Bitcoin P2P protocol has not been very bandwidth-efficient for block relay. It broadcasts new blocks to neighbors without considering that the transactions within the new block may reside in the backlog (memory pool) of the peers. When the traffic generated by the nodes increases in inbound or outbound, it causes traffic congestion. This may reduce the quality of service the user receives and delay the blocks' relay remotely [49][74]. To address these issues, the Bitcoin research community proposed Compact relay, which provides low and high bandwidth relaying [49]. For a more detailed workflow, please refer to the following references: [11][74].

3.1.1 Limitations

The platform faces challenges in addressing high throughput, scalability, and stability. Hard fork, soft fork, sharding, Block compression, and off-chain have been proposed to reduce the wait time and increase the throughput. These methods concentrate on finding a solution either by adding more block size or validating outside the chain. Based on the default Bitcoin design, the average inter-block generation time is 10 minutes so that the block reaches all the available nodes. Thus, changing the block size may increase it, not improve the throughput. However, shortening the time interval between blocks improves latency and throughput but degrades the platform's security. This means increasing the block size or shortening the block inter-generation time may not solve the throughput problem.

Furthermore, these recent proposals focus on increasing the block and including more transactions, while some gossiping protocol reduces the scalability

hustle. The proposals do not consider the fundamental characteristics of Bitcoin. For instance, rather than focusing on the number of transactions per block, the miner prefer a transaction with a high fee. The Bitcoin community provided a method fee per byte to control and select the top stack transaction from memory pool. However, the method guarantees only transactions with an optimal size relative to the price attached to them. As the fee per byte is not optimized based on the memory pool waiting list, an increase in the block size or extra effort outside the chain may not bring a better solution. Furthermore,the proposed methods have fewer considerations of the effect of a high arrival rate, the number of orphan blocks, occurrence fork, and poor management of the memory pool.

3.2 Blockchain Studies

In modeling a blockchain, the main challenges are due to the distributed nature of the technology. Each node is independent of the others, which means there is no intermediary to collect the traffic in the middle to provide data or perform analysis. Subsection 3.2.1 illustrates the recent works that used measurement-based studies. Then subsection 3.2.2 discusses works that utilized a queueing theory to examine the platform's performance. Finally, subsection 3.2.3 gives a short highlight/overview of machine learning models to review fee fluctuations.

3.2.1 Measurements-based study

Collecting transactions and traffic data from a Bitcoin network is complicated. Some researchers used publicly available data that provide partial information about the technology. Significant works have focused on either analyzing transaction identity [10] [34] [39] [46] or examining the system's long-term credibility [55] [70]. Based on the first paper on Bitcoin [52], the owner account and the account's identity are kept separate, which means they can be anonymous. In [72], the authors analyzed the transaction graph, deriving some global statistics, including an estimate that 78% of the issued Bitcoins are not circulating, and an in-depth analysis of a highly active region in the transaction graph. Babaioff et al. [7] analyzed the incentives for nodes to forward information at all in the network and found that they are insufficient. A dominant strategy in the current system is for a miner to hold on to transactions that include fees and claim them by eventually creating a block that includes the transaction.

Although these recent works focus on the system's security aspect, some works focused on developing a framework for investigating the impact of block size and inter-arrival time, which may reduce the platform's security. Gervais et al. [30] demonstrated that the block size increase from 0.5 to 8 MB lengthens the block propagation time, and the stale block rate increases exponentially. Furthermore, it was demonstrated in [59] that even with a high fee, transactions exhibit long confirmation times.

The limitation in measurement-based studies is the lack of sufficient information about the transactions and traffic data. Most of the works were not validated with observations from a Bitcoin.

3.2.2 Queuing model–based study

A substantial number of works have investigated the average wait time of transactions before they are confirmed. S. Geissler et al. [29] proposed a $GI/GI^N/1$ model, in which the inter-arrival and batch service times follow an independent general distribution. Based on the model, the authors were able to show that the average arrival intensity variations and the block size play a significant role in the confirmation times. Similarly, Li et al. [43] showed that the block size and the average arrival intensity are a significant factor in the average wait time by developing a $GI/M^N/1$ model, in which the inter-arrival time follows a general distribution, but the batch service time follows an exponential distribution. Memon et al. [47] implemented a simulation that abstracted the blockchain mining process. The developed $M/M/n/L$ queuing system was tested in Bitcoin data and trace simulations. The number of transactions per block, mining time, and each block's utilization power were predicted and compared with the sample Bitcoin data. Kawase and Kasahara [38] developed a $M/G^B/1$ model, in which a batch service was used to illustrate the block size limitations with the arrivals blocked from entering a block during the mining phase. The sojourn time of a transaction corresponds to its confirmation time. A similar author [59] showed that because of a high arrival intensity, even high fee transactions exhibit a higher average wait time. Additionally, it was observed how the low fee and the block size significantly affect the transaction confirmation time. Similarly, authors in [37] developed a batch processing queueing system that used a numerical and trace-driven simulation to validate an exponential distribution, and a hyper-exponential simulation that could accurately estimate the mean transaction confirmation time for the legacy 1 MB block size limit. Srivastava [63] developed an M/M/1 model to examine the performance of a blockchain. The result showed that the block delay increases as the block size rises, and the number of transactions within a block is independent of the block generation time.

Most of the works mentioned above focus on investigating the performance of a Bitcoin. Nevertheless, the assumption considered by the previous works does not entirely capture the Bitcoin workflow. For instance, the transaction and block arrival mostly assumed a homogenous Poisson process, but these assumptions have never been validated. Furthermore, the weak dependence between the transaction fee and size were not added to the models to approximate the average wait time experienced by transactions, especially low-fee transactions.

3.2.3 Machine learning models

Network traffic prediction plays a vital role in many study fields, including P2P applications, such as active traffic pattern predictions to boost P2P col-

laboration [33]. There are various machine learning models for network traffic prediction, which are usually classified as linear [22][23][35] and nonlinear [1][25] models. The best forecasting approach is chosen based on considering factors such as the traffic matrix characteristics or performing data analysis before choosing a model. Machine learning models have also attracted attention from the blockchain world to predict transaction price fluctuations. Jang and Lee [36] developed a neural network–based forecast on the volatility of a Bitcoin price and extended their analysis to identify the best feature set that can provide more information about the Bitcoin process. Sin and Wang [60] implemented an artificial neural network to predict the next Bitcoin prices and the amount of profit that could be gained by making such predictions. In [45], the authors applied a conventional neural network model to predict Bitcoin price. Additionally, Lischke and Fabian [44] collected a dataset for a four-year ledger state, which enabled them to make a price prediction and Bitcoin's measured log return distribution. Most of the available literature focused on Bitcoin price volatility, including [3][4][56][69].

More studies have applied linear and nonlinear models to predict network traffic, including [2][57][71]. However, other than using machine learning models to predict price fluctuations and volatility, few researchers have applied machine learning models to predict the short-term evolution of Bitcoin traffic. Motivated by these facts, In this thesis, short-term predictions are considered to understand the impact of internal and external factors in the evolution of a ledger.

3.2.4 Discussion

In recent literature, different researchers used different methods to investigate and characterize what is going on in Bitcoin. In this section, the researchers' work is classified into measurement-based, queueing models, and machine learning models and given a high-level state of the available works.

However, an article that addressed transaction identity and the possibility of security concerns used minimal data, which lacks detailed information to provide such justification. Similarly, queueing models use a different assumption that was not validated by measurements, which will make the result not approximate the Bitcoin very well. However, machine learning models have been introduced to examine fluctuations in transaction fees. This makes most of the works focus on predicting the possible price rather than investigating the patterns that may provide useful insights into the ledger.

3.3 Challenges

The following open challenges are the basis for defining the research goals for this thesis.

- The main challenge in studying blockchain is that a distributed ledger that removes the central unit makes it difficult to collect measurement data. Furthermore, as each node is independent, collecting information

from participating parties requires the owners' permission and willingness to provide correct and valid data. Thus, a valid data is very rare.

- Miners play a significant role in the stability and consistency of the platform. However, financial incentives impact the overall activities. The Bitcoin documentation implies miners use a fee per byte ratio to order the arrival, but the literature shows that is not followed most of the time. It is natural for miners to pick up transactions with a high fees; thus, low-fee transactions wait longer. This leads to Bitcoin becoming unstable for low-fee transactions.

- Increasing the block size may increase the number of transactions processed, but it does not guarantee it always stays that way. Increasing the block size increases the block propagation delay and the number of forks in the system. Understanding the fundamental relationship between the block size and the number of transactions within may provide a different insight into the transaction demand.

- The stability of blockchain technology is affected by the number of fork occurrences. It requires an independent investigation of the significance of the impact and the extent to which it causes damage.

- Performing active or passive measurement of network topology, P2P formation strategies, end-to-end delay, and the impact of arrival intensity on Bitcoin is problematic. It is challenging to estimate as it requires thousands of node deployments.

- Although blockchain has issues to resolve regarding its capability, it is worth highlighting in what sense it has been considered in networking and the possible setback that may arise.

Chapter 4

Research Design

This section presents the research goal, the scope of the thesis, and the methodology employed. Section 4.1 gives the three research goals achieved by this work. Then, Section 4.2 provides a discussion of the scope of the thesis. Finally, Section 4.3 explains the methodology used to address the research goals.

4.1 Research Goals

The main aim of this thesis is to investigate the impact and support of networking for the evolving technology, blockchain. To achieve this goal, we set three research goals that cover different objective angles. The first research goal focuses on exploring the transaction traffic characteristics of Bitcoin, the first application that adopted blockchain. Based on what is observed from research goal 1, research goal 2 is to make an extensive analysis of the impact of a node to node delay, topology formation, and arrival intensity. The research goal 3 studies the suitability of blockchain in complex systems like networking, such as 5G, SDN, and IoT.

4.1.1 Research Goal 1:

To provide a comprehensive study of the blockchain traffic and transaction characteristics

As nodes are autonomous, independent, and hidden behind a firewall, NAT, it is hard to gather information to conduct a measurement-based study. The first research goal focused on performing measurement-based research to provide insight into the transaction and traffic characteristics of a blockchain. The first step is to deploy a testbed that enables collecting transactions and traffic data to achieve the second research goal. Such data is essential to investigate the arrival process, block generation behavior, user behavior, and confirmation times. There have been some studies of transaction characteristics and miners' activity. However, these studies used assumptions. No or few works investigated

transaction and traffic-related data to examine the characteristics of the transactions. In the literature, it is more often assumed that transactions and block arrival at the node are Poisson processes, block inter-generation time follows an exponential distribution, and the miners follow the default fee per byte ordering. However, most of these assumptions were not validated with experimental-based analysis. To this end, we conducted an experimental-based analysis to confirm the assumptions.

4.1.2 Research Goal 2:

To investigate the performance of blockchain technology

The first research goal provided extensive insight into the transaction and traffic characteristics of the technology. These results enabled the second research goal to define models, develop simulators/emulators, and deploy a testbed to investigate the performance of the technology.

The primary goal of a blockchain is to provide a platform that allows two unknown parties to perform valid transactions without paying an extra fee. However, because of the unstructured P2P network, the end-to-end delay, miner incentives, P2P network protocol, and load in the network affect quality of the service a user gets. These network and internal workflow parameters significantly affect the performance of the technology. The second research goal focuses on investigating the impact of these parameters on the transaction characteristics of the technology. For instance, these parameters affect the amount of time a transaction has to wait before receiving confirmation. The research community is still debating increasing the block size to reduce the average wait time experienced by transactions. However, the other parameters have been ignored as they influence less in the overall activity. This work aims to bring new insight into how internal and external factors affect the wait time.

4.1.3 Research Goal 3:

To study the suitability of blockchain for addressing security, privacy, and performance in networking

The third research goal focuses on exploring the applicability of blockchain to networking to enhance security, privacy, and performance. This consideration lacks a state-of-the-art study on the applicability and possible setbacks of blockchain. We analyzed the suitability of blockchain in networking and the challenges that may come with such considerations. We also highlighted how blockchain can be used in networking as a service provider for information management or as a second-tier security mechanism through record-keeping.

4.2 Research Scope

In this work, most of our research analysis focuses on using Bitcoin as the primary source for the study. This is because it is the first application that

used blockchain technology. We used a public permissionless blockchain. Many applications use blockchain other than Bitcoin, but most of these applications came after Bitcoin, and most of these applications follow a similar workflow. Most of the studies and results are applicable to other applications.

4.3 Research Methodology

The research started with the deployment of an independent live full Bitcoin node. Fig. 4.1 shows the deployment of blockchain technology to perform an experimental analysis, define models, and develop simulators/emulators. The proposed models study the dynamic behaviors of blockchains. Most research articles extracted the necessary information or attributes from the testbed infrastructure. The extracted features are used to investigate the impact and support of networking in blockchains, such as the number of transactions per second, size of the blocks, inter-arrival rate, P2P formation strategy, topology, and amount of traffic generated by the blockchain for mining. The information gathered contains quantitative and qualitative data.

Quantitative data is used to analyze the capability of the technology. This types of data examines aspects of the various blockchains, including stability and maintainability. We also explored the applicability of blockchain to networking, such as possible setbacks and challenges. Similarly, we shed light on the use blockchain as information management and second-tier security mechanisms. This led us identify the open challenges that form the basis for defining the future goal. Below are the main topics and methods employed during the research study.

4.3.1 Measurements-based study

The structure of the distributed ledger technology, blockchain, makes it challenging to investigate the transaction and traffic characteristics of the technology. This enforces the first deployment of an independent Bitcoin node to collect transactions and traffic characteristics to perform an experimental analysis. The insight gained from the experimental analysis was used as the input to define models and develop simulators to investigate the impact of internal (such as end-to-end delay and P2P topology) and external (such as miner incentives) factors on the overall performance of the technology. The results of experimental and defined models were compared for cross-validation. Fig. 4.1 illustrates the research methodology employed in this work.

4.3.2 Defining models to examine blockchain traffic and transaction characteristics

Although Bitcoin has been around for more than a decade, the number of studies that investigated the technology' performance is very few. Three choices can be

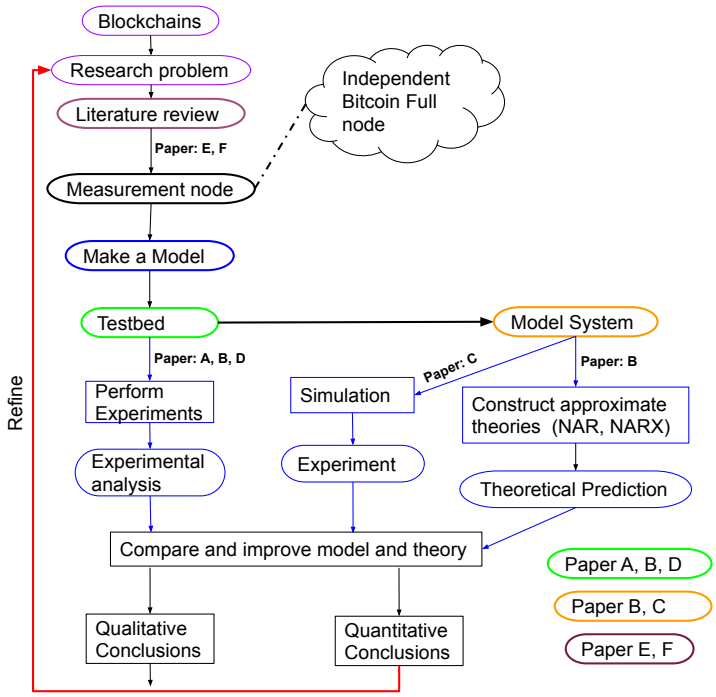


Figure 4.1: Research methodology

used to evaluate the performance: (i) analytic or mathematical, (ii) simulation, and (iii) machine learning models.

Analytic approaches such as Markov models can be solved mathematically, but the limitation is that the model may fall short of capturing detailed information about the technology. Pure state models cannot capture the dynamism without the models becoming excessively complex. Developing a simulation/emulator to realize the Bitcoin workflow has an advantage over the classical queuing model. To this end, we developed a simulator to capture the internal interactions and consider some new details, such as the weak dependency between the fee and the size. Additionally, machine learning models have been used to predict short-term transaction characteristics. However, linear machine learning models were found to be less useful for examining traffic predictions [71]. Nonlinear models show better performance. Nonlinear machine learning models such as NAR, NARX, ARIMA, and ARIMAX have been used to predict the short-term evolution of the ledger.

Chapter 5

Contributions

In this section, the contributions of papers that constitute the thesis are discussed in Section 5.1. Then, a brief discussion on how these papers relate to the research goals is presented in Section 5.2. Lastly, the limitation and applicability of the developed framework and results from the studies are discussed in Section 5.3.

5.1 Contributions of the papers

The published papers are listed below. Fig. 5.1 shows the papers' connection to address the research goals. Papers A and B address the first research goal. Paper A is a testbed deployment that examines the blockchain's fundamental characteristics, providing new insight into the blockchain workflow. Papers B and C are an extended analysis of the Paper A exploration. Paper B proposes uses an existing supervised machine learning model to predict block size and the number of transactions within the throughput. Furthermore, it uses the models to detect the strategy of major mining pools in Bitcoin. Paper C develops a simulator/emulator to determine the wait time of low-fee transactions, while critical parameters such as block size, scheduling discipline, and weak dependence between transaction fee and size.

Paper D examines the impact of P2P formation, fork occurrence, and effect on the confirmation time of transactions. It is an extended paper from an observation in Paper A about the possibility of a fork occurrence and miners' peer formation strategy.

Papers E and F discuss the contribution of blockchains to networking while pointing out possible limitations and setbacks of introducing a blockchain into a complex system in networking, such as 5G, smart grids, and the IoT. Paper E provides insight into the possible limitations of considering public and private blockchains in addressing privacy and security and enhancing performance in networking. In contrast, Paper F highlight to what extent blockchain can be used to improve information management and provide support to enhance

security in smart grids.

As a highlight, the color code refers to which research goal the papers address—for instance, the black papers A, and B address the first research goal. The green papers C and D address the second research goal. Finally, the blue papers E and F address the third research goal. The labels on the arrows from the paper connection represents the paper's extension to address what is observed from the previous paper.

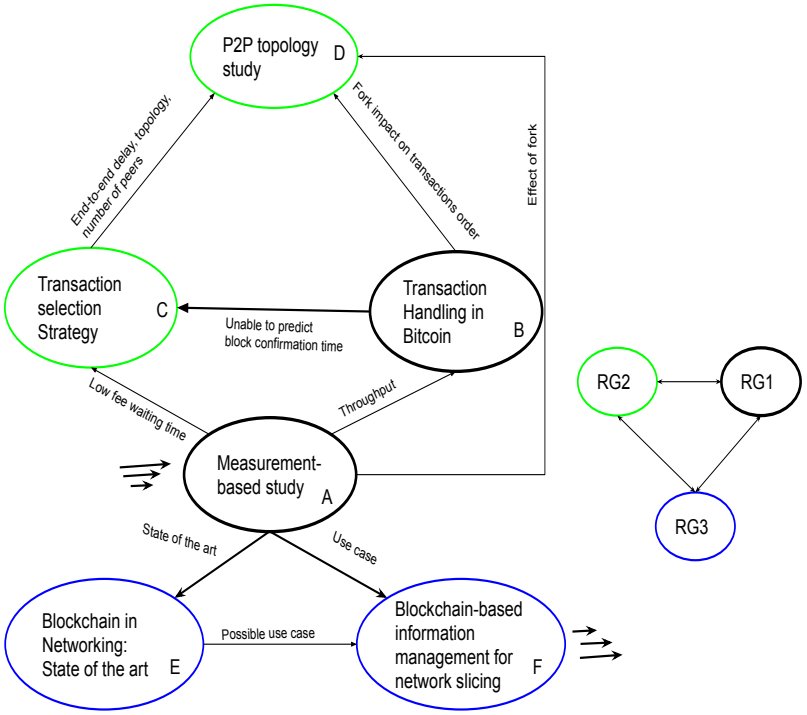


Figure 5.1: Papers connection

Paper A:

Title: **Transaction characteristics of bitcoin**
Authors: *Befekadu G. Gebraselase, Bjarne E. Helvik, Yuming Jiang*
Status: *Published*
Venue: *IM 2021*

This paper presents a measurement-based study that collects data about the ledger and memory pool through a testbed deployment. The collected datasets are used to investigate the block size and inter-arrival time distributions, memory pool arrival characteristics, and the relationships between the transaction

attributes. The results provide insight into the transaction and traffic characteristics of Bitcoin. This paper addresses the first research goal.

Abstract: Blockchain has been considered as an important technique to enable secure management of networks and network-based services. To understand such capabilities of a blockchain, e.g. transaction confirmation time, demands a thorough study on the transaction characteristics of the blockchain. This paper presents a comprehensive study on the transaction characteristics of Bitcoin – the first blockchain application, focusing on the underlying fundamental processes. A set of results and finding are obtained, which provide new insight into understanding the transaction and traffic characteristics of Bitcoin. As a highlight, the validity of several hypotheses / assumptions used in the literature is examined with measurement for the first time.

Paper B:

Title: **An Analysis of Transaction Handling in Bitcoin**
Authors: *Befekadu G. Gebraselase, Bjarne E. Helvik, Yuming Jiang*
Status: *Published*
Venue: *2021 IEEE International Conference on Smart Data Services (SMDS)*

This paper applies nonlinear autoregressive models to predict short-term transaction and block characteristics. It uses the selected feature set from Paper A, with an additional feature set analysis. Furthermore, it introduces a decision-tree model to detect the two major private mining pools with a hidden block generation strategy. This paper addresses the first research goal.

Abstract: Bitcoin has taken the world by storm. It is the leading electronic, decentralized, cryptocurrency system that removes an intermediary between the participating parties. With an increasing integration demand, the ledger is struggling to provide higher throughput. Block related attributes like the block size, the average fee, the number of transactions per block, mempool size, mining pools behavior, and block inter-arrival times are critical elements that may provide valuable information about the system's throughput. To this aim, we propose nonlinear autoregressive neural network-based models to predict short-term Bitcoin transactions. These forecasts may help understand and capture dynamic fluctuations of the block size and the transactions inside varying over time. Furthermore, we conducted an independent analysis of how some of these feature sets can classify some major mining pools to understand any pattern followed by these mining pools. The developed scheme is tested on a dataset collected from a setup that runs a live Bitcoin full node. The analysis shows that the nonlinear autoregressive (NAR) and nonlinear autoregressive with exogenous inputs (NARX) based prediction and decision-tree based classifications are suitable for predicting short-term Bitcoin transactions and analyzing the behavior of major mining pools.

Paper C:

Title: **Effect of Miner Incentive on the Confirmation Time of Bitcoin Transactions**
Authors: *Befekadu G. Gebraselase, Bjarne E. Helvik, Yuming Jiang*
Status: *Published*
Venue: *2021 IEEE International Conference on Blockchain (IEEE Blockchain)*

This paper examines the performance of Bitcoin. At the same time, new insights from Paper A and B are considered, such as weak dependence between transaction attributes, transaction characteristics, and miners' incentives. We developed a simulator/emulator to include these facts to examine the impact on the average wait time. This paper addresses the second research goal.

Blockchain is a technology that provides a distributed ledger that stores previous records while maintaining consistency and security. Bitcoin is the first and largest decentralized electronic cryptographic system that uses blockchain technology. It faces a challenge in making all the nodes synchronize and have the same overall view with the cost of scalability and performance. Furthermore, with miners' financial interest playing a significant role in choosing transactions from the backlog, small fee or small fee per byte value transactions will exhibit more delays. To study the issues related to the system's performance, we developed an $M(t)/M^N/1$ model. The backlog's arrival follows an inhomogeneous Poison process while the backlog has infinite buffer capacity, and the service time is distributed exponentially, which removes N transactions at times. Besides this, we used the model to study the reward distribution when miners choose transaction selection from fee per byte, fee-based, and FIFO. The analysis shows that the smaller transactions exhibit higher confirmation times, even with increasing the block size. Moreover, the miner transaction selection strategy impacts the final gain.

Paper D:

Title: **Bitcoin P2P Network Measurements: A testbed study of the effect of peer selection on transaction propagation and confirmation times**
Authors: *Befekadu G. Gebraselase, Bjarne E. Helvik, Yuming Jiang*
Status: *Accepted*
Venue: *2022 IEEE Transactions on Network and Service Management*

This paper deploys a testbed to examine the impact of end-to-end delay, P2P formation strategies, and fork occurrence on the performance of the technology. It is motivated by the lack of an independent testbed to study network-related parameters' on Bitcoin from Papers A, B, and C. The analysis shows how the transaction propagation and confirmation time are affected by different network conditions. This paper addresses the second research goal.

Bitcoin is the first and the most extensive decentralized electronic cryptocurrency system that uses blockchain technology. It uses a peer-to-peer (P2P) network to operate without a central authority and propagate system information such as transactions or blockchain updates. The communication between participating nodes is highly relying on the underlying network infrastructure to facilitate a platform. Understanding the impact of peer formation strategies, peer list, and delay is vital in understanding node to node communication and the system performance. Therefore, we performed an extensive study on the transaction characteristic of Bitcoin through a testbed. The analysis shows that peer selection strategies affect the transactions propagation and confirmation times. In particular, better performance, in terms of smaller transaction confirmation time and lower number of temporary forks, may be achieved by adjusting the default nearby-based peer selection strategy.

Paper E:

Title: **Suitability of blockchains to enable and support networking functions: State of art**
Authors: *Befekadu G. Gebraselase, Bjarne E. Helvik, Yuming Jiang*
Status: *Published*
Venue: *International Conference on Cloud Computing and Internet of Things (CCIOT), 2019*

This paper provides a literature review of the current state of the art in considering blockchain in networking and the possible setbacks. The paper highlights introducing blockchain in networking to resolve limitations in performance and security. The possible setbacks and limitations of considering public, private, and consortium blockchains are highlighted. This paper addresses the third research goal.

Abstract: The underlying network infrastructure faces challenges from addressing maintenance, security, performance, and scalability to make the network more reliable and stable. Software-defined networking, blockchain, and network function virtualization were proposed and realized to address such issues in both academic and industry wise. This paper analyzes and summarizes works from implementing different categories of blockchains as an element or enabler of network functions to resolve the limitation. Blockchain as a network function has been proposed to give support to the underlying network infrastructure to provide services that have less lag, are more cost-effective, have better performance, guarantee security between participating parties, and protect the privacy of the users. This paper provides a review of recent work that makes use of blockchain to address such networking related challenges and the possible setbacks in the proposal.

Paper F:

Title: **Blockchain-based information management for network slicing**
Authors: *Befekadu G. Gebraselase*
Status: *Published*
Venue: *International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), 2021*

This paper is an extension of Paper E, looking for where blockchain can be relevant. It introduces blockchain into the new emerging 5G and beyond generations. Many recent works focus on network slice isolation and sharing. None or few works have investigated blockchain as a use case to provide high-level isolation and sharing. Thus, we proposed blockchain for information management in network slicing. This paper addresses the third research goal.

Abstract: Network slicing is the crucial enabler of the new emerging 5G and beyond network generations. It facilitates the facility to compose logical networks over shared physical infrastructures, from the implementation perspective of view slice isolations and sharing, become very challenging. It introduces challenges to provide secure information to the subscribers and enables users to modify and configure the registrations while following the service level agreement. To this aim, we introduce a blockchain as a service, in which the distributed ledger technologies provide security and accessibility to the end-users while removing a third-party involvement. Additionally, it allows tenants and subscribers to manage the slice information as necessary without violating the agreement. The primary advantage of including blockchain in architecture is using it to slice isolation and sharing.

5.2 Research contributions

- Contribution 1: The first contribution of this thesis is the extensive analysis of the transaction characteristics of the first blockchain application, Bitcoin (paper A). This analysis provided insight into the most fundamental processes of the technology. The inter-block and inter-transaction arrival process to a node is exponentially distributed with noticeable deviation. The inter-block generation time also follows an exponential distribution, likely attributed to major miners' exponentially distributed inter-block generation time. In addition, the number of transactions inside a block and the block size can vary in different periods.

- Contribution 2: The observations from Contribution 1 are used as input to examine how transactions are handled in Bitcoin (paper B). Through a comprehensive study on how transactions are handled, the analysis shows that machine learning modes could not predict the short-term values of block generation and transactions confirmations time. This is because inter-block generation time and the transaction confirmation time are

closely approximated by an exponential distribution, in which the memoryless property of the probability of some future event occurring has little relation to whether it has happened in the past. However, the models accurately predicted the number of transactions within a block and blocked size. In addition, F2Pool, a major mining pool, its block generation has a distribution that is well distinguished by classifier than the others. Implies F2Pool may use different strategies than the other miners and mining pools.

- Contribution 3: The observations from Contribution 1 and Contribution 2 are used as input to examine the impact of miner incentives on the confirmation time of a transaction (paper C). The model is comprehensive as it enables to mimic Bitcoin properly so that different transaction selection strategies are employed to observe the effect of financial incentives. The analysis showed that miners' financial incentives in picking up hefty transactions affect transaction confirmation times, especially low fee transactions with longer waiting times. In addition, miners' transaction selection strategy impacts the final reward. When all miners follow the same strategy, the reward distribution is equal.

- Contribution 4: The observations from Contribution 1 and Contribution 3 are used as input to examine the impact of P2P formation strategies, arrival intensity, and node-to-node delay on the overall performance (paper D). This study provided a testbed to study network conditions' effect on transaction characteristics. In addition, it shows that adding random peers to the peer formation strategies improves transaction propagation and confirmation times. The fork occurrence impacts transactions confirmation to get higher than 5000 seconds, which is 1400 seconds more elevated than the expected 3600 seconds.

- Contribution 5: The fifth contribution is the study on the suitability of blockchains to enable and support network functions (paper E). The study showed that public blockchains are not fit enough to be used in domains that require stringent QoS requirements. The study also highlights the possible outcomes of considering different types of blockchains in time-sensitive, high throughput, and heterogeneous system requirements.

- Contribution 6: Contribution 5 provided a state-of-the-art study on the suitability of blockchain in networking. This observation is used as an input to provide possible use cases where blockchain can contribute to networking as information management in network slicing (paper F).

These six contributions are purposed to address the three research goals of the Ph.D. work. Specifically, contributions 1-2 answer the first research goal (RG1), and contributions 3-4 address the second research goal (RG2). The third research goal (RG3) is tackled by contributions 5 and 6.

5.3 Methodological consideration

The contribution listed above depends on the input, assumption, and methods we used during this Ph.D. study. This section addresses the limitations and relevance of the results and contributions.

5.3.1 Discussion on RG1

Paper A and B address research goal 1. Paper A analysis depends on the live Bitcoin node we deployed to collect transactions and traffic characteristics from the Bitcoin network. Since our node is in Norway, whereas the rest of the nodes are distributed globally, the observation seen from our node may differ if the node is in a different content/country. Similarly, paper B performed explanatory and predictive analysis from the data collected from the paper A testbed. Thus, the result and insight gained from this work depend on the node. However, the critical characteristic observed from our node is expected to happen in the other nodes in the network. Thus, even though node distribution and user demands may have some effect, the technology's fundamental transaction and traffic characteristics become the same.

5.3.2 Discussion on RG2

Paper C and D address research goal 2. Paper C developed an emulator that captures Bitcoin workflow to study the impact of miner incentives on the confirmation time. The fundamental process characteristic is gained from the RG1, in which the properties are observation dependent. In addition, the emulator the whole bitcoin network as a single node, which means the model does not consider adding network conditions. However, paper D considered this limitation and prepared a testbed to study the impact of network conditions on the overall performance. The testbed contains 104 nodes, whereas the real Bitcoin network has 6000 - 7000 nodes, a significant number difference; however, the 104 nodes generate the same number of blocks and transactions on average. However, this also makes the result to be optimistic to represent the whole Bitcoin nodes.

5.3.3 Discussion on RG3

Paper E and F address research goal 3. Paper E summarizes the potential limitation of using blockchain in complex and heterogeneous systems. Based on this observation, paper F proposed a use case where blockchain can be used to provide support to network functionalities. Paper F is a conceptual article that proposes blockchain to manage information between participating units but lacks a study on its impact on performance and dependability.

Chapter 6

Concluding Remarks

6.1 Conclusion

Blockchain has been considered a disruptive technology that provides a decentralized, distributed ledger that removes intermediaries between participating units while guaranteeing security and consistency. These key attributes have attracted significant attention from different domains, such as networking, health, finance, and energy.

This thesis conducted an extensive analysis of the capability of blockchain. As blockchain is an evolving technology, very few related works have investigated the fundamental relationship between blockchain and networking. This thesis provided significant insight into the fundamental characteristics of blockchain, such as transaction inter-arrival, block inter-generation, miner incentives, and the applicability of blockchain. Based on the studies, the public blockchains are not yet suitable for use in areas where stringent QoS requirements are needed.

Moreover, the P2P communication protocol spends significant time on validation and processing, reducing the overall performance. This becomes worse when the network becomes congested with inbound and outbound traffic. Each node pushes the new update and receives updates from other nodes, making the backlog clogged with arrivals waiting for pick up. Increasing the network load impacts the transaction confirmation and fork occurrence.

Miners play a significant role in validating and processing transactions. The fair distribution of reward and responsibility is affected by the lack of rules and policy settings to control miner incentives and motives. This directly affects the QoS provided to the user. For instance, when many arrivals wait at the mempool, the miners prioritize transactions with a higher fee, making smaller fee trasaction wait longer than expected. In addition, the trend of joining the mining pool to earn a reward affects the decentralization ledger, which is controlled by a few mining pools. It is essential to set rules and procedures that all miners follow to maintain consistency and stability.

Blockchain must solve fundamental issues such as latency, fork occurrence,

miners' incentives and reduce the long wait times as a technology to be considered in heterogeneous systems. Although there have been incentives in proposing methodology and techniques to improve the performance, most of these proposals lack to consider how the memory pool is managed, why the P2P protocol invests significant time to validate the transactions, or how to improve the P2P formation strategy. Thus, without addressing these fundamental questions, the public blockchain becomes short to be used in complex systems, such as 5G and smart grids.

6.2 Future direction

There are many directions where this research can continue. One of the most important is to develop distributed backlog management algorithms that guarantee every transaction has some chance of being confirmed. The second direction is to investigate or propose a better P2P communication protocol that can optimize the extra workload done by the nodes to validate the same transaction many times. Finally, for blockchain to be used in a complex system, it must improve its scalability and performance limitations. In this regard, few works propose sharding, compact block, and others to improve the propagation time. However, it requires an independent investigation of how scalable it becomes by introducing DNS servers that act as a distribution relay between nodes. As we demonstrated in this thesis, adding some random nodes may help. This also needs attention since figuring out which node to add randomly can compromise the security of the technology. Therefore, it requires detailed knowledge and expertise to improve the peer selection strategies.

Bibliography

[1] A. Abdennour. "Evaluation of neural network architectures for MPEG-4 video traffic prediction". In: *IEEE Transactions on Broadcasting* 52.2 (2006), pp. 184–192.

[2] A. M. Adas. "Using adaptive linear prediction to support real-time VBR video under RCBR network service model". In: *IEEE/ACM Transactions on Networking* 6.5 (1998), pp. 635–644.

[3] A. Aggarwal et al. "Deep Learning Approach to Determine the Impact of Socio Economic Factors on Bitcoin Price Prediction". In: *2019 Twelfth International Conference on Contemporary Computing (IC3)*. 2019, pp. 1–5.

[4] R. Albariqi and E. Winarko. "Prediction of Bitcoin Price Change using Neural Networks". In: *2020 International Conference on Smart Technology and Applications (ICoSTA)*. 2020, pp. 1–4.

[5] T. Ali Syed et al. "A Comparative Analysis of Blockchain Architecture and its Applications: Problems and Recommendations". In: *IEEE Access* 7 (2019), pp. 176838–176869.

[6] I. D. Alvarenga, G. A. F. Rebello, and O. C. M. B. Duarte. "Securing configuration management and migration of virtual network functions using blockchain". In: *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*. 2018, pp. 1–9.

[7] Moshe Babaioff et al. "On Bitcoin and Red Balloons". In: EC '12. Valencia, Spain: Association for Computing Machinery, 2012, pp. 56–73. ISBN: 9781450314152.

[8] R. Banno and K. Shudo. "Simulating a Blockchain Network with SimBlock". In: *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. 2019, pp. 3–4.

[9] I. Safak Bayram et al. "A survey on energy trading in smart grid". In: *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. 2014, pp. 258–262.

[10] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. "Deanonymisation of Clients in Bitcoin P2P Network". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. CCS '14. Scottsdale, Arizona, USA: Association for Computing Machinery, 2014, pp. 15–29.

[11] Bitcoin. *bitcoin blocks*. URL: `https://bitcoincore.org/en/2016/06/07/compact-blocks-faq/`. (accessed: 02.11.2021).

[12] Bitcoin. *channel*. URL: `https://en.bitcoin.it/wiki/Lightning_Network`. (accessed: 01.01.2021).

[13] Bitcoin. *merkel tree*. URL: `https://github.com/bitcoin/bips/blob/master/bip-0114.mediawiki`. (accessed: 01.01.2021).

[14] Bitcoin. *merkel tree*. URL: `https://en.bitcoin.it/wiki/Atomic_swap`. (accessed: 01.01.2021).

[15] Bitcoin. *mining guide*. URL: `https://github.com/bitcoin/bips/`. (accessed: 01.01.2021).

[16] Bitcoin. *mining guide*. URL: https://bitcoin.org/en/mining-guide. (accessed: 01.01.2021).

[17] Bitcoin. *mining guide*. URL: https://en.bitcoin.it/wiki/Block%5C_size. (accessed: 01.01.2021).

[18] A. Boudguiga et al. "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)*. 2017, pp. 50–58.

[19] Nikola Bozic, Guy Pujolle, and Stefano Secci. "A tutorial on blockchain and applications to secure network control-planes". In: *2016 3rd Smart Cloud Networks & Systems (SCNS)*. IEEE. 2016, pp. 1–8.

[20] A. Chauhan et al. "Blockchain and Scalability". In: *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*. 2018, pp. 122–128.

[21] Hash Time Locked Contracts. *channel*. URL: https://en.bitcoin.it/wiki/Hash_Time_Locked_Contracts. (accessed: 01.01.2021).

[22] P. Cortez et al. "Internet Traffic Forecasting using Neural Networks". In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. 2006, pp. 2635–2642.

[23] J. Dai and J. Li. "VBR MPEG Video Traffic Dynamic Prediction Based on the Modeling and Forecast of Time Series". In: *2009 Fifth International Joint Conference on INC, IMS and IDC*. 2009, pp. 1752–1757.

[24] C. Decker and R. Wattenhofer. "Information propagation in the Bitcoin network". In: *IEEE P2P 2013 Proceedings*. 2013, pp. 1–10.

[25] V. B. Dharmadhikari and J. D. Gavade. "An NN approach for MPEG video traffic prediction". In: *2010 2nd International Conference on Software Technology and Engineering*. Vol. 1. 2010, pp. V1-57-V1–61.

[26] Omar Dib et al. "Consortium blockchains: Overview, applications and challenges". In: *International Journal On Advances in Telecommunications* 11.1&2 (2018), pp. 51–64.

[27] Donghui Ding et al. *Txilm: Lossy Block Compression with Salted Short Hashing*. 2019. arXiv: 1906.06500 [cs.CR].

[28] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. "The Bitcoin Backbone Protocol: Analysis and Applications". In: Apr. 2015, pp. 281–310. ISBN: 978-3-662-46802-9.

[29] S. Geissler et al. "Discrete-Time Analysis of the Blockchain Distributed Ledger Technology". In: *2019 31st International Teletraffic Congress (ITC 31)*. 2019, pp. 130–137.

[30] Arthur Gervais et al. "On the Security and Performance of Proof of Work Blockchains". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS '16. Vienna, Austria: Association for Computing Machinery, 2016, pp. 3–16.

[31] Taylor Hardin and David Kotz. "Blockchain in Health Data Systems: A Survey". In: *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*. 2019, pp. 490–497.

[32] S. Harrabi, W. Chainbi, and K. Ghedira. "A multi-agent proactive routing protocol for Vehicular Ad-Hoc Networks". In: *The 2014 International Symposium on Networks, Computers and Communications*. 2014, pp. 1–6.

[33] S. Horovitz and D. Dolev. "Collabrium: Active Traffic Pattern Prediction for Boosting P2P Collaboration". In: *2009 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises*. 2009, pp. 116–121.

[34] Yuheng Huang et al. *Characterizing EOSIO Blockchain*. 2020. arXiv: 2002.05369 [cs.CR].

[35] Huifang Feng and Yantai Shu. “Study on network traffic prediction techniques”. In: *Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing, 2005.* Vol. 2. 2005, pp. 1041–1044.

[36] H. Jang and J. Lee. “An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information”. In: *IEEE Access* 6 (2018), pp. 5427–5437.

[37] Y. Kawase and S. Kasahara. “A Batch-Service Queueing System with General Input and Its Application to Analysis of Mining Process for Bitcoin Blockchain”. In: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 2018, pp. 1440–1447.

[38] Yoshiaki Kawase and Shoji Kasahara. “Transaction-Confirmation Time for Bitcoin: A Queueing Analytical Approach to Blockchain Mechanism”. In: Aug. 2017, pp. 75–88. ISBN: 978-3-319-68519-9. DOI: 10.1007/978-3-319-68520-5_5.

[39] Philip Koshy, Diana Koshy, and Patrick McDaniel. “An Analysis of Anonymity in Bitcoin Using P2P Network Traffic”. In: vol. 8437. Mar. 2014, pp. 469–485.

[40] K. Kotobi and S. G. Bilén. “Blockchain-enabled spectrum access in cognitive radio networks”. In: *2017 Wireless Telecommunications Symposium (WTS)*. 2017, pp. 1–6.

[41] Brijesh Kumar and Jon Crowcroft. “Integrating Security in Inter-Domain Routing Protocols”. In: *SIGCOMM Comput. Commun. Rev.* 23.5 (Oct. 1993), pp. 36–51. ISSN: 0146-4833.

[42] Tran Le Nguyen. “Blockchain in Healthcare: A New Technology Benefit for Both Patients and Doctors”. In: *2018 Portland International Conference on Management of Engineering and Technology (PICMET)*. 2018, pp. 1–6.

[43] Quan-Lin Li, Jing-Yu Ma, and Yan-Xia Chang. “Blockchain Queueing Theory”. In: (Aug. 2018).

[44] Matthias Lischke and Benjamin Fabian. “Analyzing the Bitcoin Network: The First Four Years”. In: *Future Internet* 8 (Mar. 2016).

[45] Sean McNally, Jason Roche, and Simon Caton. “Predicting the Price of Bitcoin Using Machine Learning”. In: Mar. 2018, pp. 339–343.

[46] Sarah Meiklejohn et al. “A Fistful of Bitcoins: Characterizing Payments among Men with No Names”. In: *Commun. ACM* 59.4 (Mar. 2016), pp. 86–93.

[47] R. A. Memon et al. “Modeling of Blockchain Based Systems Using Queuing Theory Simulation”. In: *2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*. 2018, pp. 107–111.

[48] Du Mingxiao et al. “A review on consensus algorithm of blockchain”. In: *2017 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE. 2017, pp. 2567–2572.

[49] Jelena Mišić, Vojislav Misic, and Xiaolin Chang. “On the Benefits of Compact Blocks in Bitcoin”. In: Feb. 2020.

[50] S. G. Motlagh, J. Mišić, and V. B. Mišić. “Impact of Node Churn in the Bitcoin Network”. In: *IEEE Transactions on Network Science and Engineering* 7.3 (2020), pp. 2104–2113.

[51] Ahmed S. Musleh, Gang Yao, and S. M. Muyeen. “Blockchain Applications in Smart Grid–Review and Frameworks”. In: *IEEE Access* 7 (2019), pp. 86746–86757.

[52] Satoshi Nakamoto. “Bitcoin: A Peer-to-Peer Electronic Cash System”. In: *Cryptography Mailing list at https://metzdowd.com* (Mar. 2009).

[53] Dinh C Nguyen et al. *Blockchain for 5G and Beyond Networks: A State of the Art Survey*. 2019. arXiv: 1912.05062 [cs.NI].

[54] Kai Otsuki et al. “Effects of a Simple Relay Network on the Bitcoin Network”. In: *Proceedings of the Asian Internet Engineering Conference*. AINTEC ’19. Phuket, Thailand: Association for Computing Machinery, 2019, pp. 41–46. ISBN: 9781450368490.

[55] D. Pavithran and R. Thomas. “A Survey on Analyzing Bitcoin Transactions”. In: *2018 Fifth HCT Information Technology Trends (ITT)*. 2018, pp. 227–231.

[56] Lukáš Pichl and Taisei Kaizoji. “Volatility Analysis of Bitcoin Price Time Series”. In: *Quantitative Finance and Economics* 1 (Dec. 2017), pp. 474–485.

[57] Sang-Jo Yoo. “Efficient traffic prediction scheme for real-time VBR MPEG video transmission over high-speed networks”. In: *IEEE Transactions on Broadcasting* 48.1 (2002), pp. 10–18.

[58] Mennan Selimi et al. “Towards Blockchain-Enabled Wireless Mesh Networks”. In: CryBlock’18. Munich, Germany: Association for Computing Machinery, 2018, pp. 13–18. ISBN: 9781450358385.

[59] Jun.Kawahara Shoji.Kasahara. “Effect of Bitcoin fee on transaction-confirmation process”. In: *Journal of Industrial and Management Optimization* 15.1547 (2019), p. 365.

[60] E. Sin and L. Wang. “Bitcoin price prediction using ensembles of neural networks”. In: *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. 2017, pp. 666–671.

[61] S. Smetanin et al. “Modeling of Distributed Ledgers: Challenges and Future Perspectives”. In: *2020 IEEE 22nd Conference on Business Informatics (CBI)*. Vol. 1. 2020, pp. 162–171.

[62] Yonatan Sompolinsky and Aviv Zohar. “Secure High-Rate Transaction Processing in Bitcoin”. In: Jan. 2015. ISBN: 978-3-662-47853-0.

[63] R. Srivastava. “Blockchain and transaction processing time using M/M/1 queue model”. In: *International Journal of Recent Technology and Engineering* 7 (Jan. 2019), pp. 399–401.

[64] Harish Sukhwani et al. “Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric)”. In: *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*. IEEE. 2017, pp. 253–255.

[65] Wai Yan Maung Maung Thin et al. “Formal analysis of a proof-of-stake blockchain”. In: *2018 23rd International Conference on Engineering of Complex Computer Systems (ICECCS)*. IEEE. 2018, pp. 197–200.

[66] K. Valtanen, J. Backman, and S. Yrjölä. “Creating value through blockchain powered resource configurations: Analysis of 5G network slice brokering case”. In: *2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. 2018, pp. 185–190.

[67] L. Wan, D. Eyers, and H. Zhang. “Evaluating the Impact of Network Latency on the Safety of Blockchain Transactions”. In: *2019 IEEE International Conference on Blockchain (Blockchain)*. 2019, pp. 194–201.

[68] C. Wang, X. Chu, and Y. Qin. “Measurement and Analysis of the Bitcoin Networks: A View from Mining Pools”. In: *2020 6th International Conference on Big Data Computing and Communications (BIGCOM)*. 2020, pp. 180–188.

[69] C. Wu et al. “A New Forecasting Framework for Bitcoin Price with LSTM”. In: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. 2018, pp. 168–175.

[70] Y. Wu, A. Luo, and D. Xu. “Forensic Analysis of Bitcoin Transactions”. In: *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*. 2019, pp. 167–169.

[71] W. Xu and A. G. Qureshi. “Adaptive linear prediction of MPEG video traffic”. In: *ISSPA ’99. Proceedings of the Fifth International Symposium on Signal Processing and its Applications (IEEE Cat. No.99EX359)*. Vol. 1. 1999, 67–70 vol.1.

[72] Beverly Yang and Hector Garcia-Molina. “PPay: Micropayments for Peer-to-Peer Systems”. In: *Proceedings of the 10th ACM Conference on Computer and Communications Security*. CCS ’03. Washington D.C., USA: Association for Computing Machinery, 2003, pp. 300–310. ISBN: 1581137389.

[73] H. Yang et al. “Blockchain-based trusted authentication in cloud radio over fiber network for 5G”. In: *2017 16th International Conference on Optical Communications and Networks (ICOCN)*. 2017, pp. 1–3.

[74] Q. Zhou et al. “Solutions to Scalability of Blockchain: A Survey”. In: *IEEE Access* 8 (2020), pp. 16440–16455.

[75] Muhammad F. Zia et al. “Microgrid Transactive Energy: Review, Architectures, Distributed Ledger Technologies, and Market Analysis”. In: *IEEE Access* 8 (2020), pp. 19410–19432.

Part II
Included Papers

Paper A:
B. G. Gebraselase, B. E. Helvik and Y. Jiang, "Transaction Characteristics of Bitcoin," 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM), 2021, pp. 544-550.

Transaction Characteristics of Bitcoin

Befekadu G. Gebraselase, Bjarne E. Helvik, Yuming Jiang
Department of Information Security and Communication Technology
NTNU, Norwegian University of Science and Technology, Trondheim, Norway
{befekadu.gebraselase, bjarne, yuming.jiang}@ntnu.no

***Abstract*—Blockchain has been considered as an important technique to enable secure management of networks and network-based services. To understand such capabilities of a blockchain, e.g. transaction confirmation time, demands a thorough study on the transaction characteristics of the blockchain. This paper presents a comprehensive study on the transaction characteristics of Bitcoin – the first blockchain application, focusing on the underlying fundamental processes. A set of results and finding are obtained, which provide new insight into understanding the transaction and traffic characteristics of Bitcoin. As a highlight, the validity of several hypotheses / assumptions used in the literature is examined with measurement for the first time.**

***Index Terms*—Blockchain, Bitcoin, Transaction Characteristics**

I. Introduction

Blockchain has been considered as an important technique to enable secure management of networks and network-based services, such as virtual network functions (VNF) [1] and network slices in 5G and beyond networks [23]. To this aim, understanding the capabilities of the blockchain, e.g. in terms of delay or transaction-confirmation time, is necessary. This naturally demands a thorough study of the transaction characteristics of the blockchain [25], with which, analytical methods (e.g. queueing theory) may be employed to estimate the performance of the blockchain [11] [17].

Surprisingly, even for the first blockchain application, Bitcoin [22], such studies are still limited. Most of the literature studies focus on analyzing the Bitcoin transaction's identity and security impact, such as [2], [12], [14], [15], [16], [19] [21], [24], and [27], while only a few have investigated the transaction and block characteristics. For instance, to motivate an exponentially distributed block inter-generation time, two hypotheses on block generation at each miner have been made, namely Bernoulli trial in [13] and uniform distribution in [26]. However, no existing work has investigated whether exponentially block inter-generation times can be justified by measurements. In addition, among the existing results, e.g. various Bitcoin statistics [9], block propagation delay [7], block arrival process [5], transaction rate and transaction confirmation time [11] [17], most are directly generated or derived from the information carried on the Bitcoin blockchain. However, to obtain a deeper understanding of the transaction characteristics of Bitcoin, such information is not sufficient. For instance, in the literature, Poisson transaction arrival process has been widely assumed, e.g., [11] [17], but due to lack of information on the blockchain about the arrival time of a transaction to a node, the validity of this assumption has never been verified.

978-3-903176-32-4

The objective of this paper is to report results and findings from an extensive study of the transaction characteristics of Bitcoin, which not just provide answers to the above mentioned open questions, but also sheds new light on understanding and studying the capabilities of the Bitcoin blockchain. Specifically, the focus is on the most fundamental processes behind Bitcoin, which include the transaction arrival process, the block generation and arrival processes, and the mining pool process. To this aim, a measurement-based study has been conducted, where a dataset has been gathered which contains both information that is globally available from the Bitcoin blockchain, i.e. the ledger, and information that is not available from the ledger but is measured from the local memory pool (mempool). It is worth highlighting that, among these focused processes, the ledger only has timing information for the block generation process, and for the other processes, local measurements are necessary. Based on the collected data, an exploratory study on the transaction characteristics of Bitcoin has been conducted.

The results and findings, which constitute the main and novel contributions of this paper, are organized and presented from three angles. Firstly, transaction characteristics at the block level, such as block generation, block arrival and block size characteristics, are considered. As a highlight, it is found that, even though the block generation time (at the Bitcoin system level) fits well with an exponential distribution, *the two hypotheses on block generation at each miner are both not justified*. Instead, we find another explanation, which is, *block generation at major miners has exponentially distributed inter-block generation time*. Secondly, transaction level characteristics are focused, which include transaction generation, transaction arrival, transaction size and fee characteristics. Here, *the Poisson transaction arrival assumption is examined*. Thirdly, the dynamics of the mining pool, which underlays the block generation process and relates it to the transaction arrival process, are investigated. In particular, the effect of fee, a fundamental element of Bitcoin as a digital currency, is included. As a highlight, it is found that *the fee-based priority queueing model* assumed in the literature [11] [17] *does not match with the observation*. These results and findings, to the best of our knowledge, has not been previously reported, which provide new insights into understanding the transaction characteristics of Bitcoin.

The rest of this paper is organized as follows. Section II introduces the measurement setup and the collected dataset. After that, Section III introduces results and findings on block level transaction characteristics. Section IV presents results and findings on transaction level characteristics. Following that, in Section V, the dynamics of mempool are focused. Finally, Section VI summarizes the paper.

II. Measurement Setup and Dataset Summary

For the measurement study, a testbed as shown in Fig. 1, has been implemented to record information about Bitcoin transactions. The testbed includes a server installation of a full Bitcoin node.

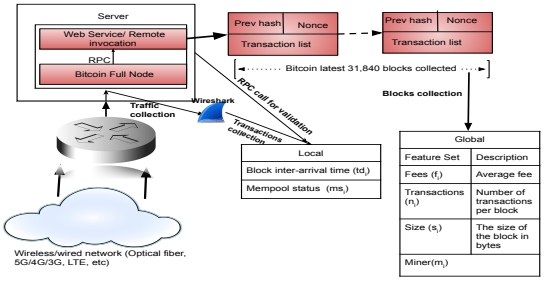


Figure 1. Testbed deployment and dataset attributes

Through the testbed, a dataset, consisting of two parts, has been collected. One part of the dataset records information from the ledger that is globally available, called the *global information part*. Another part records locally available information about each transaction and block as well as the backlog status of the mempool. This part is called the *local information part*. The measurement period of the dataset is from 7th March 2019 till 3rd October 2019, and the dataset consists of over 79 million transactions contained in 31 thousand blocks on the ledger and recorded at the installed full node.

The ledger dataset was collected through a REST API that enables RPC calls to the installed node to collect information about blocks and transactions. The mempool dataset was collected through Wireshark that collects traffic information from the network interface of the node, while RPC calls to the installed node were done to validate that the extracted transaction is available at the mempool. To do so, we used a C++ code to act as a middleman between the installed node and traffic collection from the interface, as demonstrated in Fig. 1.

The recorded information in the global information part of the dataset includes, for each block b on the blockchain, the number of transactions (n_b) in the block, the block generation time (g_b), its miner (m_b), the size of the block (s_b), and the fee (f_b). The locally recorded information from the installed full node includes for each transaction i, the arrival time timestamps (a_i), the transaction fee (f_i), and the size (s_i), and additionally for each block b, its arrival time (a_b). A brief summary of these focused features is also shown in Fig. 1.

In the literature, several platforms provide similar datasets. However, the data extracted from such a source lacks some information that is available in ours. For instance, the set of mempool features, timestamp (a_i), transaction fee (f_i), and size (s_i), which are related to transaction arrivals, are unique in our dataset which generally is not available form the literature platforms. With such information, we can extract the number of bytes that arrive at the mempool in an interval. Additionally, some more detailed information related to each block, which is gathered from the installed full node in our testbed, is not available in the other sources. In particular, in each block, there are many transactions, and each transaction has a number of attributes such as size, fee, and timestamp. Such detailed information cannot be found from outside sources: What is available there is only some piece of general information. Table I provides a comparison of what transaction and block attributes are included in the several well-known platforms and ours, where *IIK testbed* represents our testbed.

Table I
Data source comparison

Dataset	Locally recorded attributes				Block attributes			
	a_i	f_i	s_i	a_b	g_b	f_b	n_b	s_b
Blockstream [4]	×	✓	✓	×	✓	×	✓	✓
Bitaps [3]	×	×	×	×	×	✓	✓	✓
Btc [9]	×	×	×	×	✓	×	✓	✓
Explorers [8]	×	×	✓	×	×	✓	✓	✓
IIK testbed	✓	✓	✓	✓	✓	✓	✓	✓

III. Block-Level Characteristics

In this section, a number of transaction characteristics at the block level are investigated, which are related to block generation and arrival time processes, the number of transactions in a block, and block size.

A. Inter-Block Generation Time

The Bitcoin system uses the UTC +1 zone to synchronize full nodes. Using the same timezone among nodes helps to reduce wrong interpretation or modification of information to a different order. At the generation of a block b, its generation time g_b is added to the block. In this way, the Bitcoin blockchain keeps track of block generations in the system.

Fig. 2 shows that the inter-block generation time of Bitcoin can be excellently matched with a negative exponential distribution, as also reported in the literature [13][26], event though there is some deviation at the tail likely attributed to the very low number of observations in the tail. Additionally, the inter-block generation times are tested for dependencies and none are found

To further find explanation for the exponentially distributed inter-block generation time, we investigate this distribution of each miner. To this aim, the contributions of main miners to block generation is first examined and the results are shown in Fig. 3(a). The figure shows that the majority (80%) are contributed by the few top private miners including Antipool, BTC, BTC.Top, BitFury, F2pool, and viaBtc. In addition, some public mining pools such as Poolin exist, used by

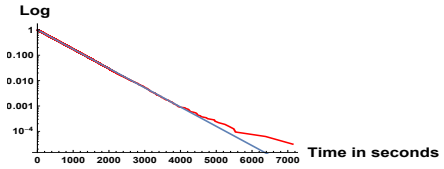
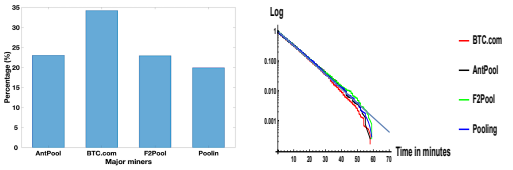


Figure 2. The inter-block generation time and the inter-block arrival time, fit to an exponential distribution

nodes to participate in the mining pool. We also observed that BTC.com, AntPool, F2Pool, and Poolin mining pools contributes the majority of the blocks to the ledger. The number of blocks generated by these pools are not evenly distributed, while the few major miners take most of the valid block.

Fig. 3(a) shows the contribution of the chosen major mining pools and Fig. 3(b) reports their inter-block generation time distribution, normalized to the same mean. In Fig. 3(b) the fit to an exponential distribution, the straight line, is observed, and hence, the inter-block generation time from each mining pool may be well approximated by an exponential distribution. This is different from the two hypotheses found in [13] and [26].



(a) Mining pool contribution in terms of generating blocks

(b) Inter-block arrival time distribution at a miner

Figure 3. Block contribution by miners and per-miner inter-block generation

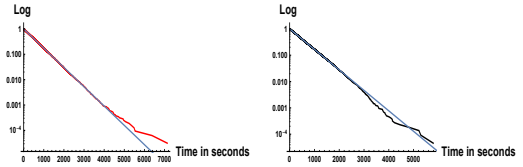
Note that, It is well-known from Palm-Khintchine theorem states that if we combine events from significant, continuous, independent renewal processes, the result will have Poisson properties under certain conditions [18], or in other words, the aggregate point process of independent point processes, each of which has exponentially distributed inter-arrival time, also has exponentially distributed inter-arrival time. It is then worth highlight that the finding in Fig. 3(b) provides a previously unreported explanation for the exponentially distributed inter-block generation time in the Bitcoin system, i.e., it is resulted from similar distributions at the miners.

Finding 1: **The exponentially distributed inter-block generation time on the blockchain is likely attributed to the exponentially distributed inter-block generation time at major miners.**

B. Inter-Block Arrival Time

It is worth highlighting that the block arrival process to a node is different from the block generation process of the Bitcoin system. This is due to that after the generation of a new block, the updated ledger containing the new block needs to be propagated through the Bitcoin network to each node. This causes propagation delay from the generation of each block at its miner to the arrival of the block to a node, $a_b - g_b$.

In the literature, e.g. [7], it has been discussed and conjectured that the block propagation delay is exponentially distributed, but the conjecture is not examined with measurement. We have also performed analysis on the propagation delay with our collected measurement dataset. Based on the arrival time a_b recorded at our node and its generation time g_b, we have found an average of 53 seconds for the block propagation delay. Its distribution is shown in Fig. 4(b). It can be observed from the figure that the block propagation delay well fits an exponential distribution, validating the conjecture in [7].



(a) The block inter-arrival time fitting to a n.e.d

(b) Block propagation delay distribution fitting to a n.e.d

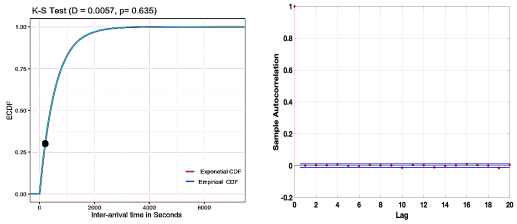
Figure 4. Block arrival time and block propagation delay

For the inter-block arrival time between two adjacent blocks b_i and b_{i+1}, it can be calculated from their arrival times recorded in the local information, i.e. $a_{b_{i+1}} - a_{b_i}$. Its distribution is shown in Fig. 4(a). As can be observed from Fig. 4(a), the distribution can be well approximated by an exponential distribution. This appealing finding can indeed be expected from the distribution of inter-block generation time and the distribution of propagation delay due to the following relationship between them:

$$(a_{b_{i+1}} - a_{b_i}) = (g_{b_{i+1}} - g_{b_i}) + [(a_{b_{i+1}} - g_{b_{i+1}}) - (a_{b_i} - g_{b_i})]$$

where, on the right side, the inter-block generation time $g_{b_{i+1}} - g_{b_i}$ is approximately exponentially distributed as discussed in the previous subsection, and the second term is the propagation delay difference. Since propagation delay is also approximately exponentially distributed, the difference shown as the second term can be approximated to have a Laplace distribution, from the well-known result of difference of two exponentially distributed random variables. Furthermore, from the sum of exponential and Laplace distributions [6], an exponential decay in the inter-block arrival time is expected.

In addition to a K-S test [20] confirming the excellent match, which is shown in Fig. 5(a), we have also examined if blocks arrive independently. This is done by checking the autocorrelation of the block arrival time series, under different time lags. A summary of the autocorrelation values is presented in Fig. 5(b). As can be seen from the table, the autocorrelation is close to zero under all these lags with the largest difference only around 1%, which is an indication that block arrivals are not correlated.



(a) K-S test for block inter-arrival time distribution where D represents the maximum distance between the exponential and empirical CDF

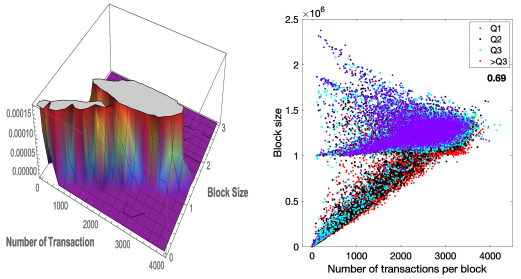
(b) The autocorrelation of block inter-arrival time in seconds

Figure 5. K-S test and autocorrelation

Finding 2: **The block arrival process to a node approximately has an exponentially distributed inter-block arrival time with independent block arrivals, i.e. a homogeneous Poisson process.**

C. Number of Transactions in a Block and Block Size

In contrast to very few results about inter-block generation and arrival time distributions, the literature has a lot of results about n_b, the number of transactions in a block, and s_b, the size of a block, such as those reported for the various platforms [3] [4] [8] [9]. In this and the subsequent subsections, we report results that are either with more detailed information or from new different perspectives.



(a) The joint PDF of n_b and s_b

(b) scatterplot of n_b and s_b

Figure 6. Relation between n_b and s_b

1) Correlation between n_b and s_b: Fig. 6(a) illustrates the joint PDF of n_b and s_b. As we can see from the figure, the dependence between the two variables varies. In general, a larger block has a higher number of transactions included. While this is as expected, Fig. 6(a) details this relationship. In addition, Fig. 6(b) illustrates the scatter diagram of the size of a block, s_b vs. the number of transactions in the block, n_b. It also demonstrates the relationship of s_b and n_b for Q1 (25%), Q2 (50%), Q3 (75%), and greater than Q3 (>Q3) for f_b. These intervals are (0,Q1), (Q1,Q2), (Q2,Q3), and (Q3,∞). As can be observed from Fig. 6(b), there is strong correlation between s_b and n_b. There is a clear pattern shown by the correlation scatter points.

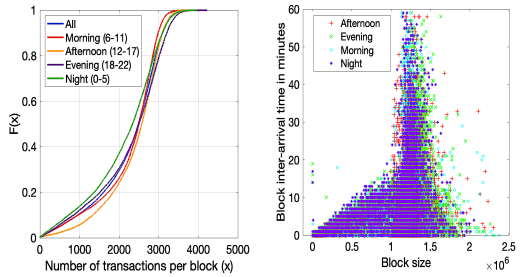
Specifically, it is visible that having a large n_b often implies a higher chance of being in a bigger s_b, as illustrated by blue and bold black dotted blocks when the n_b higher than 2000, even though some high size blocks have a small number of transactions in the block. Sometimes, the number of transactions waiting for confirmations is smaller than the block size capacity; in such cases, we will see blocks filled with fewer numbers than the expected. In Fig. 6(b), we can see the black and red dotted straight line around 0 - 1 MB, indicating generating a block not filled with a maximum capacity as the consequence of the mempool containing a small number of transactions waiting. On the other hand, we can also observe a horizontal line around the s_b 1 - 1.5 MB and where n_b is more significant than 2000, which indicates more transactions waiting while the block filled to the maximum limit. Additionally, we can also see pink and light-green colored blocks with a small number of transactions in a block while the size is pushed to the maximum limit.

Furthermore, we can also observe that the average gain of miners playing a crucial role. The blocks with a higher average fee per block (>Q3) contain a higher gain; on the other hand, most less-filled black and red colored blocks contain less average gain.

Finding 3: **There is positive, strong, and nonlinear relation between the size of a block and its number of transactions.**

D. Characteristics in Different Time Periods

We are interested in finding if and how n_b and s_b may differ in different time periods. As the CDF of n_b reported in Fig. 7(a), in the morning and evening, a block holds on average 2500 transactions, and in the night and afternoon, a block contains no more than 3300 transactions in 90% of cases. Still, in all the cases, it can grow larger than 3500 in 1% of the cases.

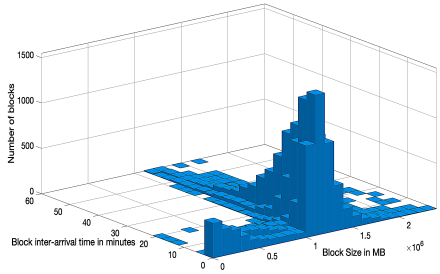


(a) Observed CDF of transactions per block, n_b

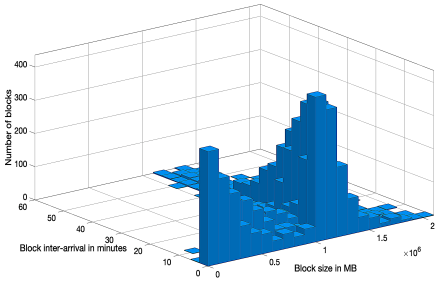
(b) Sctterplot of interarrival times vs. block size, s_b

Figure 7. n_b and s_b characteristics in different time periods

Fig. 7(b) reports that s_b's having values that varies with different time periods . In the afternoon and evening, the s_b's ranges are higher than in the morning and night. The s_b in the evening is relatively larger than in other periods. This may be due to a higher n_b in the evening. In the morning and evening, the number of blocks are generated less frequently, i.e. with



(a) Working days



(b) Weekend

Figure 8. Block generation in working and weekend days

higher inter-block generation time shown in the figure, than the rest of the day.

Fig. 8(a) and Fig. 8(b) further show how s_n's distribution dependent on the interarrival time varies over working and weekend days. In the working days, the s_n is more concentrated over the range of 1 to 1.5 MB, and there are 9229 blocks arrival with an inter-generation time of less than 5 minutes. However, in the weekend days, s_n stands between 0.2 to 1.8 MB, and about 3700 blocks are found with an inter-generation time of less than 5 minutes.

Finding 4: **The characteristics of block size and number of transactions can differ significantly in different time periods.**

IV. Transaction-Level Characteristics

In Bitcoin's design, a transaction confirmation time of 10 minutes is inherent [22]. Based on the arrival time a_b recorded at our node and its generation time g_b, we have found that on average a transaction needs 600 seconds (T_w) from it is received by the Bitcoin system till the corresponding block is generated, i.e. the transaction is confirmed then. This confirms the design principle of Bitcoin.

In the remainder of this section, we focus on the transaction arrival process itself, which is characterized by transactions' inter-arrival times, and the size and fee of each transaction. Fig. 9 provides a trace of this process, where 5000 unique transaction arrivals are ordered based on their arrival times a_i recorded at our full node.

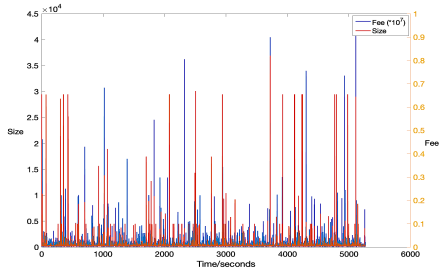


Figure 9. An overview of the transaction arrival process

A. Transactions' Inter-Arrival Time

In the literature, it is often assumed that the transaction arrival process is a Poisson process. However, the validity of this assumption was not examined previously. To bridge this gap, a random period in the dataset was picked, which consists of 1861 transactions, and the inter-arrival time distribution of these transactions is illustrated in Fig. 10.

As we can see from Fig. 10, the transactions' inter-arrival times can be approximately fitted with an exponential distribution, which partially supports the Poisson arrival assumption. However, the figure also shows noticeable deviation. While the deviation for the CCDF value below 1% may be attributed to the number of samples in this fitting test, the derivation is also visible for CCDF above 1%, which can hardly be found in the inter-block generation time and inter-block arrival time curves in Fig. 2 and Fig. 4(a).

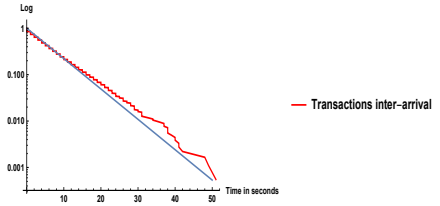


Figure 10. Distribution of transaction inter-arrival times, fitted with n.e.d

Finding 5: **The transaction inter-arrival time may be approximated by an exponential distribution, but with noticeable deviation.**

B. Transaction Size and Fee

According to the design of Bitcoin [22], how a miner selects transactions to form a block depends on the sizes s_i and fees f_i of transactions in the mempool. In Fig. 9, an overview of them with regard to each transaction has been shown. To have a better understanding of them, we investigate their distributions and the correlation between them.

Fig. 9 shows that transaction size and fee do not seem to exhibit a clearly visible, strong positive correlation. While some of the low fee transactions have high sizes s_i, we can also

see transactions with higher fees having smaller transaction sizes. To gain a more complete view, the joint distribution of s_i and f_i is investigated. For the same transactions shown in Fig. 9, the joint distribution result is shown in Fig. 11.

Fig. 11 indicates that 90% of the transactions have a size of not more than 500 bytes. But, in 1% of the cases, the transaction size can be more than 30 kilobytes. Similarly, the fee associated with each transaction is below 0.0006 BTC 90% of the time, but it can grow higher than 0.001 BTC in 1% of the cases. The distribution shows that while there are a lot of small transactions, there is a significant fraction of tens and hundreds of transactions with a higher fee. Fig. 11 also confirms that the correlation between transaction size and fee is weak.

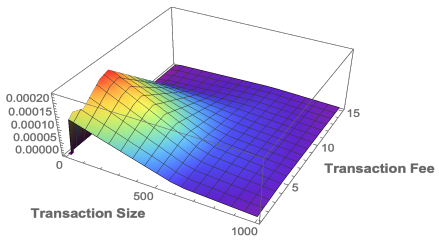


Figure 11. The joint PDF of s_i and f_i

Finding 6: **The correlation between the size of a transaction and its fee is weak.**

V. Mining Pool Dynamics

In this section investigates the dynamics of the memory pool (mempool), which is affected by the transaction arrival process and underlays the block generation process.

A trace of the mempool size in terms of bytes and accumulated fee over ten-block formations is shown in Fig. 12. The x-axis represents arrival times of the blocks, and the y-axis the accumulated entry size and fee, where the fee is scaled for better visibility. Each vertical descent in the size curve represents a new block formation and the height of the descent implies the total size of transactions included in the block, i.e. the size of the block. The corresponding vertical descent in the fee curve represents the fee of the block.

As indicated by Fig. 12, the relationship between block size and block fee is not linear: a bigger block does not guarantee a higher fee and vice versa. When adding transactions into a block, higher priority may be given to the fee than to the number of transactions waiting for confirmation. For instance, we have observed there were often 5000 - 15000 transactions waiting, while the blocks consider fee rather than the mempool size. It is also visible that the mempool state has a fee close to zero at two times, implying that most transactions by then have been confirmed. However, we have also observed that these are low fee transactions that have to wait even longer time to be processed. If a transaction has a bigger size and small fee combination, it may occupy the memory space for a longer time before confirmation.

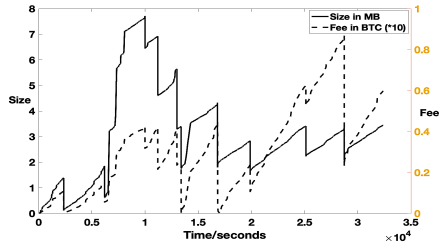


Figure 12. The mempool state change at block generation

For the same reason, in the literature, a fee-based priority queueing model has been simply assumed for the mempool [4] [5]. However, this assumption is too coarse to explain what are shown in Fig. 9 and Fig. 12. For instance, Fig. 12 shows some blocks contain only a few transactions while the block size is filled to the maximum, implying that in these cases, transaction size seems to have been prioritized rather than fee.

Finding 7: **A simple fee-based priority queueing model cannot well capture the dynamics of the mempool.**

VI. Conclusion

Through analyzing the data collected from a measurement setup, which contains transaction and block information both on the blockchain and from the node, we presented a comprehensive study on the transaction characteristics of Bitcoin. A set of new results and findings have been reported, including examining the validity of several hypotheses / assumptions used in the literature.

Specifically, for exponentially distributed inter-block generation / arrival times, we found that the two literature hypotheses cannot be justified by the measurement, and it is likely attributed to exponentially distributed block generation at major miners. In addition, for transaction inter-arrival time, though its distribution may be approximated with an exponential distribution, there is noticeable deviation. Besides, for characterizing the mining pool, no convincing evidence has been found to support the fee-based priority queueing model. Furthermore, while the size of a block and the number of transactions in it exhibit a strong functional relationship dependent on the size and value of the mempool, transaction size and fee seem to be more independent.

As a highlight, the idea of involving the mempool in the measurement, in addition to the commonly used ledger information, has enabled us to study the transaction characteristics of Bitcoin and find the fundamental relationships among the core features. As a future work, we will investigate how to exploit this idea to manage the mempool to improve the throughput and reduce transaction waiting time while keeping the current block size limit.

For more discussion and results, such as Bitcoin workflow and details of various distribution fitting results, they can be found from an extended version of this paper [10].

References

[1] I. D. Alvarenga, G. A. F. Rebello, and O. C. M. B. Duarte. "Securing configuration management and migration of virtual network functions using blockchain". In: *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*. 2018, pp. 1–9.

[2] Alex Biryukov, Dmitry Khovratovich, and Ivan Pustogarov. "Deanonymisation of Clients in Bitcoin P2P Network". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. CCS '14. Scottsdale, Arizona, USA: Association for Computing Machinery, 2014, pp. 15–29.

[3] Bitaps. *Today bitcoin blocks*. URL: https://bitaps.com/blocks. (accessed: 01.07.2020).

[4] Blockstream.info. *Recent Transactions and blocks*. URL: https://blockstream.info/tx/recent. (accessed: 01.07.2020).

[5] R. Bowden et al. "Modeling and analysis of block arrival times in the Bitcoin blockchain". In: *Stochastic Models* 0.0 (2020), pp. 1–36.

[6] Craig Cahillane. "Sum of exponential and laplace distributions". In: *Preprint, California Institute of Technology*. June 2020.

[7] Christian Decker and Roger Wattenhofer. "Information Propagation in the Bitcoin Network". In: *13th IEEE International Conference on Peer-to-Peer Computing*. 2013.

[8] Explorer. *Blockchain Explorer*. URL: https://www.blockchain.com/explorer. (accessed: 01.07.2020).

[9] Btc Block Explorere. *Block Explorer*. URL: https://btc.com/. (accessed: 01.07.2020).

[10] Befekadu G. Gebraselase, Bjarne E. Helvik, and Yuming Jiang. "Transaction Characteristics of Bitcoin". In: *CoRR* abs/2010.10858 (2020). URL: https://arxiv.org/abs/2010.10858.

[11] S. Geissler et al. "Discrete-Time Analysis of the Blockchain Distributed Ledger Technology". In: *2019 31st International Teletraffic Congress (ITC 31)*. 2019, pp. 130–137.

[12] Arthur Gervais et al. "On the Security and Performance of Proof of Work Blockchains". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS '16. Vienna, Austria: Association for Computing Machinery, 2016, pp. 3–16.

[13] Johannes Göbel et al. "Bitcoin Blockchain Dynamics: The Selfish-Mine Strategy in the Presence of Propagation Delay". In: *Performance Evaluation* 104 (May 2015).

[14] W. Guo and J. Zhang. "Towards Tracing Bitcoin Client using Network Traffic Analysis". In: *2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP)*. 2019, pp. 1–5.

[15] Y. Guo, J. Tong, and C. Feng. "A Measurement Study of Bitcoin Lightning Network". In: *2019 IEEE International Conference on Blockchain (Blockchain)*. 2019, pp. 202–211.

[16] Yuheng Huang et al. *Characterizing EOSIO Blockchain*. 2020. arXiv: 2002.05369 [cs.CR].

[17] Yoshiaki Kawase and Shoji Kasahara. "Transaction-Confirmation Time for Bitcoin: A Queueing Analytical Approach to Blockchain Mechanism". In: Aug. 2017, pp. 75–88.

[18] A. Y. Khinchine. *Mathematical Methods in the Theory of Queueing*. London: Griffin, 1960.

[19] Philip Koshy, Diana Koshy, and Patrick McDaniel. "An Analysis of Anonymity in Bitcoin Using P2P Network Traffic". In: vol. 8437. Mar. 2014, pp. 469–485.

[20] F. J. Massey. "The Kolmogorov-Smirnov Test for Goodness of Fit". In: 46.253 (1951), pp. 68–78.

[21] Sarah Meiklejohn et al. "A Fistful of Bitcoins: Characterizing Payments among Men with No Names". In: *Commun. ACM* 59.4 (Mar. 2016), pp. 86–93.

[22] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". In: *Cryptography Mailing list at https://metzdowd.com* (Mar. 2009).

[23] Dinh C Nguyen et al. *Blockchain for 5G and Beyond Networks: A State of the Art Survey*. 2019. arXiv: 1912.05062 [cs.NI].

[24] D. Pavithran and R. Thomas. "A Survey on Analyzing Bitcoin Transactions". In: *2018 Fifth HCT Information Technology Trends (ITT)*. 2018, pp. 227–231.

[25] Daniel Perez, Jiahua Xu, and Benjamin Livshits. "Revisiting Transactional Statistics of High-scalability Blockchain". In: *ACM Internet Measurement Conference (IMC)*. 2020.

[26] Jun.Kawahara Shoji.Kasahara. "Effect of Bitcoin fee on transaction-confirmation process". In: *Journal of Industrial and Management Optimization* 15.1547 (2019), p. 365.

[27] Y. Wu, A. Luo, and D. Xu. "Forensic Analysis of Bitcoin Transactions". In: *2019 IEEE International Conference on Intelligence and Security Informatics (ISI)*. 2019, pp. 167–169.

Paper B:
B. G. Gebraselase, B. E. Helvik and Y. Jiang, "An Analysis of Transaction Handling in Bitcoin," 2021 IEEE International Conference on Smart Data Services (SMDS), 2021, pp. 162-172, doi: 10.1109/SMDS53860.2021.00030.

An Analysis of Transaction Handling in Bitcoin

Befekadu G. Gebraselase, Bjarne E. Helvik, Yuming Jiang
Department of Information Security and Communication Technology
NTNU, Norwegian University of Science and Technology, Trondheim, Norway
{befekadu.gebraselase, bjarne, yuming.jiang }@ntnu.no

***Abstract*—Bitcoin has become the leading cryptocurrency system, but the limit on its transaction processing capacity has resulted in increased transaction fee and delayed transaction confirmation. As such, it is pertinent to understand and probably predict how transactions are handled by Bitcoin such that a user may adapt the transaction requests and a miner may adjust the block generation strategy and/or the mining pool to join. To this aim, the present paper introduces results from an analysis of transaction handling in Bitcoin.**

Specifically, the analysis consists of two parts. The first part is an exploratory data analysis revealing key characteristics in Bitcoin transaction handling. The second part is a predictability analysis intended to provide insights on transaction handling such as (i) transaction confirmation time, (ii) block attributes, and (iii) who has created the block. The result shows that some models do reasonably well for (ii), but surprisingly not for (i) or (iii).

***Index Terms*—Bitcoin, Transaction handling, Linear and non-linear prediction models, Classification, Machine Learning, Artificial Intelligence**

I. Introduction

Blockchain has been considered as an essential technique to resolve privacy and security issues of Big Data, such as privacy protection in IoT [40] and Big Data analytics [17, 20]. To this aim, understanding transaction handling in Bitcoin, e.g., explanatory data analysis or predictability analysis, such as transaction confirmation time, block attributes, is necessary. This naturally demands a thorough study of the transaction characteristics of the blockchain [28], which gives helpful insight into designing and developing new blockchains for smart data or managing big data.

Bitcoin is the first and the largest decentralized electronic cryptocurrency system that uses blockchain technology [27]. It adapts a cryptographic proof of work (PoW) mechanism that allows anonymous peers to create and validate transactions through the underlying peer-to-peer (P2P) network. The peers that maintain and update the chain of blocks are called miners [38, 39]. In addition to transaction generation by user nodes, transaction handling in Bitcoin is done by the full nodes, among which, the miners play a central role: They find the mathematical puzzle to generate a valid block confirming the related transactions.

Due to the design and structure of proof of work (PoW) in Bitcoin, the difficulty of finding the mathematical puzzle increases exponentially, every 2016 blocks. As a consequence, independent miners struggle to find the puzzle. This has forced miners to collaborate to form a team to find the puzzle through a combined computational effort, a *mining pool* [22], and earn a reward, depending on their overall mining power share and the reward mechanism and policy of the mining pool [29] [30]. The mining pools' behavior significantly affects the Bitcoin end users since the mining pools process most of the users' transactions: The throughput of Bitcoin depends partially on those major miners [39]. Additionally, as the number of users increases, the system's internal traffic of transaction handling escalates faster than expected, and at the same time, the throughput requirement increases proportionally with the number of users.

This paper investigates how transactions are handled by the Bitcoin system. The aim is to, through analyzing transaction handling, provide valuable insights to both users and miners:

- A user may expect when his/her transaction will be confirmed and hence choose an appropriate time to request a transaction to reduce the waiting time.
- A miner may define block generation strategies that utilize the current state of the system.
- A miner may also explore which mining pools are more recognizable in the block generation and use this knowledge to join or dis-join a mining pool.

Specifically, through an exploratory data analysis, we reveal key transaction handling characteristics and provide answers to several fundamental transaction handling questions, such as, what is the current throughput, how frequently blocks are generated, how long it takes for a transaction to be approved, and who has created a block. Besides, through a predictability analysis on throughput related features and classification of mining pools, we provide additional insights on these fundamental questions.

The investigation is based on a dataset collected at a Bitcoin full node which contains transaction handling information over a period of 543 days from 7th March, 2019 to 31st August 2020. As a highlight, the dataset includes locally available information that cannot be found on the public ledger blockchain. The results indicate that with a proper prediction model taking into account both internal and external factors, the prediction performance can be appealing for block size and number of transactions in a block, as well as for block generation intensity. However, in terms of predicting when a next block will be generated and a transaction be approved, the effort does not lead to conclusive observation. In addition, also surprisingly, in predicting / classifying the mining pool, clear distinguishing is only found for one specific mining

pool, the F2Pool. Discussion is provided for these findings, including the surprising ones, with the help of findings from the exploratory analysis.

The rest of the paper is organized as follows. Section II illustrates the workflow of transaction handling in Bitcoin, and introduces the dataset used in the analysis. Then, Section III introduces our analysis approach, highlighting the adopted statistical and artificial intelligence techniques. Following that, an exploratory analysis on the dataset is conducted and results are reported and discussed in Section IV. Next, Section V reports results and findings from the predictability study. The current state of the art is covered in Section VI. Finally, Section VII concludes the paper.

II. Bitcoin Transaction Handling: Workflow and Dataset

A. Workflow

Bitcoin is a distributed ledger platform that enables information about transactions to be distributed than centralized, where the ledger is the Bitcoin blockchain that records the transactions. In Bitcoin, all full nodes, also called miners, take part in creating and validating/invalidating transaction blocks and propagating such information, independently [27]. Specifically, the users generate transactions for being processed, and the distributed ledger components, i.e. the full nodes or miners, work together to generate and validate transaction blocks and add them to the blockchain.

Fig.1 illustrates the workflow of transaction handling in Bitcoin, which includes transaction arrival, block formation, propagation and validation. Briefly, after transactions are generated by the users, they are sent to all full nodes for validation. At a full node, upon the arrival of a transaction, the node stores the transaction in its mining pool, called mempool in Bitcoin, waiting for confirmation.

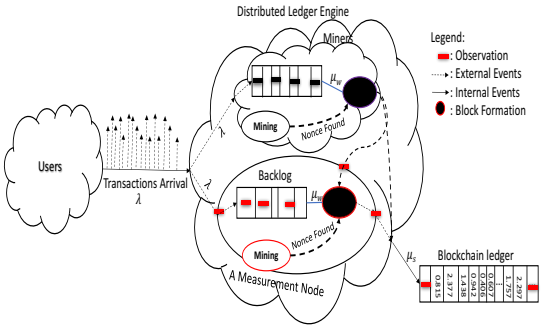


Fig. 1. An illustration of the work flow of Bitcoin

In addition, a full node may choose unconfirmed transactions in the backlog to pack into a new transaction block, and perform mining to find the mathematical puzzle given by the Bitcoin to gain the right to add the block to the ledger. If the puzzle finding is successful, this newly generated block is added to the blockchain, and this information is sent to all the nodes.

At each node, the validity of the newly generated block is checked. If the validity is confirmed with consensus, the updated blockchain is accepted and the transactions in the new block are validated. Such validated transactions are removed from the mempool at each full node that then repeats the above process. Note that, while the above description is brief, the essence of the workflow is kept. For more details about how Bitcoin works, the original introduction [27] is the best source.

B. Dataset

To analyze transaction handling in Bitcoin, we implemented a server installation of a full Bitcoin node to collect related information. The information has two parts. One part records information from the ledger that is globally available, called the *global information part*. Another part records locally available information about the backlog status of the mempool. This part is called the *local information part*.

More specifically, the global information part includes, for each block i on the blockchain, the number of transactions (n_i) in the block, its miner (m_i), the size of the block in bytes (s_i), the timestamp or generation time of the block (T_i), and the average per-transaction fee of the block (f_i). The local information part records the mempool' status (ms_i) in terms of size and fee of backlogged transactions in mempool when each block i is received at our full node.

In total, the dataset consists of information related to 80,408 Bitcoin blocks with more than two hundred million (203432240) transactions for a period of 543 days from 7th March 2019 to 31st August 2020.

III. The Analysis Approach

The dataset is essentially a composition of time series. We hence employ time series analysis on the dataset to provide insights and/or gain findings about transaction handling in Bitcoin. In the rest, the following time series are specifically used: $y = [y_1, y_2, \ldots, y_M]$, $x = [x_1, x_2, \ldots, x_M]$, $c = [c_1, c_2, \ldots, c_M]$, and $D = \{\{y_1, x_1, c_1\}, \{y_2, x_2, c_2\}, \ldots, \{y_M, x_M, c_M\}\}$, defined with:

$$\begin{aligned} y_i &= \{s_i, n_i\} \\ x_i &= \{Td_i, f_i, ms_i\} \\ c_i &= \{m_i\} \end{aligned} \tag{1}$$

where $Td_i \equiv T_i - T_{i-1}$ denotes the inter-block time, s_i, n_i, f_i, ms_i and m_i are defined in the previous section, and $M = 80408$ representing the total number of blocks in the dataset.

Our analysis consists of two parts. In the first part, i.e., Section IV, the focus is on revealing fundamental characteristics and/or basic statistical properties of transaction handling related time series, using exploratory data analysis techniques such as histogram, scatter plot and curve fitting.

In the second part of the analysis, i.e. Section V, the focus is on investigating if / how Bitcoin transaction handling may be predicted. To this aim, both classical and modern time series

forecasting approaches are considered for prediction of various transaction related attributes. In addition, a decision tree based classification approach is adopted for miner inference. The following subsections give an introduction of these approaches.

A. Autoregressive models for forecasting

For time series forecasting, a large number of approaches are available, including both classical ones and modern artificial intelligence (AI) based approaches [10].

For the former, we tested various autoregressive (AR) models. Due to their generally better performance, this paper focuses on ARIMA (AutoRegressive Integrated Moving Average) and ARIMAX (Autoregressive Integrated Moving Average with Exogenous input). Equations (2) and (3) define these models respectively, where B is the backshift operator and ∇ the difference operator.

$$\begin{aligned} y_i^+ = \phi_1 y_{t-1} + \cdots + \phi_p y_{i-p} + \theta_1 \varepsilon_{i-1} + \cdots + \\ \theta_q \varepsilon_{i-q} + \varepsilon_i, \\ \Phi(B)\nabla^d y_i^+ = \Theta(B)\varepsilon_i, \end{aligned} \tag{2}$$

$$\begin{aligned} (y_i^+|T_i = t) = \phi_1\{x_{i-1}, y_{i-1}\} + \cdots + \phi_p\{x_{i-p}, y_{i-p}\} \\ +\theta_1 \varepsilon_{i-1,t_{i-1}} + \cdots + \theta_q \varepsilon_{i-q,t_{i-q}} + \varepsilon_{i,t_i}, \\ \Phi(B)\nabla^d (y_i^+|T_i = t) = \beta x_i + \Theta(B)\varepsilon_{i,t_i}, \end{aligned} \tag{3}$$

where $(y_i^+|T_i = t)$ or y_i^+) is the predicted block, $E(\varepsilon_{i,t_i}) = 0$, Var($\varepsilon_{i,t_i}$) = σ^2, ∇^d=(1-B)^d is difference factor, $\nabla^d(y_i^+|T_i = t)$ is the sequence of y_i by d times differed, $\Phi(B)$= $1-\phi_1 B, \ldots, \phi_p B^p$ is an auto regressive coefficient polynomial, and $\Theta(B)$=$1-\theta_1 B, \ldots, \theta_q B^q$ is a moving smoothing coefficient polynomial of the smooth invertible autoregressive moving average model ARMA (p, q).

To assess the forecasting performance, we use mean average error (MAE) and root mean square error (RMSE), which are respectively defined as: with $e_i = y_i - y_i^+$,

$$\begin{aligned} MAE &= \frac{\sum_{i=1}^{N} |e_i|}{N} \\ RMSE &= \sqrt{\frac{\sum_{i=1}^{N} e_i^2}{N}} \end{aligned} \tag{4}$$

where N denotes the number of predicted data points.

B. AI-based forecasting models

For AI-based models, NAR (nonlinear autoregressive neural network) and NARX (nonlinear Autoregressive Network with Exogenous Inputs) are chosen because they have a feedback connection that encloses several layers of the network, which uses memory to remember the time series's past values to get better performance [18] [36]. Additionally, the models have nonlinear filtering that helps to capture the dynamic fluctuations of the input values.

Equations (5) and (6) describe NAR and NARX network's function to predict a particular value of data series y_i^+ using p previous values of y and x.

$$(y_i^+) = f_{\mathrm{NAR}}(y_{i-1}, y_{i-2}, \ldots, y_{i-p}) \tag{5}$$

$$\begin{aligned} (y_i^+|T_i = t) = f_{\mathrm{NARX}}(\{x_{i-1}, y_{i-1}\}, \{x_{i-2}, y_{i-2}\}, \\ \ldots, \{x_{i-p}, y_{i-p}\}) \end{aligned} \tag{6}$$

The functions f_{NARX} and f_{NAR} in (5) and (6) are unknown, and the neural network training approximates the function by optimizing the network weights and neuron bias. The NAR and NARX model uses Levenberg-Marquardt, Bayesian regularization, and scaled conjugate gradient training algorithms to train the model [2]. Specifically, Bayesian regularization (BR) is used to conduct the analysis. BR minimizes a combination of squared errors and weights; then determines the correct combination to produce a network that generalizes well. It uses network training function Levenberg-Marquardt to optimize network weights and neuron bias. The Levenberg–Marquardt is a popular numerical solution to find the smallest nonlinear function over parameter space.

The following explains the input and output of the neural network model we use.

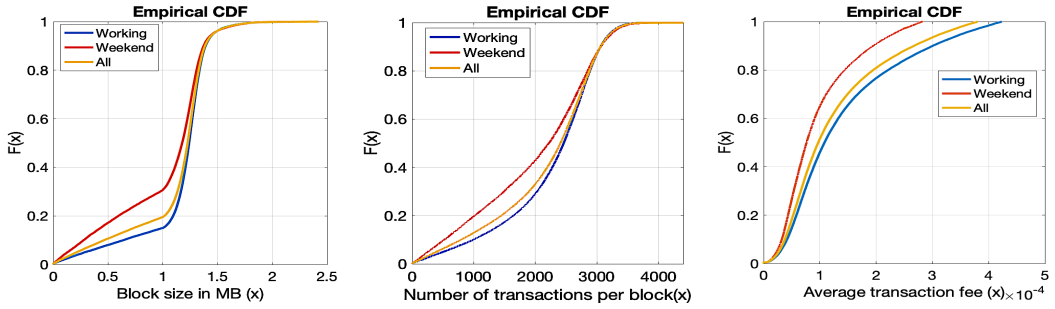
- Input: Block values in the form of vector length, which indicate the number of previous values of the block time series. The models without external input take a vector of the input y_i = $\{n_i,\ s_i\}$ while predicting the next blocks content either n_i or s_i. Similarly, the models with external input additionally take $\{x_i\}$ as an input when the model is used to predict the subsequent blocks.
- Hidden layer: For NAR and NARX, the number of hidden neurons is determined by performing a pre-analysis using the collected dataset. Based on this analysis, the models satisfy the Mean Square Error (MSE) value when the neurons are equal to ten.
- The input delay p and q are approximated by using an autocorrelation (p) and partial-autocorrelation (q) plot.
- Output: The predicted blocks $(y_i^+|T_i = t)$ or y_i^+) contain the predicted values of the blocks $\{n_i,\ s_i\}$ of the weekend, working, and the combinations.

C. Decision tree based classification

Starting in 2010, there are more than 23 mining pools worldwide, as reported in Fig. 3. It has been illustrated that mining pools compete to find the mathematical puzzle and the mining behavior is a game [16][35].

In this paper, we investigate if the mining pools are detectable using a machine learning, decision tree based approach [1][15][41]. It has a tree structure: Each branch represents the outcome of the test, and each leaf node represents a class label. In some cases, it is essential to combine several decision trees to produce a better classification performance. Such a combination produces an ensemble of different methods. In the present work, we considered two methods: booted and RSUbooted [26].

The accuracy, area under curve (AUC), sensitivity, and miss rate are used to test the classification performance, in addition to false negative rate (FN), true positive rate (TP), and receiver operating characteristic (ROC) curve of TP versus TN, as commonly used for machine learning based classification [37].



(a) Empirical CDF of s_i (b) Empirical CDF of n_i (c) Empirical CDF of f_i

Fig. 2. CDFs of basic block attributes

IV. Results: Exploratory Analysis

This section reports results and observations from an exploratory analysis of the collected data.

A. Basic block attributes

The size s_i, the number of transactions n_i and the fee f_i are fundamental attributes of a block. Since the transaction activities are time-varying process [12], they may have distributions that vary from time to time, e.g., the weekend and working day demands have a different distribution.

Fig. 2(a) reports in most cases (80%), s_i in working days has less than 1.4 MB, whereas it has a 1.2 MB size during weekends. In both cases, the s_i can grow more than 1.5 MB in 1% of the cases. Relatively, 30% of the blocks have a size less than the default legacy size s_i of 1 MB on weekend days; nevertheless, in working days, less than 20% of the blocks have a size less than 1 MB.

Similarly, Fig. 2(b) illustrates that the n_i also varies as s_i: 50% of the blocks have less than 2200 transactions per block in weekend days, while 2500 transactions per block in working days. In working days, only 20% of the blocks have n_i less than 2100 transactions wherein the weekend, 40% of the generated blocks have n_i less than 2200.

In addition, the miner's economic incentives affect which transactions to include in a block and this financial interest may also show some differences over time. Fig. 2(c) reports, 50% of the blocks, during weekdays, the f_i is smaller than $1.3{*}10^{-4}$ while during working days, the f_i is less than $1.43{*}10^{-4}$. In both cases, 80% of the f_i is smaller than 0.00033 BTC, and with less than one percent, the f_i can grow more significantly than 0.0004 BTC.

B. Miners

Fig. 3 reports miners' contribution in terms of the number of valid blocks in the main chain. As we can observe from the figure, unknown(?), F2Pool, BTC.com, Poolin, and AntPool contribute a higher number of blocks. Combined, these five major mining pools generate around 50% of the valid blocks.

Driven by the financial interest, a mining pool might use a strategy to increase the financial gain [9]. To explore, we

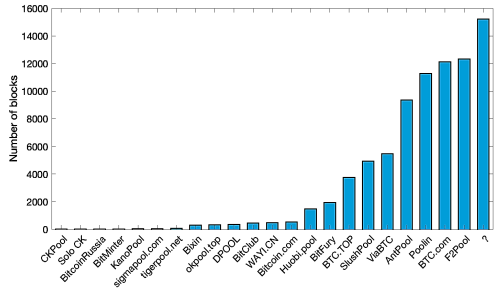


Fig. 3. Miners

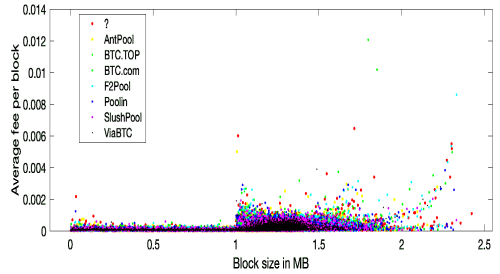


Fig. 4. f_i vs s_i

analyze the blocks generated by the major mining pools. Fig. 4 and 5 report that when the size s_i is greater than 1.5 MB, it is visible from the figures that some of the major mining pools become more recognizable. However, when s_i is less than 1 MB, it is challenging to see any difference between the pools. Similarly, When the s_i is between 1- 1.5 MB, we can see a high concentration of the mining pools. The figures also report that as the s_i increases, the n_i and f_i also rise together.

To further investigate the number of block contributions in the working and weekend days, we focus on the five major miners. Fig. 6 illustrates that these miners contribute similar number of blocks in the working days, except for

TABLE I
MAJOR MINING POOLS BLOCK RELATED ATTRIBUTES PROPERTIES

Mining pool	$\mu(s_i, n_i, f_i)$	$\sigma(s_i, n_i, f_i)$	$\min(s_i, n_i, f_i)$	$\max(s_i, n_i, f_i)$
?	$(1.1252,\ 2.14 * 10^3,\ 1.83 * 10^{-4})$	$(0.3657,\ 844.2627,\ 2.18 * 10^{-4})$	$(2 * 10^{-4},\ 1,\ 0.00)$	$(2.4229,\ 4402,\ 0.0065)$
AntPool	$(1.1141,\ 2.18 * 10^3,\ 1.8 * 10^{-4})$	$(0.3622,\ 844.2076,\ 1.9 * 10^{-4})$	$(3.34 * 10^{-4},\ 1,\ 0.00)$	$(2.2151,\ 4063,\ 0.0050)$
BTC.com	$(1.0960,\ 2.15 * 10^3,\ 1.86 * 10^{-4})$	$(0.3782,\ 868.4394,\ 2.487 * 10^{-4})$	$(2.38 * 10^{-4},\ 1,\ 0.00)$	$(2.3056,\ 4243,\ 0.0121)$
F2Pool	$(1.1099,\ 2.14 * 10^3,\ 1.76 * 10^{-4})$	$(0.3680,\ 845.6503,\ 2.16 * 10^{-4})$	$(2.66 * 10^{-4},\ 1,\ 0.00)$	$(2.3316,\ 4377,\ 0.0086)$
Poolin	$(1.1091,\ 2.17 * 10^3,\ 1.67 * 10^{-4})$	$(0.3635,\ 842.1800,\ 1.87 * 10^{-4})$	$(2.17 * 10^{-4},\ 1,\ 0.00)$	$(2.3165,\ 3988,\ 0.0038)$

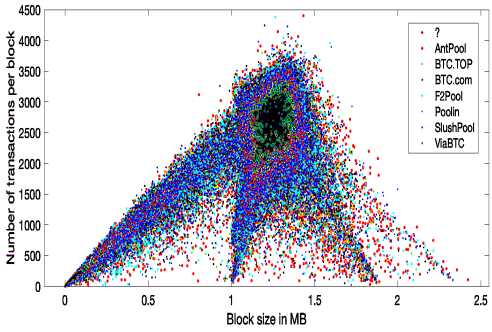


Fig. 5. n_i vs s_i

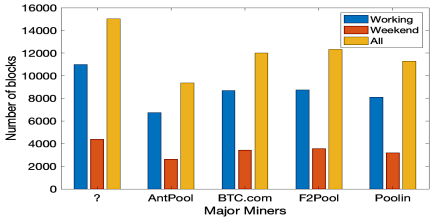
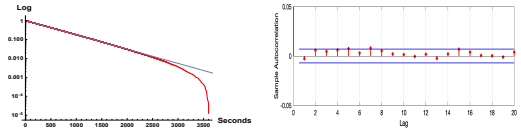


Fig. 6. Miners contribution

the unknown(?) pool. The same observation is also found for the weekend days. The unknown(?) pool generates a higher number of blocks in all cases.

To gain a deeper insight into the block contents than the number of blocks, Table I is presented, where the mean μ, standard deviation σ, minimum and maximum values of the basic block attributes (s_i, n_i, f_i) are shown. Note that these major mining pools become operational starting 2016, except for Pooling in 2018 [39]. Even though there is a gap in years between Poolin and the rest, Table I shows that Poolin, F2Pool, BTC.com generate blocks with similar average size, standard deviation and max values. However, the unknown (?) and AntPool generates block with size greater than the three. The unknown (?) has a block size mean close to 1.214 MB, and the maximum block size is also found in this mining pool. Additionally, the public mining pool, Poolin, comparing the maximum values of f_i and n_i, has the smallest than the other four mining pools.

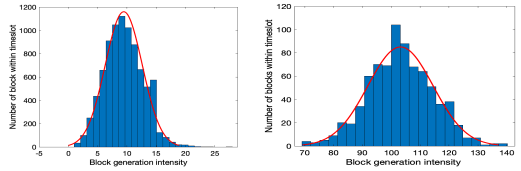


(a) Inter-block generation (b) Autocorrelation plot

Fig. 7. Fitting of inter-block generation time to n.e.d

C. Block generation

1) Distribution of inter-block generation time: Based on the Bitcoin design [27], it has been expected that the inter-block generation time follows an exponential distribution, and the validity has also been checked [12]. Along the same line, Fig. 7(a) reports the fitting of the inter-block generation time to an exponential distribution. Additionally, to check the independence of block generation time, its autocorrelation plot is illustrated in Fig. 7(b). As can be seen from Fig. 7(a) and Fig. 7(b), **the inter-block generation time fits well with an exponential distribution** with increasing mismatch at the tail, partly due to the limited number of blocks in the dataset, and the autocorrelation is close to zero under all the lags in the figure, with the most significant difference only around 1%, indicating that block generation is little correlated.



(a) 100 minutes; $\lambda = 9.44707$ (b) 1000 minutes; $\lambda = 103.184$

Fig. 8. Block generation histogram fitting to a Poisson distribution with intensity λ under different time slot length

2) Fitting to a Poisson process: Since the block generation process has exponentially distributed inter-generation times, we investigate if it can also be further treated as a Poisson process. For this, we make histograms of the number of blocks generated in different length of time and fit them with Poisson distributions. If the process is Poisson, these Poisson distributions must have the same intensity after being scaled. For this investigation, Fig. 8(a) and Fig. 8(b) are presented, where the best fitting intensity of Poisson distribution is shown under two time lengths, 100 and 1000 minutes. Clearly, the obtained two intensities differ noticeably, after taking into consideration

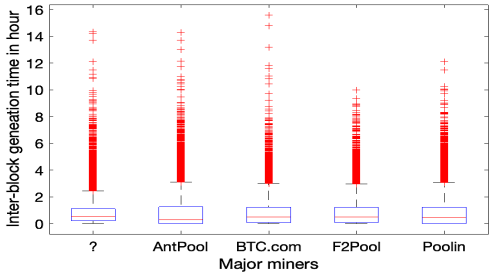


Fig. 9. Major mining pools' inter-block generation time

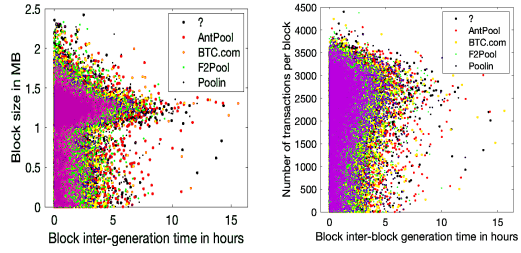
that there is 10x scaling difference. This observation, which is surprising, implies that **block generations can at most be approximated but cannot be treated as a Poisson process.**

3) Relation with miners: To have a closer look on block generation, we made further investigation over the five major mining pools. Fig. 9 reports the inter-block generation time of the major mining pools. As the figure shows, **while the average inter-block generation time is almost the same among the major mining pools, there is visible difference for the median**: While for Unknown(?) and F2Pool, the median time is close to 52 minutes, for BTC.com and Poolin, it is near 45 minutes and for AntPool, it is close to half-hour. The minimum inter-block generation time is the same for all major mining pools, close to zero. However, for the maximum inter-block generation time, while AntPool and Unknown(?) need 14 hours and 30 minutes, BTC.com demands 16 hours. In addition, the public mining pool, Poolin, requires 12 hours, and unlike or shorter than the others, F2Pool needs only 10 hours. As a highlight from Fig. 9, F2Pool stands clearly out of the others with shortest tail.

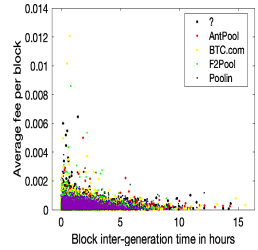
4) Relation with basic block attributes: We further explored the relationship between block generation and the three basic block attributes, shown by Fig. 10(a), 10(b), and 10(c). Specifically, Fig. 10(a) illustrates that when the block size s_i is greater than 1.5 MB, the inter-block generation time seen by the blocks is less than two hours. However, when the block size is concentrated between 1-1.5 MB, Unknown(?), AntPool, and BTC.com block can have the inter-block generation time greater than 13 hours. On the other hand, the blocks from Poolin and F2Pool seem to be generated with shorter interval than the rest three, which is also indicated by Fig. 9.

In addition, Fig. 10(b) demonstrates that the number of transactions n_i in a block of Poolin is on average smaller than the other mining pools. Most of the n_i from F2Pool seem to have a shorter inter-block generation time. However, it is hard to say for the Unknown(?) and AntPool, because the plot shows most of the block with n_i seems to have a larger inter-block generations time. These effects may arise from the state of the mempool, when the mempool contains more transactions then the miners can pick as much number of transaction to include in block.

Furthermore, it is natural the miners prioritize the finical incentives, which encourages the miners to pick up transactions with a higher fee. Fig. 10(c) illustrates this fact. Specifically, **when the fee f_i is higher, the inter-block generation time of the block is lower, maybe even shorter than an hour**. The figure also shows that the blocks with the smaller average fee from Unknown(?), AntPool, and BTC.com may experience inter-block generation time greater than 14 hours. On the other hand, the blocks from Poolin seem to have a less average fee and seeingly smaller inter-block generation time.

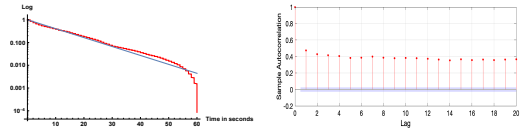


(a) Inter-block generations vs s_i (b) Inter-block generations vs n_i



(c) Inter-block generations vs f_i

Fig. 10. Inter-block generations v.s. block size, transaction number and fee



(a) Transaction inter-arrival time (b) Autocorrelation plot

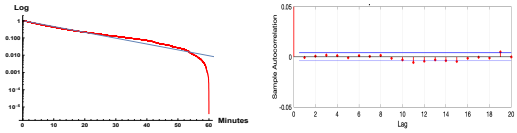
Fig. 11. Transaction inter-arrival time fitting n.e.d

D. Transaction arrival and confirmation time

Users generate transactions for validation. New arrivals stay at the backlog (memory pool) until the nonce finding is successful and they are picked up by the miner.

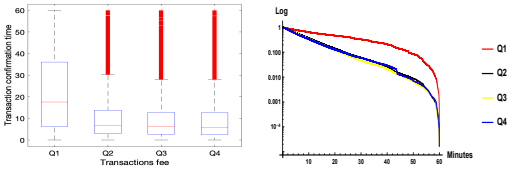
1) Transaction inter-arrival time: Fig. 11(a) shows that the fitting of transaction inter-arrival times to a negative exponential distribution is only reasonable well with visible deviation. Additionally, Fig 11(b) reports the inter-arrival between the transactions is correlated. These reflect that *there exists some level of dependence between transaction arrivals*.

2) Transaction confirmation time: Fig. 12(a) reports the transaction confirmation time fitting to a negative exponential



(a) Transaction confirmation time (b) Autocorrelation plot

Fig. 12. Transaction confirmation time fitting n.e.d



(a) Transaction fee effect (b) log plot

Fig. 13. Transaction fee effect on transaction confirmation time

distribution, with a sharp drop at the tail. Additionally, Fig. 12(b) illustrates that the transaction confirmation time is uncorrelated, reflecting that *the transaction confirmation time is independent*.

Since *a miner tends to choose transactions with a higher fee*, to demonstrate this effect on the confirmation time, Fig .13(a) is presented. Specifically, it demonstrates the relationship of confirmation time and fee for Q1 (25%), Q2 (50%), Q3 (75%), and greater than Q3, i.e., (Q4) for f_b. Their intervals are respectively (0,Q1), (Q1,Q2), (Q2,Q3), and (Q4,∞). As Fig 13(a) shows, low fee transactions exhibit a higher confirmation time. On average, the low fee transactions (Q1) wait 22 minutes for validation. However, for higher fee (Q4) transactions, the average confirmation time is less than half of that of the low fee transactions. For Q2 and Q3, the transactions exhibit close to a ten-minute average confirmation time. Still, transactions from Q2, on average, wait one more minute extra than Q3. Overall, transactions wait on average 13 minutes, and we also observed a few transactions waiting for more than 24 hours at the backlog. At the same, these few transactions also tend to have a fee associated relatively very small.

V. Results: Predictability Analysis

Having explored the various characteristics of transaction handling in the previous section, this section is devoted to investigating if and what such characteristics can be predicted. For this predictability analysis, the prediction approaches introduced in Section III are used. The results are reported and discussed in the rest of this section, where the dataset is divided into three parts, i.e, training, test and validation, and the details of this division is reported in Table II.

A. Basic block attributes

Table III compares the performance of the various models in predicting the target block attributes: size s_i and number n_i, where as a benchmark, the basic autoregressive (AR) model is also included. For these models, the symbol p is order

TABLE II
Division of the Dataset

Dataset	Training	Test	Validation	#No of blocks
Working_day	40095	8591	8591	57277
Weekend_day	16190	3469	3469	23128
All_db	56286	12061	12061	80408

of the autoregressive part, d is the number of nonseasonal differences needed for stationarity, and q is order of the moving average part. In this investigation, the values for $p = 2$ and $q = 2$ are calculated from autocorrelation and partial-autocorrelation plot, and we set $d = 0$. MAE and RMSE are used to compare models' performance. In addition, to give a more direct impression, we illustrate the prediction results by the models for randomly chosen ten consecutive weekend blocks, as an example, in Fig. 14(a) and Fig. 14(b).

Table III, Fig. 14(a) and Fig. 14(b) indicate that, the prediction results by the considered forecasting approaches all follow the actual trend well. However, the models that additionally make use of the locally available information x, which are ARIMAX and NARX, generally produce better results than their counterpart models ARMA and NARX that do not have exogenous input. In addition, the AI-based models perform better than the classical autoregressive models under the same condition. Overall, NARX' performance is best, which is *an encouraging finding for applying AI-based approaches in predicting the basic block attributes' values*.

Remark: The alert reader may have noticed that among the three basic block attributes investigated in the exploratory study, we have left the fee f_i out in the predictability study. This is simply because a large related literature exists, which will be discussed in the related work section, and the results therein show that the price can be excellently predicted.

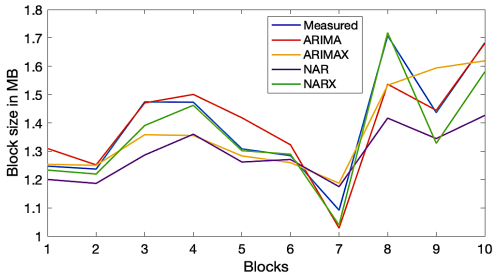
B. Block generation and transaction confirmation time

Encouraged by the prediction results for the basic block attributes, we used the NARX model to test if block generation and transaction confirmation time can also be predicted. For predicting block generation, we used T_i as the input while x= $\{f_i, n_i, s_i, ms_i\}$ as the external input. Fig. 15(a) reports the model's performance. For predicting transaction confirmation, we used transaction confirmation times as the input, while the size of the transactions and the fee associated are used as an external input. Fig. 15(b) exemplifies the model's performance at a number of random points.

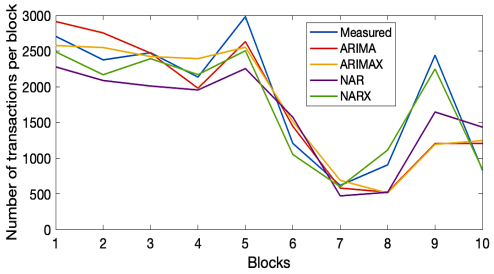
As indicated by Fig. 15(a) and Fig. 15(b), *the predication of block generation and transaction confirmation time does not work*. While this observation seems to be contradictory to the observation in predicting s_i and n_i, a closer look at the characteristics of block generation time and transaction confirmation time enables to explain. Reported in the exploratory analysis in Section IV, both the inter-block generation time and the transaction confirmation time has or can be closely approximated by an exponential distribution. Then, because of the memoryless property of exponential distribution, the likelihood of something happening in the future has little

TABLE III
FORECASTING PERFORMANCE OF BASIC BLOCK ATTRIBUTES

Models	MAE			RMSE		
	Weekend(s_i,n_i)	Working(s_i,n_i)	All(s_i,n_i)	Weekend(s_i,n_i)	Working(s_i,n_i)	All(s_i,n_i)
AR(p)	0.53, 264	0.6, 117.35	0.5, 127.12	0.5, 122.14	0.5, 141.91	0.3, 264
ARIMA(p,d,q)	0.15, 15.373	0.077, 12.840	0.13, 12.969	0.04, 12.461	0.01, 10.833	0.025, 10.942
ARIMAX(p,d,q)	0.12, 13.364	0.07, 12.092	0.06, 11.735	0.02, 11.052	0.006, 10.408	0.006, 10.408
NAR(p)	0.01, 14.770	0.06, 12.969	0.06, 12.840	0.03, 12.214	0.008, 11.275	0.008, 10.942
NARX(p)	0.011, 10.942	0.06, 10.471	0.013, 10.460	0.01, 10.121	0.006, 10.035	0.0003, 10.030



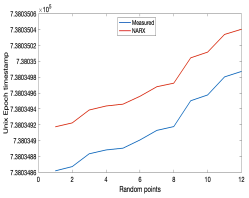
(a) measured vs predicted s_i



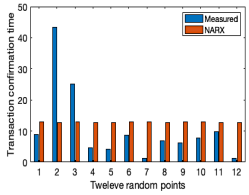
(b) measured vs predicted n_i

Fig. 14. Sample prediction results

relation to whether it has happened in the past. Implied by this and as also confirmed by Fig. 15(a) and Fig. 15(b), **any effort of predicting these two transaction handling aspects may, "surpringly", lead to no solid conclusion**.



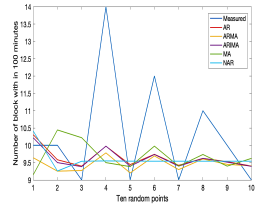
(a) Block generation time



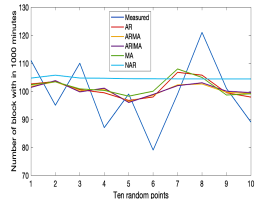
(b) Transaction confirmation time

Fig. 15. Block generation and transaction confirmation time sample prediction

We conduct further investigations on predicting block generation intensity. In this case, for the AI-based models, we only used NAR because we do not have additional input for NARX. To be in line with the counterpart exploratory investigation, we fixed the slot size of 100 and 1000 minutes and predicted the number of blocks within the slot, respectively. Fig. 16(a) and 16(a) report the performance of both the classical autoregressive models and the AI-based NAR model. In general, the AR models follow the trend better than the NAR model. Nevertheless, all models struggle to perform better than the average. This, we believe, attributes largely from that while not exactly, the number of blocks in a time period can is approximately Poisson-distributed, as reported in Section IV.



(a) Block generation intensity with fixed time slot of 100 minutes



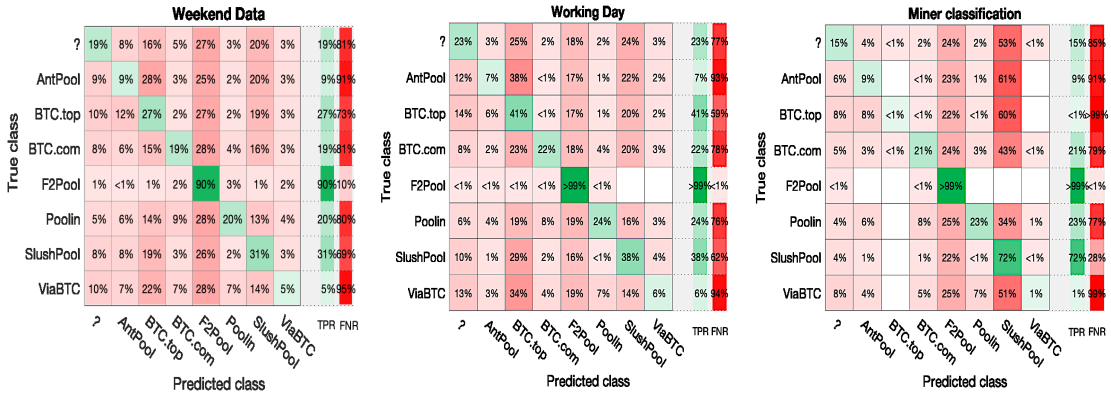
(b) Block generation intensity with fixed time slot of 1000 minutes

Fig. 16. Block generation intensity sample prediction

C. Miner classification

As we saw in the previous sections, the f_i, s_i, Td_i, n_i, and ms_i have a significant effect on the evolution of the Bitcoin ledger. Due to this, we use these feature sets to test if they can help infer a miner's relationship, and if some mining pools use some specified strategies while generating a block. To study these, we take two cases, first working and weekend days, and in the second case, considering all the data together. The feature set, including f_i, s_i, n_i, and Td_i, is used to perform classifications of mining pools (c_i). As a remark, we have also tried other features in the mempool state ms_i but observed that they do not bring significant increase over the accuracy.

1) Case-I (Working and Weekend day): The top-eight mining pools are used to detect the block generation behavior. Fig. 17(a) and Fig. 17(b) report that the major mining pools have a true positive rate (TP) more significant than the rest of the pools. As Fig. 17(a) and Fig. 17(b) report, the better model, the RSUBoosted decision tree with the booted method, shows a promising result classifying the F2Pool in better approximation relative to the other pools. As we can see from Fig. 17(a) and Fig. 17(b), the TP for BTC.com, AntPool, and Poolin is smaller than 25%, but for the SlushPool and



(a) Weekend days (b) Working days (c) All

Fig. 17. Confusion Matrix of major miners (RSUBoosted decision tree)

BTC.TOP, it is more significant than 25%. Especially in the case of the public mining pool, Poolin, the false-negative rate is five times higher than the TP. This indicates *the Poolin has less detectable block generation strategy than the rest. However, for SlushPool, it is has a block generation behaviour more distinguishable than the top five major mining pools.*

2) Case-II (All Data): The previous case showed that F2Pool was approximated very reasonably from the major mining pools. Fig. 17(a) and Fig. 17(b) report a confusion matrix illustrating the F2Pool and SlushPool having a higher positive rate than the rest of the mining pools. Additionally, Fig.17(c) reports that the two major mining pools, SlushPool and F2Pool, the TP are more significant than 70%, which is 40% more accurate than the first case for SlushPool. Similarly, the false-negative rate is less than 20%, especially in F2Pool, which is even less than 3%. To have a better understanding, we performed further investigation on only these two mining pools, F2Pool and SlushPool. The results are reported in Table IV, Fig. 19(a) and 19(b), and Fig. 18. Table IV compares the performance of the two DT methods. Due to better accuracy of the RSUBoosted-tree, it is used in Fig. 19(a) and 19(b), and Fig. 18. Specifically, the true-positive rate (TPR) and the false-negative rate (FNR) are shown in Fig. 18, and Fig. 19(a) and 19(b) further illustrate the model accuracy in terms of AUC and ROC.

TABLE IV
PERFORMANCE OF CLASSIFICATION BETWEEN F2POOL AND SLUSHPOOL

Models	Accuracy	Sensitivity	Miss rate
RSUBoosted-tree	0.90	0.885	0.115
Boosted-tree	0.883	0.881	0.119

3) Discussion: Fig. 17(a), Fig. 17(b), and Fig. 17(c) essentially show that **other than for a few mining pools, particularly F2Pool, mining pools have a minimal positive classification rate, implying they are hard to distinguish**. This is in line with Fig. 9 in the exploratory analysis part,

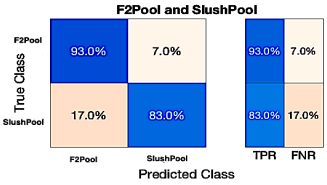
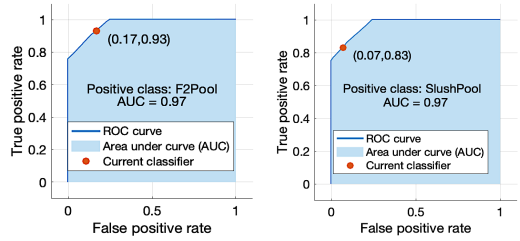


Fig. 18. F2Pool and SlushPool



(a) F2Pool AUC curve (b) ? AUC curve

Fig. 19. AUC curves for F2Pool and SlushPool

which shows that while the block generation distributions of other miners are similar, for F2Pool it is visually distinguishable from the others. We believe this characteristic difference has been explored by the decision tree approach in the classification. In addition, a closer investigation as illustrated by Fig. 17(c) and Fig. 18 implies that the two major private mining pools P2Pool and SlushPool use different strategies that have caused their block generations with special properties making the classification with higher accuracy.

VI. RELATED WORK

1) Statistical analysis of transaction handling characteristics: While a lot of such analysis results are available, e.g.,

various Bitcoin statistics [7], block propagation delay [6], block arrival process [3], transaction rate and confirmation time [13] [33], we focus on fundamental aspects underlying transaction handling and particularly their distributions, different from the literature. Through analyzing these distributions, we have been able to reason some seemly surprising observations in the predictability study. In addition, very few results in the literature take into account information that is only locally available. In this sense, the work [12] is most related. However, except for inter-block generation time fitting, which is similar as we already highlighted, the other results are not found in [12], due to different focuses of [12] and the present work.

2) Forecasting transaction handling characteristics: The focus of the literature has been on bitcoin price. For instance, Huisu Jang and Jaewook Lee [19] developed a neural network-based forecast model on the volatility of a Bitcoin price and extended analysis to identify the best feature set that gives more information about the Bitcoin price process. Similarly, Edwin Sin and Lipo Wang [34] implemented an artificial neural network to predict the next Bitcoin price and the amount of profit that could be gained by making such predictions. Shah et al. [32] considered the Bayesian regression method to predict the price of Bitcoin. Pavel Ciaian et al. [4] estimates Bitcoin price formation based on a linear model by introducing several factors such as market forces, attractiveness for investors, and global macro-financial factors. Greaves et al. [14] analyzed the Bitcoin blockchain data to predict the price of Bitcoin using SVM and ANN, which score 55% accuracy. Similarly, models such as Random Forest, SVM, and Binomial Logistic algorithms are used to predict short-term Bitcoin price and achieve a high accuracy result of 97% in [25]. To the best of our knowledge, no previous work combines the feature sets to predict the transaction handling characteristics focused in this paper.

3) Mining pool classification: There have been some research works that studied block withholding and unfair distribution of reward. For instance, Schrijvers et al. [31] analyzed the incentive compatibility of the Bitcoin reward mechanism. In their model, a miner can decide between honest mining and delaying her found blocks' submission. They proved that the proportional mining reward mechanism is not incentive compatible. Eyal [8] computed the pools' optimal strategy in the block withholding attack and their corresponding revenues. It was demonstrated that the no-pool-attack strategy is not a Nash equilibrium in these games because if none of the pools run the attack, one pool can increase its revenue by launching the attack. Luu et al. [24] experimentally demonstrate that block withholding can increase the attacker's revenue. They do not address the question of mutual attacks. Courtois and Bahack [5] have recently noted that a pool can increase its overall revenue with block withholding if honest pools perform all other mining. We consider the general case where not all mining is performed through public pools and analyze situations where pools can attack one another. M. Salimitari et al. [30] used prospect theory to predict a miner's Profit from joining one of the major mining pools. The hash rate power, total number of the pool members, reward distribution policy of the pool, electricity fee in the new miner's region, pool fee, and the current Bitcoin value are used to predict which pools are profitable specific miners.

Most mining pool studies do either emphasis on **(i)** block withholding [16] [21] or **(ii)** unfair distribution of rewards [11] [22] [23] [35], but none or little has been investigated to detect the major mining pools with hidden block generation strategies. Our work tries to further investigate these block formation strategies, by introducing decision tree to distinguish one of the major mining pools following having a detectable block formation strategy.

VII. Conclusion

An exploratory analysis on fundamental transaction handling characteristics of Bitcoin is conducted, together with a novel analysis on their predictability. The results from the former have been used to help reason the findings from the latter. Specifically, the focused block attributes include the size, the number of transactions and the fee. In addition, block generation and transaction confirmation, two fundamental processes resulted from transaction handling, are investigated. Furthermore, the contribution of miners to these attributes and processes is particularly taken into consideration.

The results show that while it is possible to use measurement-based collected data in predicting the basic attributes of the next block with reasonable accuracy, care is needed in predicting block generation and transaction confirmation. While the latter seems contradicting the expectation from the former, the explanation is supported and implied by results from the exploratory analysis. Additionally, it shows that combining internal and external factors enables better performance in prediction / classification. Furthermore, although it is difficult to distinguish among mining pools through prediction in general, the investigation shows that F2Pool is well distinguished from the others. A closer investigation in the exploratory analysis shows that block generation of F2Pool has a distribution with visible characteristic difference, implying that it has used a different strategy than the other miners. These results shed new light and may also be considered by users and miners when deciding their transaction strategies.

References

[1] A. Abdelhalim and I. Traore. "A New Method for Learning Decision Trees from Rules". In: *2009 International Conference on Machine Learning and Applications*. 2009, pp. 693–698.

[2] Oludare Abiodun et al. "State-of-the-art in artificial neural network applications: A survey". In: *Heliyon* 4 (Nov. 2018), e00938.

[3] R. Bowden et al. "Modeling and analysis of block arrival times in the Bitcoin blockchain". In: *Stochastic Models* 0.0 (2020), pp. 1–36.

[4] Pavel Ciaian, Miroslava Rajcaniova, and d'Artis Kancs. "The economics of BitCoin price formation". In: *Applied Economics* 48.19 (2016), pp. 1799–1815.

[5] Nicolas T Courtois and Lear Bahack. "On subversive miner strategies and block withholding attack in bitcoin digital currency". In: *arXiv preprint arXiv:1402.1718* (2014).

[6] Christian Decker and Roger Wattenhofer. "Information Propagation in the Bitcoin Network". In: *13th IEEE International Conference on Peer-to-Peer Computing*. 2013.

[7] Btc Block Explorere. *Block Explorer*. URL: https : / / btc . com/. (accessed: 01.07.2020).

[8] Ittay Eyal. *The Miner's Dilemma*. 2014. arXiv: 1411.7099 [cs.CR].

[9] Ittay Eyal and Emin Gun Sirer. *Majority is not Enough: Bitcoin Mining is Vulnerable*. 2013. arXiv: 1311.0243 [cs.CR].

[10] Christos Faloutsos et al. "Classical and Contemporary Approaches to Big Time Series Forecasting". In: *Proceedings of the 2019 International Conference on Management of Data*. ACM. 2019, pp. 2042–2047.

[11] Ben Fisch, Rafael Pass, and Abhi Shelat. "Socially optimal mining pools". In: *International Conference on Web and Internet Economics*. Springer. 2017, pp. 205–218.

[12] Befekadu G. Gebraselase, Bjarne E. Helvik, and Yuming Jiang. "Transaction Characteristics of Bitcoin". In: *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 2021, pp. 544–550.

[13] S. Geissler et al. "Discrete-Time Analysis of the Blockchain Distributed Ledger Technology". In: *2019 31st International Teletraffic Congress (ITC 31)*. 2019, pp. 130–137.

[14] Alex Greaves and Benjamin Au. "Using the bitcoin transaction graph to predict the price of bitcoin". In: *No Data* (2015).

[15] Y. Gu and W. Guo. "Apply the Decision Tree Model to Enterprise Informatization Indicators Analysis". In: *2010 International Forum on Information Technology and Applications*. Vol. 2. 2010, pp. 6–9.

[16] Alireza Toroghi Haghighat and Mehdi Shajari. "Block withholding game among bitcoin mining pools". In: *Future Generation Computer Systems* 97 (2019), pp. 482–491.

[17] Hossein Hassani, Xu Huang, and Emmanuel Silva. "Big-Crypto: Big Data, Blockchain and Cryptocurrency". In: 2 (Oct. 2018), pp. 34–.

[18] Mona Ibrahim et al. "Nonlinear autoregressive neural network in an energy management strategy for battery/ultra-capacitor hybrid electrical vehicles". In: *Electric Power Systems Research* 136 (2016), pp. 262–269.

[19] H. Jang and J. Lee. "An Empirical Study on Modeling and Prediction of Bitcoin Prices With Bayesian Neural Networks Based on Blockchain Information". In: *IEEE Access* 6 (2018), pp. 5427–5437.

[20] Konstantinos Lampropoulos, Giorgos Georgakakos, and Sotiris Ioannidis. "Using Blockchains to Enable Big Data Analysis of Private Information". In: *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. 2019, pp. 1–6.

[21] W. Li et al. "Mining Pool Game Model and Nash Equilibrium Analysis for PoW-Based Blockchain Networks". In: *IEEE Access* 8 (2020), pp. 101049–101060.

[22] X. Liu et al. "Evolutionary Game for Mining Pool Selection in Blockchain Networks". In: *IEEE Wireless Communications Letters* 7.5 (2018), pp. 760–763.

[23] Y. Liu et al. "An Intelligent Strategy to Gain Profit for Bitcoin Mining Pools". In: *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*. Vol. 2. 2017, pp. 427–430.

[24] Loi Luu et al. "On power splitting games in distributed computation: The case of bitcoin pooled mining". In: *2015 IEEE 28th Computer Security Foundations Symposium*. IEEE. 2015, pp. 397–411.

[25] Isaac Madan, Shaurya Saluja, and Aojia Zhao. "Automated bitcoin trading via machine learning algorithms". In: *URL: http://cs229. stanford. edu/proj2014/Isaac% 20Madan* 20 (2015).

[26] Mathworks. *ensemble algorithms*. URL: https://www.mathworks.com/help/stats/ensemble-algorithms.html. (accessed: 01.07.2020).

[27] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". In: *Cryptography Mailing list at https://metzdowd.com* (Mar. 2009).

[28] Daniel Perez, Jiahua Xu, and Benjamin Livshits. "Revisiting Transactional Statistics of High-scalability Blockchain". In: *ACM Internet Measurement Conference (IMC)*. 2020.

[29] R. Qin, Y. Yuan, and F. Wang. "Research on the Selection Strategies of Blockchain Mining Pools". In: *IEEE Transactions on Computational Social Systems* 5.3 (2018), pp. 748–757.

[30] M. Salimitari et al. "Profit Maximization for Bitcoin Pool Mining: A Prospect Theoretic Approach". In: *2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC)*. 2017, pp. 267–274.

[31] Okke Schrijvers et al. "Incentive compatibility of bitcoin mining pool reward functions". In: *International Conference on Financial Cryptography and Data Security*. Springer. 2016, pp. 477–498.

[32] Devavrat Shah and Kang Zhang. "Bayesian regression and Bitcoin". In: *2014 52nd annual Allerton conference on communication, control, and computing (Allerton)*. IEEE. 2014, pp. 409–414.

[33] Jun.Kawahara Shoji.Kasahara. "Effect of Bitcoin fee on transaction-confirmation process". In: *Journal of Industrial and Management Optimization* 15.1547 (2019), p. 365.

[34] E. Sin and L. Wang. "Bitcoin price prediction using ensembles of neural networks". In: *2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. 2017, pp. 666–671.

[35] Yonatan Sompolinsky et al. "Bitcoin Mining Pools: A Cooperative Game Theoretic Analysis". In: vol. 2. Jan. 2015.

[36] Tsung-Nan Lin et al. "A delay damage model selection algorithm for NARX neural networks". In: *IEEE Transactions on Signal Processing* 45.11 (1997), pp. 2719–2730.

[37] E VALUATIONS. "A REVIEW ON EVALUATION METRICS FOR DATA CLASSIFICATION EVALUATIONS". In: 2015.

[38] C. Wang, X. Chu, and Y. Qin. "Measurement and Analysis of the Bitcoin Networks: A View from Mining Pools". In: *2020 6th International Conference on Big Data Computing and Communications (BIGCOM)*. 2020, pp. 180–188.

[39] C. Wang, X. Chu, and Y. Qin. "Measurement and Analysis of the Bitcoin Networks: A View from Mining Pools". In: *2020 6th International Conference on Big Data Computing and Communications (BIGCOM)*. 2020, pp. 180–188.

[40] Steven A. Wright. "Privacy in IoT Blockchains: with Big Data comes Big Responsibility". In: *2019 IEEE International Conference on Big Data (Big Data)*. 2019, pp. 5282–5291.

[41] Yurong Zhong. "The analysis of cases based on decision tree". In: *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. 2016, pp. 142–147.

Paper C:
B. G. Gebraselase, B. E. Helvik and Y. Jiang, "Effect of Miner Incentive on the Confirmation Time of Bitcoin Transactions," 2021 IEEE International Conference on Blockchain (Blockchain), 2021, pp. 521-529, doi: 10.1109/Blockchain53845.2021.00079.

Effect of Miner Incentive on the Confirmation Time of Bitcoin Transactions

Befekadu G. Gebraselase, Bjarne E. Helvik, Yuming Jiang
Department of Information Security and Communication Technology
NTNU, Norwegian University of Science and Technology, Trondheim, Norway
{befekadu.gebraselase, bjarne, yuming.jiang}@ntnu.no

***Abstract*—Blockchain is a technology that provides a distributed ledger that stores previous records while maintaining consistency and security. Bitcoin is the first and largest decentralized electronic cryptographic system that uses blockchain technology. It faces a challenge in making all the nodes synchronize and have the same overall view with the cost of scalability and performance. In addition, with miners' financial interest playing a significant role in choosing transactions from the backlog, small fee or small fee per byte value transactions will exhibit more delays. To study the issues related to the system's performance, we developed an $M(t)/M^N/1$ model. The backlog's arrival follows an inhomogeneous Poison process to the system that has infinite buffer capacity, and the service time is distributed exponentially, which removes N transactions at time. Besides validating the model with measurement data, we have used the model to study the reward distribution when miners take transaction selection strategies like fee per byte, fee-based, and FIFO. The analysis shows that smaller fee transactions exhibit higher waiting times, even with increasing the block size. Moreover, the miner transaction selection strategy impacts the final gain.**

***Index Terms*—Bitcoin, Transaction waiting time, Miner strategy**

I. Introduction

Cryptocurrency, which is a digital equivalence of fiat currency, is becoming popular. As of April 2020, there were approximately 5,392 cryptocurrencies being traded with a total market capitalisation of 201 billion dollars[1]. The volume of transactions and circulation of these cryptocurrencies are uneven [21]. As of April 2020, Bitcoin (BTC), ether (ETH) and Ripple (XRP) were the top three cryptocurrencies with market capitalization of 128, 19.4, and 8.22 billion dollars respectively. Bitcoin is an autonomous decentralized virtual currency that removes the intermediary between participating parties while the cryptographic encryption and peer-to-peer formations provide the security. This property has attracted much attention from the research and industry world to develop and integrate blockchain in the supply-demand chain. The amount of Bitcoin usage and integration exhibits rapid increases in recent years. For instance, the number of transactions per day in 2020 is twice higher as from 2016 to 2018 [42].

[1]https://finance.yahoo.com/news/top-10-cryptocurrencies-market-capitalisation-160046487.html

Bitcoin has become popular with an increasing number of transaction requests over time. However, due to the limited capacity by design (average one block per 10 minutes) of Bitcoin, the number of transactions that the system can handle is also limited. This necessitates a strategy for a miner to select transactions in forming blocks. While it is natural for the miners to priority higher fee transactions to gain financially, such a strategy may cause a long delay in transaction confirmation for lower fee transactions. As a consequence, such financial gain oriented strategies may reduce the overall quality of the services provided by the ledger. Furthermore, as the number of users increases unexpectedly while the number of mining nodes and pools rises linearly [38], this makes Bitcoin unsuitable for small fee transactions.

Bitcoin is facing criticism over the scalability and performance [5]. It is imperative to study Bitcoin's transaction confirmation process' characteristics since they are critical indicators of how scalable the ledger is [19]. To this end, some models have been proposed to study the average waiting time seen by transactions while considering the coefficient of arrival variation, batch processing, and block sizes [20][28]. However, based on a recent measurement-based work reported in [41], it is found that the transaction arrivals follow an inhomogeneous Poisson process and the arrival attributes have week correlations. In addition, the fee per byte is the default ordering mechanism in Bitcoin, while not just fee [13]. However, this fact is not addressed by most of the available modeling works, including [31][37]. In this paper, we consider these insights to model and study the transaction waiting time.

This paper aims to investigate how different transaction selection strategies may affect the performance of Bitcoin in terms of transaction waiting time. However, there is a challenge: We cannot widely introduce such a strategy on Bitcoin. (i) For this reason, we develop a queueing model that simulates the behavior of Bitcoin with a focus on transaction waiting time. In the literature, several queueing models have been proposed. Our work proposes a new queueing model based on our previous extensive investigation on transaction handling and characteristics of Bitcoin. Based on this queueing model, a simulator is developed. The model/simulator is validated with measurement data from Bitcoin. (ii) With the simulator, we then study the transaction waiting time under different transaction selection/scheduling strategies, which include (Bit-

coin default) fee per byte and fee-based. Beside this, we also consider the impact of increasing the block size on transaction waiting time. (iii) In addition, to account for that different miners may adopt different strategies, an investigation is also provided to check potential gain or loss to a miner.

The rest of the paper is organized as follows. The current state of the art is covered in Section II. Following that, Section III presents the queueing model description and the simulator workflow, and validates the model results. After that, experimental results are discussed in Section IV. Next, Section V presents results from comparing different strategies. Section VI opens up a discussion on what has been observed in the analysis. Finally, Section VII concludes the paper and outlines future research extensions.

II. Related Work

A. Queueing Models of Transaction Waiting / Confirmation Time

There are several works related to studying the average waiting time of transactions before their confirmations using queueing models. S. Geissler et al. [28] proposed a $GI/GI^N/1$ model where the inter-arrival and batch service times follow an independent general distribution. Based on the model, they were able to show that the average arrival intensity variations and block size play a significant role in the confirmation times. Similarly, Lie et al. [20] illustrated that the block size and average arrival intensity exhibiting a significant factor in the average waiting time by developing a $GI/M^N/1$, where the inter-arrival time follows a general distribution but the batch service time follows an exponential distribution.

Yoshiaki Kawase and Shoji Kasahara [13] developed an $M/G^B/1$ model where a batch service is used to reflect the block size limitations with the arrivals being blocked from entering into a block under the mining phase. The sojourn time of a transaction corresponds to its confirmation time. The same authors [31] showed that because of a high arrival intensity, even high fee transactions are exhibiting a higher average waiting time. Additionally, it was observed how the low fee and block size significantly affect transaction confirmation time. Similarly, [18] developed a batch processing queueing system that uses numerical and trace-driven simulation to validate exponential distribution, and hyper-exponential one can accurately estimate the mean transaction-confirmation time for the legacy 1MB block size limit.

Mišić et al. [34] developed an analytical model to capture the Bitcoin P2P network. They developed a priority-based queuing model $(M/G/1)$ of Bitcoin nodes and a Jackson network model of the whole network. The study illustrated that the block size, node connectivity, and the overlay network significantly affect the probability of fork occurrence. Furthermore, the study demonstrated the data distribution in the P2P network is sub-exponential, and the transaction traffic has less effect on the block propagation traffic mainly because of the priority. Motlagh et al. [35] developed a Continuous Time Markov Chain (CTMC) model to study the churning process of a node with a homogeneous sleep time. The analysis shows that results indicate that sleep times of the order of several hours require synchronization times in the order of a minute.

Most of the research mentioned above works to evaluate the blockchain technology's performance concerning block size, transactions, node connectivity, churn, and block delivery. However, little has been investigated about the impact of the transaction selection strategy in forming blocks, considering the weak dependency between transaction attributes, and the inhomogeneous transaction arrivals.

B. Reward Distribution

Salimitari et al. [15] developed a prospect theoretical model to predict what a miner can mine relative to its hash rate power and electricity costs and how much may be expected to make from each pool. It was also demonstrated that the best pool for a miner to join is not always the same for all. Liu et al. [14] proposed to introduce a forwarding node to reduce time delay for message propagation and increase the probability for a new block to be appended on the longest blockchain. Samiran et al. [11] performed an analysis on how a selfish miner could earn some extra incentive for launching a block withholding attack on a mining pool. This additional incentive comes from some other like-minded mining pool that wants to benefit from this block withholding attack. A. Laszka, B. Johnson, and J. Grossklags [7] developed a game-theoretical model to study the impact of attacks on mining pools in either short or long-term effects. This model is used to consider when the miner has an incentive to attack the pool or has no incentives to conduct the attack. Eyal [6] showed that identical mining pools attack each other. They have demonstrated no Nash equilibrium when there is no attack on the pool; this will increase earned by participating parties. When two pools can attack each other, they face a version of the Prisoner's Dilemma. If one pool chooses to attack, the victim's revenue is reduced, and it can retaliate by attacking and increase its revenue. However, at Nash equilibrium, both attacks earn less than they would have if neither attacked. With multiple pools of equal size, a similar situation arises with asymmetric equilibrium.

Pontiveros et al. [17] showed that the size and fee of the transaction have a higher importance in detecting mining pool strategies. Santos et al. [25] proposed a faster size-density table-based method that performs better in terms of the number of transactions processed and the total capital income. This approach is to remove sorting-based algorithms at block generation events. However, this method has not been compared with transaction selection strategies adopted by either public or private blockchains. Rizun [10] formalized the intuitive idea that the matching of supply with demand should determine equilibrium transaction fees. Fiz [16] modeled the transaction selection problem as a classification problem and proved that the essential features of the transactions when selecting them are their size and fee values.

III. Queuing Model Based Simulator

In this section, a new queueing model for estimating transaction waiting time is proposed, based on which a simulator has been developed. The validity of the proposed model is checked with measurement data.

By *transaction waiting time*, we mean the delay between when the transaction is received by the system and when the transaction is included in a block. Note that, there is additional delay till transaction confirmation, which is the delay for the system to achieve consensus and approve the addition of the block to the chain. Since this additional delay is not affected by the miners' transaction selection strategies, it will not be included in the model or later discussion if not explicitly stated.

A. Model Description

Figure 1. M(t)/M^N/1 model

The users generate transactions for processing, and the blockchain engine provides a secured, autonomous, and privacy-preserving platform. The number of users that integrate the service increases exponentially, leading to the case in which the backlog gets filled with a large number of transactions waiting for the process. Fig. 1 illustrates the Bitcoin workflow. In this case, taking into consideration the behaviour of a typical miner, we use a queueing model to represent the system. The users' newly generated transactions arrive at the system with an intensity of $\lambda(t)$, and the miners generate blocks with an intensity of μ.

a) Arrival process: More specifically, the transaction arrivals follow an inhomogeneous Poisson process with an intensity of $\lambda(t)$ as having been observed in a measurement study [41].

To generate the inhomogeneous arrival intensity from the homogeneous Poisson process, we can use the Lewis and Shedler thinning methodology [2],[3], as illustrated in Algorithm 1, where the λ constrains the next arrival intensity. The $\lambda(t)$ is bounded by $\frac{\lambda(S_w)}{\lambda}$, where the S_w is the next exponential inter-arrival time and λ is the upper bound of $\lambda(t)$. Based on the current state of Bitcoin processing capacity, the value of λ is set to 7.2 [42].

b) Arrival attributes: The new arrival transactions contain important features like fee and size that play a role in ranking order and filling up the block. For instance, Bitcoin orders the new arrivals according to the fee per byte ratio. The weak dependency between the transaction fee and size impacts the number of transactions added to the block. In this work, we also introduce this dependency in the model.

Algorithm 1 Inhomogeneous Poisson Process

procedure INHOMOGENEOUS($\lambda(t), T$)
  Initialisation: $n = m = 0, t_0 = s_0 = 0$
  Condition: $\lambda(t) \leq \lambda, \forall_{t \leq T}$
  while $s_m \leq T$ **do**
    x $\sim$ U(0, 1)
    $y = -\frac{ln(x)}{\lambda}$
    $s_{m+1} = s_m + y$
    $D \sim U(0, 1)$
    if $D \leq \frac{\lambda(s_{m+1})}{\lambda}$ **then**
      $t_{n+1} = S_{m+1}$
      $n = n + 1$
    return $[t_n]$

c) Service process: The transactions are waiting at the backlog to be picked up and included in a block. Block generation is an independent and identically distributed random event requiring the miner to perform some mathematical puzzles, as Bitcoin's case. The block-generation times follow exponential batch processing with a rate of μ. The block holds N number of transactions, in which the size of the block (β) can only have as many numbers of transactions possible and available at the backlog.

d) Block size: A valid block holds N number of transactions, and the maximum size of the block size (β) is fixed. The pushing block size to the maximum limit also brings the propagation delay, which may trigger a fork in the distributed system. However, it is crucial to see how the β affects the transactions' average waiting time. To see this effect, we compare block size from legacy size, which is 1 MB to 8 MB.

e) Transaction selection / scheduling strategies: To explore how much low-fee transactions may suffer from the strategy used by a miner in selecting / scheduling backlogged transactions in forming a block, three strategies are considered. One strategy we are considering is the fee per byte ordering at the backlog, which is the default strategy used by Bitcoin. In addition, the fee-careless first in first out (FIFO) strategy, and the strategy of prioritizing higher fees are also considered.

B. Simulator Workflow

This sub-section covers the workflow of the simulator. It captures the workflow of a full Bitcoin node that participates in the verification and validation of transactions.

There have been some works on developing a simulator to study the evolving technology's performance, blockchain. The currently available simulators focus on realizing node-to-node connectivity, propagation delay, and adding Merkle tree into the simulator, including [22, 24, 27, 32, 33]. Since these simulators have no functionality to include the dependence between transactions fee and size, the change of the scheduling algorithms, and realizing inhomogeneous transactions arrival process, we developed a discrete event simulator/emulator by using Simpy [43].

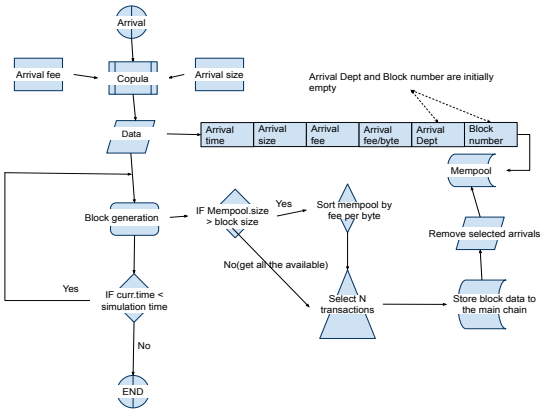


Figure 2. Flowchart showing the workflow of the model-based simulator

We can demonstrate the workflow of the simulator by using an example. Let a vector $[t, s, f, f/s, d, bN]$ represent a new arrival event at time t, in which the arrival has attributes of a fee (f), size (s), fee-per-byte (f/b), waiting time (d), and block number (bN). As it was discussed in previous sections, there is a weak dependence between arrival fee and size, this is realized by using Copulas [8]. Initially, the value of d and bN is zero. Similarly, the other arrivals will be recorded at the mempool and waiting for a pick-up.

When the block generation event happens, there are two ways of selection if there are enough arrivals stored for pick-up. Firstly, we can use the time of arrival of the transactions (t), which gives us FIFO. Secondly, we can consider the situation with the miners' knowledge having high incentives to increase the financial gain, prioritizing high fees (Fee-based). Thirdly, it uses the default method proposed by the Bitcoin community fee per byte ratio, f/s order [4]. Then, the block's size and the associated transaction size determine the number of transactions included in a block. If not, it picks-up the available arrivals and generates the block. Fig. 2 illustrates the concept by considering fee per byte as the scheduling algorithm in the form of a flow chart at a high-level detail. In this work, we consider fee-based, fee per byte, and FIFO. The first two are used to show the impact of the miner incentives and FIFO to demonstrate the difference between financial gain or not.

After the transactions are selected by one of the scheduling strategies mentioned earlier, the block containing the correspondingly selected transactions will be added to the chain. These transactions also get removed from the backlog. The chain grows in each block generation until the simulation window is finished. Like the real Bitcoin node, this simulator keeps track of each transaction's arrival time, fee, size, block number, and waiting time. These collected attributes are used to generate valid results and compare the result with currently available literature.

The transaction arrival and block generation events change the state of the system. The arrival event increases the mempool in the number of arrivals and size-wise; on the other hand, the batch processing removes N elements from the backlog.

C. Model Validation

This section presents results validating the model-based simulator with trace-driven simulation. It has been demonstrated [41] that transaction fees and sizes follow a lognormal distribution with different mean and standard deviation while showing weak correlation, which is considered in the simulation.

To validate the model, we performed a test and the results are reported in Fig. 3, where the model-based simulation results are compared with trace-driven simulation results. As the figure illustrates, the model captures the results in a good fit. The x-axis represents the block size. The y-axis indicates the transaction's average waiting time by considering the model and actual data from the bitcoin node. In this work, the average waiting time is time between a transaction generation and its addition to a valid block. As the default case in Bitcoin, a fee per byte is used to order the arrivals for pick up [4].

Table I further illustrates a comparison between our proposed model with recent related works [13, 31] and measurement results [40]. The row indicates the block size, and the column represents related models from the literature. The arrival transactions intensity is fixed with $\lambda = 3.0$. This table only considers block size from 1MB to 3MB. This is mainly because most paper commonly consider the block size from 1MB to 3MB, e.g., [18, 28]. Our proposed model seems to fit with other related works' results. As a highlight, our model produces better matching result with the measurement [40].

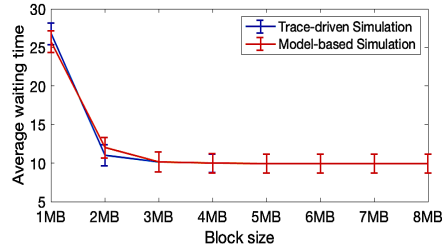


Figure 3. Transactions average waiting time vs block size while fee per byte is the scheduling algorithm ($\lambda(t) \in [3.0, \ldots, 3.3]$)

Table I
MODEL COMPARISON ($\lambda = 3.0$)

Models	1MB	2MB	3MB
$M/G^N/1$ [13, 31]	26	13.66	10.33
Bitcoin [40]	–	13.1	–
$M(t)/M^N/1$	25	13.01	10.14

IV. IMPACT OF TRANSACTION SELECTION STRATEGY

In this section, we investigate the impact of transaction selection / scheduling strategy used by a miner on transaction

waiting time. The investigation is based on the simulator introduced in the previous sections. First, the validity of the law of conservation regarding scheduling algorithms in bringing the same average time is illustrated. Then, our simulation considers two cases that have been mentioned in Section III, **(i)** the default method proposed by Bitcoin, which is the fee per byte, and **(ii)** considering the particular case demonstrating the financial interest of the miner is only the fee.

A. Conservation of Average Waiting Time

Fig. 4 reports the average waiting time transactions seen while using fee per byte and fee-based. The x-axis represents the block size in MB, the y-axis indicates the average waiting time, and the legend classifies the type of strategy used. The plot illustrates that choosing any strategy while the arrival intensity is within the range of 3.0 to 3.3 may not affect the average waiting time. However, this behavior can only apply when the number of arrivals waiting for pick up is smaller than the block can hold. Table II presents the filling rate of the block in terms of the mean and standard deviation. The row represents the block size, and the column reflects the strategy applied. As we can see from the table, the filling rate of the block in all the cases is lower than one, which means most of the time, the block is not pushed to maximum size.

B. Case-I (Fee per byte)

Miners are the backbone of Bitcoin, participating in adding, validating, and forwarding new updates to the neighbors. Mainly, what a miner involves is solving the mathematical puzzle through high computation effort. When the miner finds the nonce, it collects transactions from the backlog, ordering in fee per byte [4]. In such cases, a transaction with a higher fee per byte ratio is picked up earlier than the low fee per byte. It is natural for the miners to choose transactions with a higher fee per byte since it increases the financial gain. However, this may affect the average waiting time for a low fee per byte transaction. It was demonstrated that the transaction fee fluctuates [42], transactions with a smaller fee observe a longer average waiting time [18]. There is a gap in the literature to illustrates how much a minor transaction has to wait.

Fig. 5 illustrates the average waiting time seen by the transactions relative to the block size increase. The x-axis represents the block size ranging from 1MB to 8 MB, and the y-axis shows the average waiting time. It demonstrates the relationship between block size and average waiting time for Q1 (25%), Q2 (50%), Q3(75%), and greater than Q3 (>Q3) for a fee per byte. As the figure shows, transactions with a low fee per byte ratio observe a higher waiting time. This is highly observable within the block size ranging from 1MB - 3MB. However, after 5MB, the effect of the financial incentive becomes smaller. This can also come because the mempool has fewer waiting transactions relative to the smaller block size, which forces the miner to pick up what is in the mempool.

Fig. 6 reports the average waiting time a transaction sees while the block size is pushed to maximum and the arrival intensity are within range of 7.0 to 7.3. As it is shown, when the block size increases, the average waiting time decreases. Similarly, this trend is also visible when the intensity is within $\lambda(t) \in [3.0, \ldots, 3.3]$. The reduction of the average waiting time after 6MB is smaller enough to be considered equal.

Fig. 7 reports the sample result showing the transaction waiting time in terms of low to a high fee per byte. The x-axis represents the average waiting time in minutes. The y-axis is the empirical CDF. The legend in the plot classifies the transactions based on the block size. As we can see from the plot (7(a)) low fee per byte transactions, for 1MB block size, 80% transactions see waiting time less than 70 minutes,

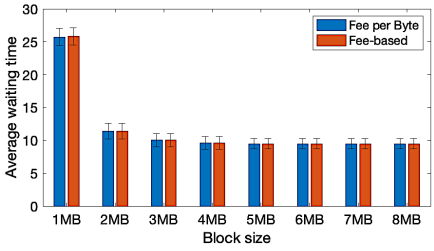


Figure 4. Scheduling algorithms comparison ($\lambda(t) \in [3.0, \ldots, 3.3]$)

Table II
FILLING RATE COMPARISON

Strategies	1MB(μ, σ)	2MB(μ, σ)	6MB(μ, σ)	8MB(μ, σ)
Fee-based	(0.86,0.02)	(0.433,0.01)	(0.144,0.003)	(0.114,0.002)
Fee per byte	(0.85,0.021)	(0.431,0.012)	(0.143,0.003)	(0.111,0.002)

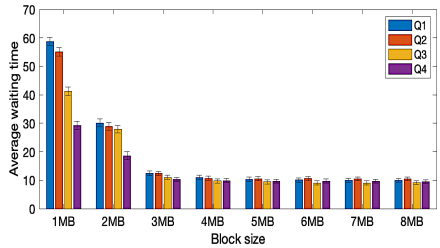


Figure 5. Fee per byte ($\lambda(t) \in [7.0, \ldots, 7.3]$)

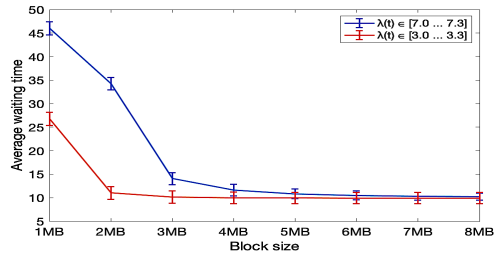
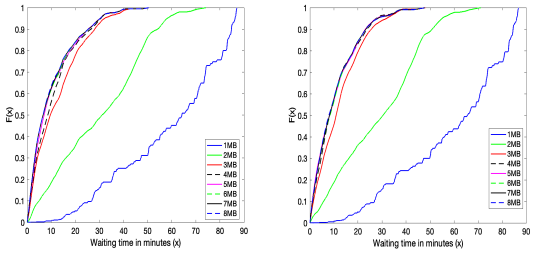
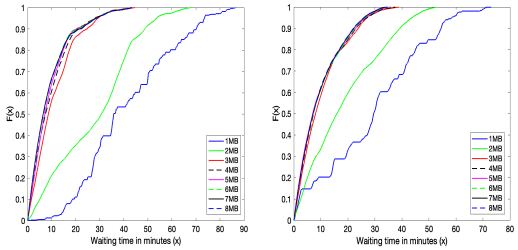


Figure 6. Arrival intensity vs block size



(a) Q1 (b) Q2



(c) Q3 (d) Q4

Figure 7. Transactions average waiting time vs. block size, where $\lambda(t) \in [7.0, \ldots, 7.3]$, fee per byte scheduling

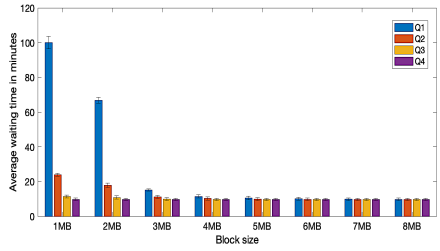


Figure 8. Fee based $\lambda(t) \in [7.0, \ldots, 7.3]$

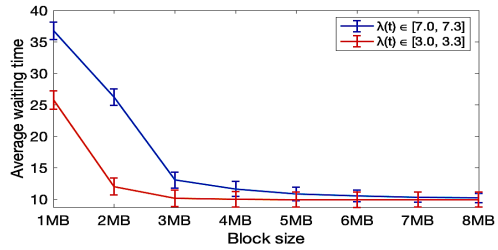


Figure 9. Arrival intensity vs block size

and for 2MB, these transactions observe less than 45 minutes.

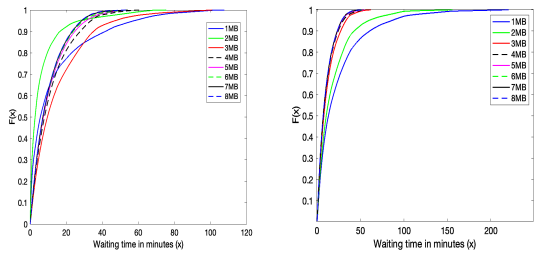
Similarly, for 3MB, smaller transactions 80% of the time see less than 15 minute waiting time. This behavior is repeated for a medium fee per byte (7(b)) size. However, a high fee and very high fee per byte ratio transactions tend to see shorter waiting times. For instance, in the case of very high fee per byte transactions, most transactions (80%) see waiting time shorter than half an hour. Even increasing the block size has a more negligible effect on exhibiting the low-fee transaction suffering from longer waiting time.

C. Case-II (Fee-based)

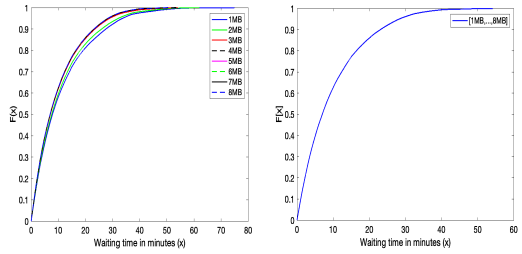
We assume the miner prioritizes the financial motives over the default consideration of fee per byte [29]. A miner can sort the transactions in descending order of fee per byte, from the most profitable one to the least one [23]. By doing such ordering, it is easier to pick up a transaction that brings a higher profit.

Fig. 8 reports that miner financial interest is affecting the waiting time. For the block size from 1MB to 3MB, the impact of miners' incentives to select the top-fee transactions is more visible. Transactions with smaller fees (Q1) wait for 30 minutes more than Q2. However, starting the block size greater than 3MB, the average waiting time between smaller and higher becomes similar.

Fig. 9 reports the average waiting time seen by transactions when fee-based scheduling is used to pick up transactions from mempool. The x-axis represents the block size, the y-axis indicates the average waiting time, and the legend classifies the two arrival intensity considered. As we can see from the plot, when the arrival intensity is $\lambda(t) \in [7.0, \ldots, 7.3]$, the average waiting time is seen by transactions when the block size is 1MB, or 2MB has smaller values than using fee per byte scheduling. Starting 3MB, the average waiting time seen by using fee per byte or fee-based looks similar.



(a) Q1 (b) Q2



(c) Q3 (d) Q4

Figure 10. Transactions average waiting time vs. block size, where $\lambda(t) \in [7.0, \ldots, 7.3]$, fee based scheduling

Fig. 10 shows how only choosing transactions with a higher fee affects the waiting time. Fig. 10(a) and 10(b) plots values for 1MB, and 2MB are scale down by 10 for better visibility. Similar to the case where the fee per byte is used as the scheduling algorithm, higher fee transactions also show similar trends but different waiting times. Since only choosing transactions with a higher fee does not consider the possibility of size limitations, it seems fee per byte is a better option in giving fairer chances for transactions. However, this may change if the backlog is always full and there are more transactions to choose. In that case, selecting transactions based on fees may bring better gain but make low-fee transactions suffer long-time wait.

Unlike the fee per byte case, when transactions have a very high fee, they see the same average waiting time, implying the block size has a more negligible effect on the confirmation time. 90% of the time, transactions see less than 25 minutes waiting time, and there are also less than 5% of the transactions see more than 42 minutes average waiting time.

Remark: The alert reader may have noticed that among the three strategies for transaction selection and block creation investigated in this study, we have left the FIFO out in the discussion above. This is simply because the fee-careless FIFO strategy does not show any significant difference between transactions of different fees or between transactions of different fees-per-byte: All transaction types have the same average waiting time.

V. Reward Comparison

In this section, we compare the strategy in terms of the reward a miner gets by adopting different strategies. We consider two miners (M1, M2) competing to generate a block with equal probability while using different strategies, as illustrated in Fig. 11. These two miners share the same backlog. We consider each block generated by these miners valid and added to the main chain for simplicity. The total reward ($R_T = \sum_{i=1} f_i$) is the sum of all transactions' fee at the backlog. In a fair chance, each miner should get half of the reward ($\frac{1}{2}R_T$). We fix each miner's strategy and then compare each miners' total gain. We used the queue model-based simulator introduced in Section III to conduct the analysis.

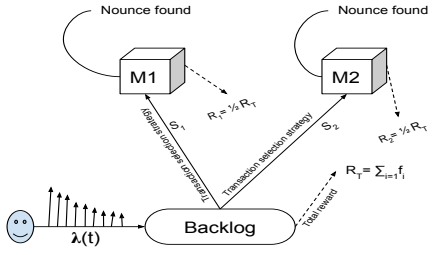


Figure 11. Miners

Table III
Strategy comparison (Block size=1MB)

Strategies	Fee based	Fee per byte	FIFO
Fee-based	(55.23,55.23)	(81.7,28.8)	(67.6,42.85)
Fee per byte	(33.19,77.23)	(55.45,55.46)	(56.89,53.35)
FIFO	(44.43,65.81)	(57.67,52.55)	(55.1,55.1)

Table IV
Strategy comparison (Block size=2MB)

Strategies	Fee based	Fee per byte	FIFO
Fee-based	(55.24,55.24)	(51.1,59.36)	(52.4,58.05)
Fee per byte	(52.24,58.23)	(55.7,55.7)	(59.9,50.6)
FIFO	(50.4,60.10)	(57.70,52.78)	(55.5,55.5)

A. Miner vs Miner

The default strategy stated by the Bitcoin research community is fee per byte. However, it is recommended but not enforced for the miners to follow this strategy. This enables the miners to choose a strategy that fits or increases the financial gain of the mining process, empowering the decision-making of miners to perform a non-cooperative game. In such a game, the Nash equilibrium states that a player can achieve the desired outcome by not deviating from their initial strategy [1].

In the investigation of this subsection, we use the two-miner case to show if the Nash equilibrium exists. Tables III and IV illustrate the results from considering the two miners. These tables also show different block sizes while the arrival intensity is within the range of 3.0 to 3.3 ($\lambda(t) \in [3.0, \ldots, 3.3]$), to show the impact on the amount of gain by the miners. The values inside the table indicate the final gain of the miner while using one of the strategies. For instance, row 2 and column 2 value (55.23, 55.23) means when both M1 and M2 use fee-based, they achieve the same financial gain. This gain is the sum of the transactions' fee picked by the miner utilizing this strategy.

Table III illustrates, there is a dominant strategy in this game for miner M1 and M2, i.e., use fee-based strategy. It is because the maximum payoff for row players in all columns occurs in the first row and first column. When M1 uses fee per byte, M2 maximum payoff occurs when it uses a fee-based strategy. Similarly, when M1 uses FIFO, M2 does best by changing into fee-based. However, M2's best strategy is not to change its current fee-based if M1 uses fee per byte or FIFO.

M1 and M2 have no incentive to change their strategy because fee-based is their dominant strategy. Since M1 uses fee-based in any case, M2's best response is not to change its fee-based strategy because it gets the maximum payoff. Given these facts, the cell gives us the maximum payoff for M2 in the first row. It is the first column that represents M2 not changing its fee-based strategy. Row 1 and column 1 hence shows a Nash equilibrium.

Table IV demonstrates the impact in terms of final reward distributions. As the case for 1MB, M2 using a fee-based strategy is dominant in this case. M2 achieves maximum payoff when the M1 uses the FIFO method. Similarly, M1

Table V
STRATEGY COMPARISON (BLOCK SIZE=1MB)

Strategies	Fee based	Fee per byte	FIFO
Fee-based	(22.23,22.23)	(35, 18.75)	(32.2, 19.5)
Fee per byte	(10.13,25.1)	(21.7,21.7)	(23.2,21.75)
FIFO	(13.13,24.2)	(21.5,22.2)	(21.6,21.6)

Table VI
STRATEGY COMPARISON (BLOCK SIZE=2MB)

Strategies	Fee based	Fee per byte	FIFO
Fee-based	(22.23,22.23)	(25, 21.25)	(22.2, 21.95)
Fee per byte	(20.9,22.27)	(22,22)	(24,21.5)
FIFO	(22.13,21.96)	(18.5,22.88)	(22.1,22.1)

does better when M2 uses FIFO. Both M1 and M1 achieve the best when both use the same strategies.

Since for miner M2, using a fee-based strategy is a dominant strategy, it makes M2 have little incentive to change its strategy, which will leave M1 to also change to fee-based. In this sense, column 1 and row 1 is a Nash equilibrium.

B. Miner vs Miners

In this case, we considered five miners. Each miner has an equal probability of chance in generating a valid block and earning the reward. Four miners follow the same strategies while one miner chooses a different or the same strategy as the others. Same as the previous case, the arrival intensity is within range of 3.0 to 3.3 ($\lambda(t) \in [3.0, \ldots, 3.3]$). Furthermore, the block size is fixed to 1MB or 2MB. In this section, M1 represents a miner with an independent incentive to change the strategy to increase the gain. However, M2 represents the other four miners following the same strategy while creating a block. Table V and VI shows miners gain from adopting different block creation strategy. The values shown as (M1, M2) indicate the final gain of miner M1, and what each of the other four miners earns.

Table V and VI demonstrate that using a fee-based strategy increases the gain of single or grouped miners. When a miner uses this strategy, it achieves better gain than following another strategy. However, when all five miners use the same strategy, the gain is equally divided. This result shows that when all the miners follow the same strategy, the reward is equally divided. Otherwise, miners can adopt different strategies to increase financial gain. This implicitly encourages miners to adopt or change their strategies to achieve higher financial gain, regardless of transactions that give smaller gains may take longer to be processed than expected.

VI. DISCUSSION

There has been some research work proposing schemes and methods in increasing the throughput of Bitcoin [39]. These proposals focus on either increasing block size [12, 30] or validating transactions outside of the main chain [26, 36]. From the Bitcoin design perspective, the average inter-block generation time is 10 minutes, making the previous blocks reach all the nodes in the network [4]. Around 18.5 million mined bitcoin are circulating on the network as of July 2020[2]. Since its inception in 2008, there has been a growing interest in studying Bitcoin. Despite its popularity, slow transaction processing speed is one of the fundamental issues that make Bitcoin struggling to address. The ever-increasing issue of smaller fee transactions waiting a long time to be processed was started around April 2017 and is still not addressed. In around April - August 2017, the throughput's reductions had been very steep. The Bitcoin community was forced to extend the block size by 1 MB to reduce the number of transactions waiting for confirmation. In [42], we can also observe the increase of the throughput monotonically after soft-fork extensions. However, based on the number of applications that integrate the bitcoin service, there is no guarantee we will not face the same issues in the future.

Increasing the block size may require more than the default average inter-generation time to propagate. As reported in Fig. 7 and 10, increasing the block size may process more transactions per block, but the low-fee transactions still suffer from a long waiting time. Increasing the block size increases the block propagation time and impacts the consistency of the ledger [9]. In addition, the backlog of transactions awaiting inclusion in future blocks will clog up the bitcoin network. The bitcoin nodes which form the collective backbone that relays transactions across the network, will be overloaded with data, and some transactions could be severely delayed or even rejected altogether. Similarly, shortening the block generation interval increases the fork rate in the system, which compromises the platform's security [9]. Hence, the main issue resides in the proper management of the backlog, which requires an independent investigation to improve the technology's quality of service.

VII. CONCLUSION

In this paper, we analyzed the transaction waiting time for Bitcoin. Specifically, we modeled the transaction waiting time process as a single server with batch processing and different transaction selection strategies. We considered that transaction priority is only dependent on the transaction fee and size. To study the transaction waiting time, we developed a single node simulator/emulator that captures the workflow of Bitcoin. The proposed model shows that transactions with a minimal fee or fee per byte sufferers from a long waiting time even with the maximum block size.

In addition, we performed analysis on the impact of a miner's transaction selection strategy on the final gain or loss. The analysis shows that when miners use the same strategy, the average income between the miners is equally divided. However, when miners choose a different strategy, they can achieve different gain relative to the opponent strategy. Other than the transaction selection strategy, we also showed that the block size also impacts miners to choose which method to

[2]https://www.blockchain.com/charts/total-bitcoins

choose from, mainly because it decides the number of transactions. These results show that increasing block size alone may not bring optimal solutions. Performing an independent investigation on the backlog to introduce fairness in terms of waiting time to the minor transactions is needed.

REFERENCES

[1] J. Nash. "Equilibrium Points in N-Person Games." In: *Proceedings of the National Academy of Sciences of the United States of America* 36 1 (1950), pp. 48–9.

[2] PA W Lewis and Gerald S Shedler. "Simulation of nonhomogeneous Poisson processes by thinning". In: *Naval research logistics quarterly* 26.3 (1979), pp. 403–413.

[3] Y. Ogata. "On Lewis' simulation method for point processes". In: *IEEE Transactions on Information Theory* 27.1 (1981), pp. 23–31.

[4] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". In: *Cryptography Mailing list at https://metzdowd.com* (Mar. 2009).

[5] J. Bonneau et al. "SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies". In: *2015 IEEE Symposium on Security and Privacy*. 2015, pp. 104–121.

[6] Ittay Eyal. "The Miner's Dilemma". In: May 2015, pp. 89–103.

[7] Aron Laszka, Benjamin Johnson, and Jens Grossklags. "When Bitcoin Mining Pools Run Dry: A Game-Theoretic Analysis of the Long-Term Impact of Attacks Between Mining Pools". In: Jan. 2015.

[8] R. Pasupathy and K. Nagaraj. "Modeling dependence in simulation input: The case for copulas". In: *2015 Winter Simulation Conference (WSC)*. 2015, pp. 1850–1864.

[9] Arthur Gervais et al. "On the Security and Performance of Proof of Work Blockchains". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. Vienna, Austria: Association for Computing Machinery, 2016, pp. 3–16. ISBN: 9781450341394.

[10] Peter R. Rizun. "A Transaction Fee Market Exists Without a Block Size Limit". In: 2016.

[11] Samiran Bag, Sushmita Ruj, and Kouichi Sakurai. "Bitcoin Block Withholding Attack: Analysis and Mitigation". In: *IEEE Transactions on Information Forensics and Security* 12.8 (2017), pp. 1967–1978.

[12] J. Göbel and A.E. Krzesinski. "Increased block size and Bitcoin blockchain dynamics". In: *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*. 2017, pp. 1–6.

[13] Yoshiaki Kawase and Shoji Kasahara. "Transaction-Confirmation Time for Bitcoin: A Queueing Analytical Approach to Blockchain Mechanism". In: Aug. 2017, pp. 75–88. ISBN: 978-3-319-68519-9.

[14] Yi Liu et al. "An Intelligent Strategy to Gain Profit for Bitcoin Mining Pools". In: *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*. Vol. 2. 2017, pp. 427–430.

[15] Mehrdad Salimitari et al. "Profit Maximization for Bitcoin Pool Mining: A Prospect Theoretic Approach". In: *2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC)*. 2017, pp. 267–274.

[16] Beltran Fiz Pontiveros, Stefan Hommes, and Radu State. "Confirmation Delay Prediction of Transactions in the Bitcoin Network". In: Jan. 2018, pp. 534–539.

[17] Beltran Fiz Pontiveros, Robert Norvill, and Radu State. "Monitoring the transaction selection policy of Bitcoin mining pools". In: Apr. 2018, pp. 1–6.

[18] Y. Kawase and S. Kasahara. "A Batch-Service Queueing System with General Input and Its Application to Analysis of Mining Process for Bitcoin Blockchain". In: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 2018, pp. 1440–1447.

[19] J. Li et al. "Transaction Queuing Game in Bitcoin BlockChain". In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. 2018, pp. 114–119.

[20] Quan-Lin Li, Jing-Yu Ma, and Yan-Xia Chang. "Blockchain Queueing Theory". In: (Aug. 2018).

[21] Xiaolei Sun, Mingxi Liu, and Zeqian Sima. "A Novel Cryptocurrency Price Trend Forecasting Model Based on LightGBM". In: *Finance Research Letters* (Dec. 2018).

[22] Y. Aoki et al. "SimBlock: A Blockchain Network Simulator". In: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2019, pp. 325–329.

[23] Damiano Azzolini, Fabrizio Riguzzi, and Evelina Lamma. "Studying Transaction Fees in the Bitcoin Blockchain with Probabilistic Logic Programming". In: *Information* 10.11 (2019). URL: https://www.mdpi.com/2078-2489/10/11/335.

[24] R. Banno and K. Shudo. "Simulating a Blockchain Network with SimBlock". In: *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. 2019, pp. 3–4.

[25] Saulo Dos Santos et al. "An Efficient Miner Strategy for Selecting Cryptocurrency Transactions". In: *2019 IEEE International Conference on Blockchain (Blockchain)*. 2019, pp. 116–123.

[26] Enes Erdin et al. "A Heuristic-Based Private Bitcoin Payment Network Formation Using Off-Chain Links". In: *2019 IEEE International Conference on Blockchain (Blockchain)*. 2019, pp. 294–301.

[27] C. Faria and M. Correia. "BlockSim: Blockchain Simulator". In: *2019 IEEE International Conference on Blockchain (Blockchain)*. 2019, pp. 439–446.

[28] S. Geissler et al. "Discrete-Time Analysis of the Blockchain Distributed Ledger Technology". In: *2019 31st International Teletraffic Congress (ITC 31)*. 2019, pp. 130–137.

[29] S. Jiang and J. Wu. "Bitcoin Mining with Transaction Fees: A Game on the Block Size". In: *2019 IEEE International Conference on Blockchain (Blockchain)*. 2019, pp. 107–115.

[30] Suhan Jiang and Jie Wu. "Bitcoin Mining with Transaction Fees: A Game on the Block Size". In: *2019 IEEE International Conference on Blockchain (Blockchain)*. 2019, pp. 107–115.

[31] Jun.Kawahara Shoji.Kasahara. "Effect of Bitcoin fee on transaction-confirmation process". In: *Journal of Industrial and Management Optimization* 15.1547 (2019), p. 365.

[32] S. M. Fattahi, A. Makanju, and A. Milani Fard. "SIMBA: An Efficient Simulator for Blockchain Applications". In: *2020 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*. 2020, pp. 51–52.

[33] S. Liaskos, T. Anand, and N. Alimohammadi. "Architecting blockchain network simulators: a model-driven perspective". In: *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. 2020, pp. 1–3.

[34] J. Mišić et al. "Modeling of Bitcoin's Blockchain Delivery Network". In: *IEEE Transactions on Network Science and Engineering* 7.3 (2020), pp. 1368–1381.

[35] S. G. Motlagh, J. Misic, and V. B. Misic. "Modeling of Churn Process in Bitcoin Network". In: *2020 International Conference on Computing, Networking and Communications (ICNC)*. 2020, pp. 686–691.

[36] Hossein Rezaeighaleh and Cliff C. Zou. "Efficient Off-Chain Transaction to Avoid Inaccessible Coins in Cryptocurrencies". In: *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*. 2020, pp. 1903–1909.

[37] S. Smetanin et al. "Modeling of Distributed Ledgers: Challenges and Future Perspectives". In: *2020 IEEE 22nd Conference on Business Informatics (CBI)*. Vol. 1. 2020, pp. 162–171.

[38] C. Wang, X. Chu, and Y. Qin. "Measurement and Analysis of the Bitcoin Networks: A View from Mining Pools". In: *2020 6th International Conference on Big Data Computing and Communications (BIGCOM)*. 2020, pp. 180–188.

[39] Qiheng Zhou et al. "Solutions to Scalability of Blockchain: A Survey". In: *IEEE Access* 8 (2020), pp. 16440–16455.

[40] Befekadu G Gebraselase, Bjarne E Helvik, and Yuming Jiang. "An Analysis of Transaction Handling in Bitcoin". In: *arXiv preprint arXiv:2106.10083* (2021).

[41] Befekadu G. Gebraselase, Bjarne E. Helvik, and Yuming Jiang. "Transaction Characteristics of Bitcoin". In: *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 2021, pp. 544–550.

[42] Btc Block Explorere. *Block Explorer*. URL: https://btc.com/. (accessed: 01.07.2020).

[43] Python. *simulation*. URL: https://simpy.readthedocs.io/en/latest/. (accessed: 01.07.2020).

Paper D:
B. G. Gebraselase, B. E. Helvik and Y. Jiang, "Bitcoin P2P Network Measurements: A testbed study of the effect of peer selection on transaction propagation and confirmation times," 2022 IEEE Transactions on Network and Service Management

Bitcoin P2P Network Measurements: A testbed study of the effect of peer selection on transaction propagation and confirmation times

Befekadu G. Gebraselase, Bjarne E. Helvik, Yuming Jiang
Department of Information Security and Communication Technology
NTNU, Norwegian University of Science and Technology, Trondheim, Norway
{befekadu.gebraselase, bjarne, yuming.jiang}@ntnu.no

***Abstract*—Bitcoin is the first and the most extensive decentralized electronic cryptocurrency system that uses blockchain technology. It uses a peer-to-peer (P2P) network to operate without a central authority and propagate system information such as transactions or blockchain updates. The communication between participating nodes is highly relying on the underlying network infrastructure to facilitate a platform. Understanding the impact of peer formation strategies, peer list, and delay is vital in understanding node to node communication and the system performance. Therefore, we performed an extensive study on the transaction characteristics of Bitcoin through a testbed. The analysis shows that peer selection strategies affect the transaction propagation and confirmation times. In particular, better performance, in terms of smaller transaction confirmation time and lower number of temporary forks, may be achieved by adjusting the default nearby-based peer selection strategy.**

***Index Terms*—Bitcoin, P2P, Peer selection strategies, Transaction characteristics**

I. Introduction

Bitcoin is becoming the leading cryptocurrency system today, with its value rising dramatically since its launch in 2009 [7, 30]. Satoshi Nakamoto, the pseudonym of Bitcoin's creator, stated that Bitcoin is an electronic payment system based on cryptographic proof instead of trust [30]. Bitcoin is the first well-known decentralized electronic peer-to-peer (P2P) system that uses blockchain technology [7, 30]. It adapts a cryptographic proof of work (PoW) mechanism that allows anonymous peers to create and validate transactions through the underlying P2P network [30]. The P2P network is vital to the communications of the blockchain system [10][15]. The nodes send and receive messages via the underlying network infrastructure while the P2P topology is formed at the application layer [25]. The way nodes form an overlay topology affects the overall performance, such as transaction confirmation time [40], block and transaction propagation delay [7][16], fork rate [7], and stability of the ledger. In this regard, we prepared a testbed to analyze the impact of P2P topology formation, end-to-end delay, and bandwidth limitation on the performance of Bitcoin.

Bitcoin operates to distribute the ledger among all the participants in a flooding P2P network [9]. When a node tries to join the Bitcoin network, it uses a hardcoded seed to reach out to the nodes nearby. Through getaddress and node discovery, each node updates/creates eight peers by default (outgoing connection), but it can have up to 124 inbound connections. The logical connections between participating nodes create a dense P2P overlay topology, a mesh network [10][11]. This P2P topology is responsible for broadcasting new updates to peers by which they learn and inform each other about transactions and blocks [10]. The reachability of these messages affects the ability of the system to process more transactions and secure the interactions [10][15].

In Bitcoin, the average inter-block generation time is 10 minutes. This enables all the newly generated blocks to reach a maximum number of nodes in the network. Shortening the inter-block generation time brings higher block propagation delay, which increases the temporary fork rate [37][39], which wastes the miner's resource and makes the transactions wait longer. Alternatively, increasing the inter-transaction generation also increases propagation delay, affecting the confirmation waiting time of the transaction [7]. Likewise, the peer formation strategies also impact the reachability of the transactions and blocks [2][38]. The nodes forward new updates to the peer nodes, in which the number of peer nodes and the delay in between impact the amount of time needed to forward a message. The network element´s delay, processing delay, and peer formation strategies affect the block's number of minutes to reach the maximum number of nodes.

This paper investigates network-related parameters' impact on the technology's overall performance. However, it is challenging to conduct such analysis because the nodes are independent and anonymous, making it challenging to collect measurement data from the unknown nodes. In addition, measuring a network fragment will bring results not representative of the overall performance. Several methods have been proposed in the literature to examine network condition effects on the performance. One is to use analytical models, which, however, are built on much simplified assumptions or approximations to allow tractable analysis, e.g., [28, 39], unable to reflect the actual Bitcoin P2P network environments. Another is to use simulation tools, e.g., [1, 2], which, however, have also made

much simplification and do not exactly implement / reflect the set of mechanisms used by the Bitcoin P2P network. The third is to develop an emulator that behaves like an actual Bitcoin [20]. For this reason, we choose to develop an emulator.

A testbed emulator has been prepared to perform a measurement-based study. As a highlight, the testbed includes 104 Raspberry Pis, six switches, and two-blade racks. Each blade rack can hold up to 40 Raspberry Pis. Each raspberry device has Bitcoin Core 0.21.0 [4] installation with additional scripts to automate transactions and block generation events. Through this testbed, a dataset has been gathered containing primary information about the chain, i.e., the ledger, and information that is not available from the ledger but measured from the local mining pool (mempool). Based on the collected dataset, an explorative study on the transaction characteristics of Bitcoin has been conducted.

This paper investigates the effect of peer selection on transaction propagation and confirmation times. In particular, the aim is to provide valuable insights into the impact of network conditions on transaction confirmation time. The paper's main contributions are the following:

- The paper presents a testbed development that can be used to examine the transaction characteristics of Bitcoin, including transaction propagation and confirmation times.
- The investigation shows that peer selection has substantial impact on the performance. In particular, adding random peer selection can improve the transaction propagation and confirmation times.
- It is also found that some transactions, particularly when the load is high, need to wait much longer than the expected 3600 seconds to get confirmed [30], and the occurrence of temporary forks, in addition to load, also contributes to this.

The rest of the paper is organized as follows. The current state of the art is covered in Section II. Next, Section III illustrates the testebd setup and what kind of parameters considered. Then, Section IV illustrates the workflow of transaction handling in Bitcoin. Section V describes the input parameters used for the prepared setup. Following that, how P2P topology formation and the strategies proposed are discussed in Section VI. Next, Section IX and X reports results gained from the analysis. Following that, Section XI presents the impact of fork occurrence over the transaction confirmation time. Section XII opens up a discussion on what has been observed in the analysis. Finally, Section XIII concludes the paper.

II. Related Work

There are several works related to studying the impact of bitcoin P2P on the security and performance of the technology. Eisenbarth et al. [10] examined the resilience of bitcoin networks from churn, detection of Sybil nodes, dynamicity, and popularity of peers. Based on one month of observation, the study showed little churn in the network, no Sybil attack, and recent updates on tackling these issues had become effective. Wang et al. [42] developed an Ethereum network analyzer, Ethna, to analyze the P2P network. The analysis showed that the average degree of an Etherium node is 47, and the P2P network of blockchain such as Bitcoin degree of distribution follows a power-law. The network has the characteristics of a scale-free network.

Fadhil et al. [15] proposed locality-based approaches to improve the propagation delay on the P2P network. This study considered clustering nodes in the exact geographical location, where the distance between is used as key on choosing which node to add as a peer. They showed that providing a less distance threshold would improve the transaction propagation delay with a high proportion. However, clustering with known deterministic distance may reduce the security of the network. Essaid et al. [11] proposed a Bitcoin P2P topology discovery framework that tracks the information exchange to discover network topologies. Based on 45 days' observation, the node distribution between the USA and China matches closely, while other parts of the world have fewer active public nodes to discover. Sudhan et al. [41] developed a model to simulate the Bitcoin network and studied the impact of the outgoing connection limit over the transaction propagation time. In addition, the study considered two peer selection strategies, proximity and random. They showed that peer selection strategies impact the transaction propagation delay.

Shahsavari et al. [39] proposed an analytical model to study the network delay and traffic delay in Bitcoin. The study considered the effect of the default number of connections and the block size on the performance of the Bitcoin network. Deshpande et al. [8] developed a fast and efficient framework named BTCmap to discover and map the Bitcoin network topology. The analysis indicates that the online peers' list remains valid (less than 1% of changes) at 56 minutes 40 seconds. Otsuki et al. [33] showed that a relay network improves the propagation time of a block. In addition, the work showed that relay network decrease in the orphan block rate and the 50th percentile of block propagation time. However, the relay network's improvement of the orphan block rate became smaller as the Internet speed increased. Regarding the mining success rate, it was demonstrated that the relay network did not significantly influence the differences between utilizing and non-utilizing nodes which were below 0.1 at any utilization rate.

The authors in [22] proposed KadRTT, an approach that tries to reduce the lookup latency and hop count. The study shows that proximity and uniform ID arrangement methods enable the proposal to improve performance. This improves P2P applications for efficient content lookup mechanisms.

Most of the research work outlined above analyzes the discovery of Bitcoin P2P topology or develops a framework to crawl the live Bitcoin network to discover the structure and security breach of the technology. However, little has been investigated about the impact of peer selection, topology formation, and end-to-end delay on the transaction characteristics of Bitcoin. Therefore, we developed a testbed that mimics the real Bitcoin network, enabling us to experiment with collected data and make further analyses.

III. Measurement Setup and Node Configuration

For the measurement study, a testbed has been implemented to record information about Bitcoin transactions. The testbed includes 104 Raspberry Pis, six switches, two-blade rack (each holding 40 Raspberry Pis). Each Raspberry Pi has an installation of a full Bitcoin core.

A. Node configuration

Every Raspberry Pi is used as a full node that participates in addition, validation, and generating valid logs. These devices boot from an SD card. The SD card has Ubuntu Server Version 20.10 for the ARM architecture. In addition, the SD cards contain the scripts necessary to run the setup, for instance, scripts to start Bitcoin daemon, adding topology and delay and generating transactions and blocks.

1) Network configuration: Each node interface is configured with an IP address 192.168.xx.1/24. Subnetting with /24 may not be necessary to have a single node, but we plan to increase nodes per subnet for the future use case. Assigning such an IP address also mimics an actual Bitcoin node with its public address. Since each node becomes part of the network, we used VPP (Version 21.6) to perform routing between the nodes. It is an open-source software that provides high-performance switching and routing features for commodity hardware [43].

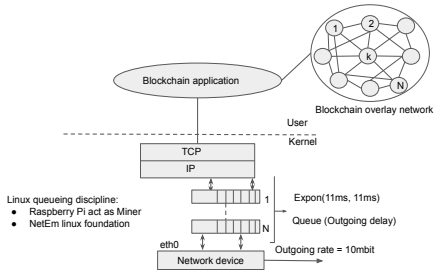


Figure 1. Bitcoin node configuration

The basic architecture of Linux queuing disciplines is shown in Fig. 1. The queuing disciplines exist between the protocol output and the network device, and the default queuing discipline is a simple packet FIFO queue. A queuing discipline is a simple object with two key interfaces. One queues packets to be sent, and the other releases packets to the network device. The queuing discipline makes the policy decision of which packets to send based on the current settings. As shown in Fig. 1, the packet leaving each node adds delay to each packet which follows an exponential distribution. Since each node has an N peer list, we can also see N queues. In addition, the bandwidth is limited to 10 Mbps capacity. These configurations mimic the real Bitcoin network's peer list, and delay arises from the node and network capacity limitations. To simulate a network of the whole Bitcoin network, we used NetEM. It provides Network Emulation functionality for testing protocols by emulating the properties of wide-area networks [19].

To emulate network traffic, the NetEm emulator provides Normal and Pareto distributions [19, 21]. However, the literature study, e.g., [17], has revealed that the inter-packet delay in Bitcoin follows more closely an exponential distribution. This is another challenge since the NetEm does not provide this distribution but allows users to add their distribution. There are different ways to prepare a user-defined distribution. For instance, extracting the RTT values from ping statistics gives the mean and standard deviation, then using it in the NetEm command when activating the distribution table produced. This is easy to do between a few nodes. Our setup mimics the actual Bitcoin network of 5670-7279 active full nodes [9][11]. The Bitcoin documentation states that a node chooses a peer within shorter latency. We generated random variables by inverse transform sampling of exponential distribution based on this fact and then used iproute2 marketable to create an exponential distribution. We set the delay (d) between 11 ms, and it is a shorter end-to-end delay to add nodes. This 11 ms is extracted from an independent full Bitcoin node [16], where we calculated the delay between the eight peers from this node and took the minimum delay between the node and its peer which was 11 ms.

2) Node to node delay: In the previous subsection, we discussed why NetEM is used to add delay and bandwidth limitation to emulate the underlying wide area network (WAN) of Bitcoin. This section shows how independent nodes communicate with each other through an open-source software router Vector Packet Processing (VPP) [43]. Nodes add delay d to each outgoing packet. The outgoing packet passes through the router and reaches the destination. Fig. 2 illustrates node to node communication delay between Node_i and Node_k where a Dell computer is used as the router. The VPP open source software router is configured in Dell OPTIPLEX 9020, with a specification of Intel® 4th generation Core™ i7/i5 Quad Core, Ubuntu 20.04, 32GB memory, Integrated Intel® I217LM Ethernet LAN 10/100/1000, and 256GB storage capacity.

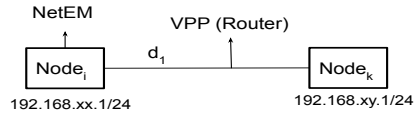


Figure 2. Node adding delay

B. Time synchronization

The devices have to be time-synchronized to enable accurate time stamping by each node in the network. For this reason, we used a well-known time synchronization application called Network Time Protocol (NTP). NTP is an application that allows computers to coordinate their system time [26, 36]. The implementation is in userspace rather than in kernel mode; however, its performance is much better than the other network time protocols [36]. Usually, it is available for most

Linux distributions, which makes it easier to integrate with applications. We have 104 nodes that generate events that require accurate timing and synchronization. Therefore, we used NTP in our setup, where, node 1 acts as an NTP server, while the rest 103 nodes act as clients. The nodes synchronize time means to set them to agree at a particular epoch with respect to coordinated universal time (UTC) [26]. Fig. 3 shows how NTP is added to the setup. As we can see from Fig. 3, node 1 is the NTP server, while the rest 103 nodes are the NTP clients.

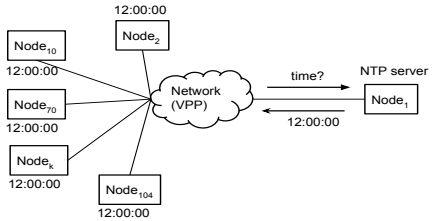


Figure 3. Time synchronization

C. Raspberry Pi specification

The Raspberry Pi devices are running the Bitcoin protocol through Bitcoin Core 0.21.0. To identify them, each device is given a unique number from 1 to 104. These devices act as full nodes, and a single device will be referred to as *node n* where n is the given number. As we see from Table I, in total, the setup has 93 "Raspberry Pi 3´´ devices and 11 "Raspberry Pi 4´´ devices. There are some differences between Raspberry Pi 3 and Pi 4 that are relevant for the setup. Raspberry Pi 4 Plus has a CPU clock speed of 1.5 GHz, 0.1 GHz more than Raspberry Pi 3, which has a clock speed of 1.4 GHz. Additionally, while Raspberry Pi 3 has an Ethernet port with a maximum throughput of 300 Mbps, Raspberry Pi 4 has Gigabit Ethernet.

Table I
RASPBERRY PI MODELS

	Raspberry Pi 3 Model B+	Raspberry Pi 4
Processor	1.4 GHz	1.5 GHz, 64 bit CPU
Memory	1GB RAM	1-4GB RAM
WiFi	2.4GHz Wireless LAN	2.4Ghz and 5Ghz Wireless
Ethernet	300Mbps	Gigabit Ethernet
SD card	8-16 GB	8-16 GB
# nodes	93	11

IV. THE WORK FLOW OF BITCOIN

This section gives essential background on how Bitcoin handles transactions, in addition to how the nodes communicate and discover each other.

1) Workflow: Fig.4 illustrates the workflow of transaction arrival, block formation, propagation, and validation in Blockchain. Briefly, after transactions are generated by the users, they are sent to all full (validation) nodes. Upon the

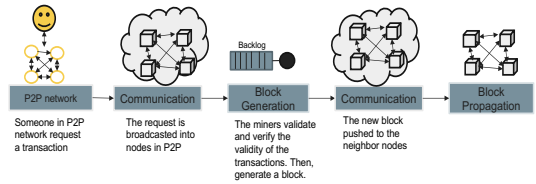


Figure 4. Blockchain process flow

arrival of a transaction at a full node, the node stores the transaction in its backlog (memory pool), waiting for confirmation. Besides, the node may choose unconfirmed transactions in the backlog to pack into a new transaction block. If the puzzle finding is successful, this newly generated block is added to the Blockchain. This information is sent to all the nodes. At each node, the validity of the newly generated block is checked. If the validity is confirmed with consensus, the updated Blockchain is accepted, and the new block of transactions are validated. Such validated transactions are removed from the mempool at each full node that then repeats the above process.

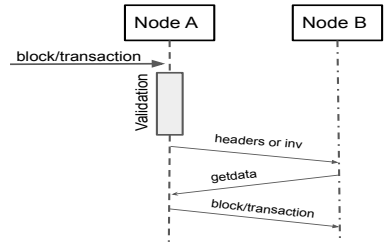


Figure 5. Legacy relaying

2) Node to node interaction: Bitcoin nodes form a P2P network, while each node by default can have eight peer list. It is a logical link that allows peers to push/pull new updates to the neighbors. Fig. 5 shows node to node message exchange sequence. The new arrival block or transaction picked up by Node A. Then, a block/transaction is validated (the bar) by Node A, which then sends an inv message to Node B requesting permission to send the block. Node B replies with a request (getdata) for the block/transaction, and Node A sends it.

3) Network discovery: A Bitcoin node is allowed to maintain up to 132 connections (maxconnections) as default, of which 8 are outgoing connections and the rest are incoming connections. Peers listen on port 8333 for inbound connections. When a node wants to join the network, as it is a public blockchain, the node uses DNS names (called DNS seeds) hardcoded in the Bitcoin Core. From this point, the new node updates its peer list by discovering nodes nearby. In this way, new nodes select peers that are part of the network. This peer formation is called nearby-based since it highly depends

on adding nearby nodes. The peers randomly choose logical neighbors without knowledge about the underlying physical topology.

The peer list used as a reference list to send an inventory or receive messages from the neighbor nodes. After the node joins the network, it can take part in propagation, consensus, and block generation. These nodes act as full nodes, which means the users/owners can create transactions and create blocks, and forward the new updates to the network. Each block created by the nodes, which is valid enough to be included in the chain will contain the hash of previous records of the blocks. Blocks that are created but ignored by the network become orphan blocks. Mostly these blocks become fragments that will never be used but waste all the computation cost and resources.

4) Peer list: Nodes can have up to 132 connection lists. This is the combination of incoming and outgoing peers. When a node initiates the connection, it is called outgoing, or if the connection initialization comes from other nodes, it is incoming bound. The number of peers (P) represents the number of outgoing peers of each node. The total connection list is the sum of P outgoing peers plus incoming peers (Q). In this work, the peer list length (p_l) is set to be $2P$, i.e. $Q = P$.

V. Setup of Input Parameters

This section describes the input parameters such as inter-transaction generation time, inter-block generation time, and node to node delay added to the network.

The transaction and block generation events must also include similar characteristics to mimic the Bitcoin network. The transaction inter-arrival time to a node follows an exponential distribution, based on the literature investigation [17][40]. Similarly, the inter-block generation time also follows an exponential distribution [17][18].

Algorithm 1 Generate transaction

```
1: procedure POISSON(λ(t), T_d)
2:     Initialisation: T_t = timenow() + T_d
3:     Condition: T_d ≤ T_t
4:     while True do
5:         w_t ∼ negExp(λ(t))
6:         if timenow() + w_t < T_t then
7:             time.sleep(w_t)
8:             generateTransaction()
```

1) Transaction inter-generation time: Each node acts as a full Bitcoin node that creates, validates, and propagates transactions and blocks. Therefore, nodes have a script that generates transactions and blocks following an exponential distribution. The script accepts duration and the inter-generation interval in terms of seconds as an input parameter, as illustrated in algorithm 1. $1/\lambda(t)$ is the mean inter-generation time (t_{gi+1} - t_{gi}) in seconds for each node. Furthermore, T_d is the total duration of running time in seconds. The result of the inter-generation time distribution follows an exponential distribution.

2) Block inter-generation time: Bitcoin network generates a block on average 10 minutes. This makes the recent block propagate to the network before the next generation. Bitcoin adjusts the difficulty after 2016 blocks are generated to control the average inter-block generation time. Although this is true for live Bitcoin nodes, the Bitcoin core regtest mode has difficulty close to zero, which means there is no difficulty generating a block. However, to mimic the real Bitcoin network, we developed a script that produces a block on average ten minutes. Overall, we have 104 nodes, which means a block is generated in 103*600 second (61800), the remaining 1 node is measurement node. Similar to the previous transaction generation case, here the Algorithm 2 takes the generation rate and duration of the simulation in seconds as an input.

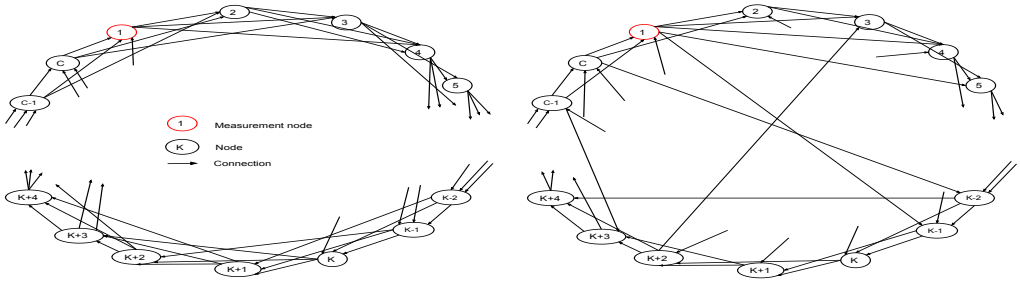
Algorithm 2 Generate Block

```
1: procedure POISSON(λ(t), T_d)
2:     Initialisation: T_t = timenow() + T_d
3:     Condition: T_d ≤ T_t
4:     while True do
5:         w_t ∼ negExp(λ(t))
6:         if timenow() + w_t < T_t then
7:             time.sleep(w_t)
8:             generateBlock()
```

3) Node-to-node delay: In the actual Bitcoin network, nodes are distributed across the globe, which are geographically and domain-wise isolated from each other. Since the underlying network infrastructure is providing the communication platform and the actual network traffic is unpredictable, it is common to consider a distribution that captures the network delay between two participating ends. To mimic the delay that arises from the network element and distance between the participating nodes we introduced a delay (d) that follows an exponential distribution with the shorter mean of 11 ms.

VI. Network Topology

The Bitcoin research community states that peer formation starts from looking at DNS seed nodes. Some of those DNS seeds provide a static list of IP addresses of stable Bitcoin listening nodes. Once a peer receives a full Bitcoin node IP address list, the peer performs up to eight outgoing connection attempts. These eight nodes that the peer attempts connection with are called entry nodes. A node can request from its neighbors the IP addresses of peers they are aware of using the addr P2P network message and increasing the nodes' awareness nearby. The distance between nodes, such as delay, the number of peers, and how to select the peer affect the overall performance. We consider three peer formation cases to investigate this impact: nearby, random, and mixed. The nearby-based approach is a method that adds neighbor nodes, as stated in the Bitcoin documentation [5]. The other way is randomly selecting peers, as expressed by other authors [7, 41]. Finally, to mix these two approaches to investigate the effect, this method is a mixed approach. The following section introduces these approaches.



(a) Nearby-based approach

(b) Mixed-based approach

Figure 6. Bitcoin overlay network example (P=3), while considering only outgoing links

A. Nearby-based peer selection

Nearby-based peer selection approach enables peers to form close by neighbor peer creating P2P topology. A full Bitcoin node can have eight peers by default, but it can have up to 132 connection link points. The nearby metric depends on adding nodes close by.

Algorithm 3 Nearby-based

1: **procedure** NEARBY(P, k, C)
2: p_l= {l—l= (k+i) mod C, i=1, …,P}

Algorithm 3 illustrates the nearby-based peer selection method. The procedure takes the number of peers to add (P), the current node (k), and the total number of nodes (C). The algorithm adds peers that are closeby.

B. Random-based peer selection

Unlike the nearby-based approach, the random-based method does not depend on the proximity of nodes, instead on the random selection of the peer to add. Even-though Bitcoin is a distributed P2P technology where each node acts as an independent node, it has little knowledge on the global distribution of the nodes. For the random-peer selection method, we consider that nodes know the number of full active nodes in the network they are participating in. Similar to the nearby-based approach, Algorithm 4 illustrates the random peer selection method. The procedure takes the number of peers to add (P), the current node (k), and the total number of nodes (C). The method adds randomly selected nodes as its peer list.

Algorithm 4 Random-based

1: **procedure** RANDOM(P, k, C)
2: Initialisation: $p_l = \{\}, p_c = \{1, \ldots, C\} \setminus \{k\}$
3: **for** $i = 1$ step 1 until P **do**
4: $p_l \leftarrow p_l \cup$ (RANDOM($p_c \setminus p_l$))

C. Nearby + Random (Mixed)-based peer selection

The third case is to combine nearby-based and random-based approaches. In these combinations, the nearby-based method adds $n-1$ peers and the random-based approach adds the last node by choosing randomly. This is to introduce a random link to the nearby-based peer list. Similar to the previous two approaches, Algorithm 5 illustrates the mixed peer selection method. The procedure takes the number of peers to add (P), the current node (k), the total number of nodes (C). As discussed in the previous subsections, the method adds the $n-1$ nodes based on the nearby-based approach. The random-based approach adds the last node.

Algorithm 5 Mixed-approach

1: **procedure** RANDOM(P, k, C)
2: $p_l = \{l | l = (k+i)\ mod\ C, i = 1, \ldots, P-1\}$
3: $p_c = \{1, \ldots, C\} \setminus \{k\} \setminus p_l$
4: $p_l \leftarrow p_l \cup$RANDOM(p_c)

From this point on forwarding, we use random to represent the random-based approach, normal for the nearby-based default approach, and mixed for the approach that mixes the two approaches. Fig. 6 shows a sample Bitcoin overlay network ($P = 3$), while considering only outgoing links

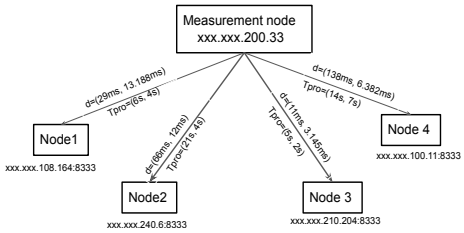
VII. Setup Validation

This section relates the timings in the testbed with those in the live Bitcoin network.

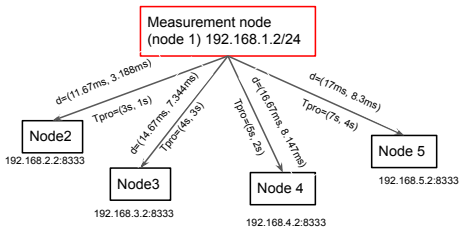
A. Node to node delay

In our previous work [16], an independent Bitcoin full node was deployed to collect transactions and block related feature sets. We used observations from this node to validate some of the input parameters and results. For instance, our nodes have 132 connected nodes. Eight of these nodes are peer nodes, while the rest are incoming bound nodes. The average ping delay between these nodes from the Bitcoin application is 156.20 ms with a standard deviation of 152.23 ms. This ping is handled in a queue with other commands in the application layer to include the processing backlog. However, we also conducted further analysis to ping these nodes from outside of the Bitcoin core, which resulted in an average of 80 ms second in deference. This 80 ms accounts for processing backlog.

As previously mentioned, the eight peers are more important than the others. These peer nodes synchronize more often



(a) Bitcoin live full node



(b) Testbed

Figure 7. Transaction propagation delay between active full nodes, where $\lambda = 3$ transactions per second per network, eight outgoing peers per node

than the other 124 incoming bound nodes. For this purpose, we conducted an independent investigation to see the delay between our node to eight peer nodes. Our analysis shows that the minimum delay between our node and the other nodes is 11ms with a standard deviation of 7ms. This 11 ms delay is used in our setup as a minimum delay guarantee between nodes.

B. Information propagation

This subsection discusses how fast a transaction propagates in the Bitcoin network and how the number of nodes impacts this. We considered four publicly available nodes to collect mempool state and compare it with our node. Fig. 7(a) shows the delay between our measurement node in [16] with four peer nodes that provide their state of the mempool in the Bitcoin network. The figure shows only four out of eight nodes because the remaining four nodes were unreachable. This is because some of them are hidden behind firewall and NAT. As we can see from the figure, transaction propagation between nodes can be up to about 20 seconds [14]. This is mainly because the P2P communication protocol makes processing check the validation of each transaction before forwarding an Inv message to its peers. At the same time, nodes that received the Inv message have to check if the transaction is at the mempool or seen before inside a block. The node sends a getdata message and gets the new transaction when the check is completed. Even though the delay between nodes may be less than 100 ms, processing a transaction takes longer.

We tested out the testbed based on the live Bitcoin full node observation to see if similar transaction characteristics occurred. As we can see from Fig. 7(b), the transaction propagation delay between the measurement node, Node 1, and its four peers also varies in the same order. This demonstrates that the timings in the testbed are similar to those in the real Bitcoin network.

A closer check at the mempool status of the four nodes on the Bitcoin network shows that the number of transactions waiting in their mempool varies between peers in an instant of time. A similar observation is found in our emulated network. For instance, each node shown in Fig. 7(b) respectively has 1566, 3976, 3000, 2244, and 2300 transactions waiting at the mempool at one checking time instant

C. Inter-block generation and inter-transaction arrival time

The average inter-block generation time is close to 10 minutes in the actual Bitcoin network. After 2016 blocks are generated, the difficulty of solving the puzzles increases to make sure nodes generate on an average of 10 minutes so that the new block reaches the maximum number of nodes in the network. Similarly, the transaction inter-arrival time to the mempool also follows exponential distribution [16][17]. These parameters are considered in our setup as input parameters.

VIII. Measurement Data Collection

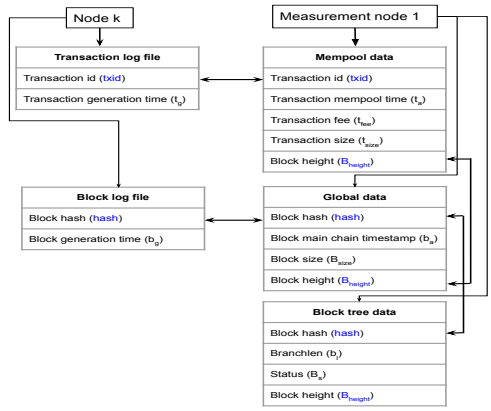


Figure 8. Data collection

A dataset consisting of four parts has been collected by the testbed, as show in Fig. 8. One part of the dataset records each node transaction and block generation events. When a node k, where $k \in [2, 104]$, generates transactions, it records a log about the transaction generation time (t_g) and transactions id ($txid$). Similarly, when a node generates a block, it records the block generation time(b_g) and block hash ($hash$). The second part of the data contains information about each transaction's arrival time at mempool (t_a), transaction size (t_{size}), transaction fee (t_{fee}), transaction id ($txid$), and block height (B_{height}). The block height (B_{height}) in which the transaction belongs can be empty or a number depending on if the transaction is added to the block or just a new arrival. The third part collects information about the blocks from the main chain, such as block hash ($hash$), block size (b_{size}),

block time (b_t), and block height (B_{height}). The fourth part of the data collection contains extracted details about the block tree of the chain, such as Block hash ($hash$), Block height (B_{height}), Branchlen (B_l), and Status (B_s). The Branchlen is the length of the branch in the block tree. It holds 0 for the main chain or a number, indicating the length of the soft fork in terms of the number of blocks in the side chain. The Status (B_s) indicates the Status of the block, whether it is active, part of the main chain, or valid-fork meaning a block is a fork or invalid-block meaning the block is not valid enough to be a candidate.

The second, third and fourth parts of the data are collected from a single node. This node is considered as a measurement node, and in our case, Node 1 is the measurement node. Node 1 is part of the network invalidation and processing transaction at the mempool, but it does not generate transactions or blocks. On the contrary, it collects information about the transactions from its mempool (Mempool data). When the emulation times are over, it also extract information about valid blocks from the main chain (Global data, Block tree data).

Fig. 8 demonstrates the collected feature set from the nodes. As we can see from the figure, measurement Node 1 collects information about the state of the mempool and keeps track of the status of the main chain. It also illustrates the primary key used to link the data set from each device with Node 1. By using the datasets, we performed analysis on transaction propagation ($t_a - t_g$) time and confirmation time ($b_{g(i+6)}$ - $t_g(x)$), where transaction x goes into block i and $i + 6$, representing when the transaction is six-block deep into the main chain.

In addition to the above-collected information, we also extracted the state of the block tree. This information includes which block is fork ($hash$), at which height this event happened (B_{height}), and the number of blocks within the same branch ($B_{branchlen}$). We used these extracted feature sets to count the number of forks that happened while considering different peer formation strategies and how they impact the confirmation time of transactions inside a fork block. These datasets are downloaded and post-processed after the emulation period is completed.

IX. Transaction Propagation Time

The transaction arrival intensity affects the number of transactions waiting at mempool and the number of transactions to be validated and pushed to the network. This section reports results and observations from examining the impact of arrival intensity variation while illustrating the effect of peer list per node.

Bitcoin uses a gossip-like protocol to broadcast updates throughout the network [11]. When a node receives new transactions, it validates and verifies the validity of the transactions, then sends an Inv message to peer nodes to notify them if the peer nodes want these new transactions, before pushing the transaction to the peers. Due to this continuous process, a delay in transaction propagation occurs. The delay combines validation time and the time it takes to disseminate

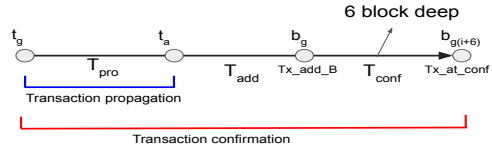
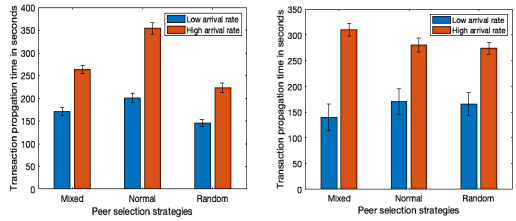


Figure 9. Transaction propagation and confirmation time sequence

the transaction. Fig. 9 shows a time sequence of the life cycle of a transaction. In this section, we focus on the transaction's propagation, and this is the typical time when the transaction is generated (t_g) until it reaches the memppol of a node. Specifically, in our case, the time difference between t_g and t_a is the propagation time, where t_a is the time transaction arrived at the memppol of the measurement node Node 1, and t_g is the time of the transaction generated by one of the nodes (2-104). As illustrated in Fig. 9, the blue line indicates the time length of transaction propagation time. The transaction propagation delay is less than 8 seconds for the default and low-intensity eight peer node case, (see Fig. 7(a)).

1) Average transaction propagation time: Fig. 10 shows the average transaction propagation time in seconds while considering different peer formation strategies. The x-axis represents peer selection strategies, the y-axis represents the propagation delay in seconds, and the legend shows the arrival intensity.



(a) $P = 8$ (b) $P = 4$

Figure 10. Average transaction propagation times for the various peer formation strategies, number of peers P and low (3 t/min) and high (6 t/min) intensity generation rate λ. Error bars indicate 95% confidence intervals from 10 independent runs

Fig. 10(a) and 10(b) illustrate that when the arrival rate is high, which means each node generates on average six transactions per second, in respective of the number of peers per node, the transaction propagation increases. However, with a low arrival rate, three transactions per minute per node, the transaction propagation is less than 170 seconds. In addition, when the number of peers is higher, the normal approach tends to perform worse overall, while random-based peer selection better than the other two.

2) Distribution of transaction propagation time: Fig. 11 and 12 illustrate the distribution of transaction propagation time under a low and a high arrival rate respectively, while the number of peers is fixed to eight. The x-axis represents the propagation time in seconds. The y-axis is the log result of

the distribution $P(t_a - t_g > t)$, while the three peer formation strategies are used.

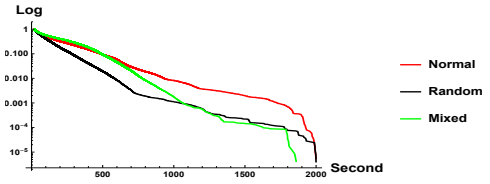


Figure 11. Transaction propagation delay, where $\lambda = 3$ transactions per minute per each node (low intensity)

Fig. 11 shows that, in most cases (80%), transaction propagation in random peer selection has less than 300 seconds propagation time, whereas it has 400 seconds during the mixed approach, while for normal peer selection transactions observe close to 500 seconds propagation time. In all three peer selection approaches, the transaction propagation time can grow more than 1000 seconds in 1% of the cases. Relatively, 90% of the transactions observe propagation time less than 500 seconds for mixed and normal approaches, while random-based peer formation brings less than 450 seconds of propagation time.

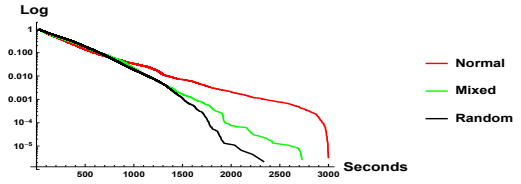


Figure 12. Transaction propagation delay, where $\lambda = 6$ transactions per minute per each node (high intensity)

Fig. 12 also illustrates the transaction propagation delay distribution where the three peer selection approaches are considered. In most cases (80%), the figure reports that transaction propagation in random peer selection has less than 400 seconds propagation time, whereas it has 500 seconds with the mixed and normal peer selection approaches. In all three approaches, the transaction propagation time can grow more than 1500 seconds in 1% of the cases. Relatively, 90% of the transactions see propagation time less than 700 seconds for mixed and normal approaches, while random-based peer formation brings less than 600 seconds of propagation time.

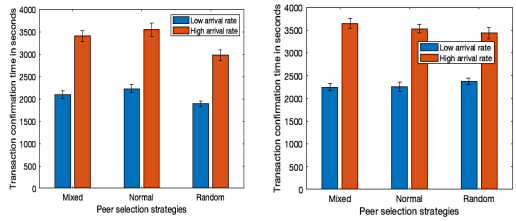
In summary, when the arrival intensity is low, the random-based peer selection strategy performs better, but when the intensity is high, all three strategies produce a comparable propagation delay. This shows arrival intensity has more significant impact on the propagation delay than the type of strategies used or the number of peers.

X. Transaction Confirmation Time

In Bitcoin, the transaction is considered confirmed six blocks deep in the main chain [5]. This tries to ensure no double-spending while maintaining security: By linking the previous block with the other six blocks, it requires more computational effort to modify the confirmed transactions. Thus, to improve security, Bitcoin reduces its performance. This section examines how the arrival intensity, peer list, and end-to-end delay affect performance. In the 'regtest' setup, the transaction is considered valid when it is 101 blocks deep [5]. However, our analysis used six blocks deep for confirmation to obtain results representative of the live Bitcoin blockchain.

Fig. 9 also demonstrates the time sequence of transaction confirmation time. The transaction confirmation time is the difference of the t_g and the t_{conf}. The t_{conf} is the amount of time for the Bitcoin network to generate six valid blocks. Similar to the previous case, t_g is the time a transaction is generated by one of the nodes, and t_{conf} is the time between the blocks from the main chain extracted at node 1. As shown in Fig. 9 the red line indicates the time sequence of the transaction confirmation time. A transaction has to wait until it is six blocks deep. Since a new block is generated every 10 minutes or 60 seconds, this means that the expected transaction confirmation time is 3600 seconds.

Fig. 13 shows the average transaction confirmation time for different peer formation strategies and number of peers. The figure shows that peer formation strategy impacts the overall confirmation time. The x-axis represents the peer selection strategies while the y-axis indicates the confirmation time in seconds. Specifically, Fig. 13(a) and 13(b) show that in respect of the number of peers per node, the arrival rate has a higher impact on the confirmation time. It is worth highlighting that peer formation strategies bring less effect when the arrival rate is lower.



(a) $P = 8$ (b) $P = 4$

Figure 13. Average transaction confirmation times for the various peer formation strategies, number of peers P and low (3 t/min) and high (6 t/min) intensity generation rate λ. Error bars indicate 95% confidence intervals from 10 independent runs

1) Distribution of the confirmation time: Fig. 14 and 15 show the distribution of the transaction confirmation times for low and high arrival rates while the number of peers is fixed to eight. The x-axis represents the confirmation time in seconds. The y-axis is the log of the distribution $P(t_g - b_{gi+6} > t)$, when the three peer formation strategies are used.

Specifically, Fig. 14 illustrates the transaction confirmation time in seconds under a low transaction intensity. The three peer selection strategies are compared. In almost 80% of the cases, random and mixed peer formation strategies produce

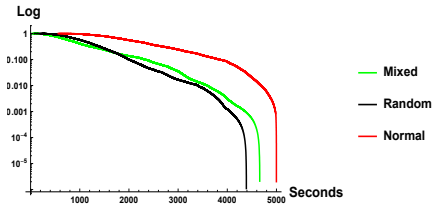


Figure 14. Transaction confirmation time, where $\lambda = 3$ transactions per minute per each node (low intensity)

transaction confirmation time less than 1654 seconds, while the normal approach introduces twice the confirmation time. 1% of the time, mixed and random strategies give confirmation time greater than 2000 seconds, while the normal approach doubles this amount.

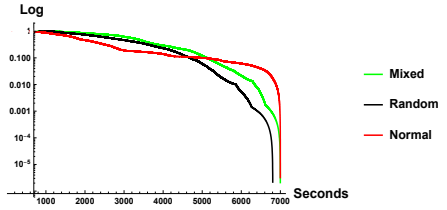


Figure 15. Transaction confirmation time, where $\lambda = 6$ transactions per minute per each node (high intensity)

Fig. 15 reports the transaction confirmation time in seconds under a high transaction intensity. In almost 80% of the cases, random and mixed peer formation strategies produce transaction confirmation time less than 4000 seconds, while the normal approach introduces 1000 seconds less confirmation time. 1% of the time, all the strategies give confirmation time greater than 5000 seconds.

Overall, for low transaction intensity, random peer selection performs better than the other two approaches. However, when we doubled the intensity, it was seen that all strategies yielded more similar distributions. Doubling the arrival intensity also affected the confirmation time. More transactions observe higher confirmation time. This reflects how the P2P protocol fails to propagate the transactions faster but spends significant time validating and processing transactions. It also means the P2P protocol is not good enough to handle high traffic, which has caused a doubt if Bitcoin will be able to catch up with the increasing user demand [24].

XI. Temporary Forking

A temporary fork occurs when two miners independently find and publish a new block referencing the same previous block. In such events, one block becomes an orphan block, where all the transactions not part of the accepted block (valid block) are pulled back to mempool for pickup again, and the miner who generated this block earns nothing for the effort. This affects the performance. The main cause of forking is propagation delay: Without such delay, the notification of a new block would be instantaneously received by all nodes avoiding them to continue working on generating new blocks. Since propagation delay depends also on the network topology that synchronizes between nodes as investigated in the previous sections, the present section is devoted to studying how peer selection strategies may affect the fork generation rate and how forking affects the transaction confirmation time.

A. Introduction to temporary fork

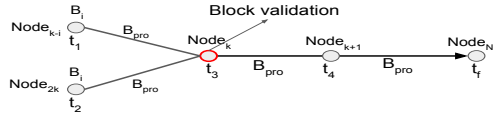
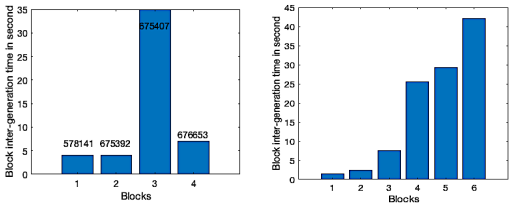


Figure 16. Block propagation time sequence

Fig. 16 illustrates the time sequence of block propagation. Two blocks are generated at time t_1 and t_2 then pushed to the neighbor nodes with some B_{pro} delay. When Node$_k$ receives these two blocks simultaneously, it validates both of them. Suppose both blocks point to the same previous hash of the block. Then the node compares the number of confirmations and an earlier timestamp. It selects one block based on these criteria, increases the confirmation, and forwards it to the neighbor nodes. Similarly, Node$_{k+1}$ will do the same operations, and this will increase the number of confirmation numbers of the valid block that will lead the orphan (fork) block to become less important with time. Once all the N nodes see these two blocks, the network ignores the orphan block while the valid block is added to the main chain [32]. In this way, the Bitcoin network maintains the ledger's consistency and security. However, this temporary fork impacts the overall performance of the technology. The validated transactions in the orphan block which are not part of the valid block are sent back to mempool to wait for pick-up again, increasing the average confirmation time. In addition, miners who created the ignored block (orphan block) wasted considerable resources for little gain.

1) Example: Based on our full independent Bitcoin live node [16], we were able to see four valid forks in the main chain from 578141 to 678853 block height. These four blocks hold from 1200 to 2400 transactions within. The average generation time between two blocks forming a fork is 12.5 seconds, which is much less than the 10-minute average block generation interval. Fig. 17(a) reports the inter-block generation time between fork and valid block in the Bitcoin network. The x-axis represents the blocks where the fork happened, and the y-axis indicates the inter-block generation time in seconds between fork and valid block. As we can see from the figure, the maximum inter-block generation time between valid and fork block is 35 seconds, which happened in the 675407 block height.

Fig. 17(b) shows the block inter-generation time between valid and fork blocks observed in our testbed, under the normal



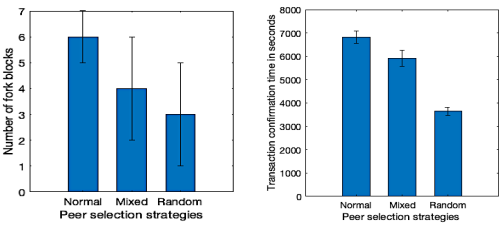
(a) $P = 8$ (b) $P = 8$

Figure 17. Fork vs. valid block example from live bitcoin full node

peer selection approach, high arrival rate, and 8 peers. The figure illustrates that block inter-generated time greater than 40 seconds might increases the high probability of creating a fork event. This plot is to demonstrate what we see from Fig. 17(a), which is from live Bitcoin node, is also seen from our setup.

B. Impact of peer selection strategy on the fork rate

The peer selection strategy impacts the performance of the system, particularly in terms of transaction propagation and confirmation time as discussed in the previous sections, in this subsection, we demonstrate its impact on the occurrence of forks.



(a) Number of forks (b) Confirmation time

Figure 18. Number of forks and their impact on transaction confirmation time

Fig. 18(a) reports the number of fork block happenings under different peer selection strategies, high arrival intensity, and the number of peers per node is 8. As we can see from the figure, the normal peer selection strategy brings a higher number of temporary forks. The mixed and random-based peer selection strategies produce lower numbers of fork blocks.

C. Impact of forking on transaction confirmation time

In the event of forking, transactions inside a fork block return to the mempool for being picked up again. This makes these transactions wait a longer time before confirmation. Fig. 18(b) reports the average transaction confirmation time seen by transactions inside the fork block. When the network ignores the fork block, all the transactions that are not part of the valid block are returned back to the mempool for pickup. The main issue with this is that the fork block may wait for more than one block to be ignored by the network, depending on the length of the pruned branch, which leads to the transaction frozen and waiting for a longer time. Fig. 18(b) also demonstrates this phenomenon. As we can see from the figure, the normal peer selection strategy produces a high number of fork blocks, which leads to transactions waiting longer than 6000 seconds.

Similarly, the mixed peer selection strategy produces a closer number of fork blocks to the normal peer selection approach, and the impact on the transaction confirmation time is more than 5500 seconds. However, the random peer selection strategy performs better than the other two approaches regarding the number of forks and confirmation time, with a transaction confirmation time of fewer than 3670 seconds.

D. Valid vs fork block overlap

When two blocks arrive within a shorter time difference window having the same hash pointing to the previous block, we call it a temporary fork. When this event happens, one of the blocks will become part of the chain, and the other will become an orphan block. Since the comparison is based on the previous block's hash, we further analyze the extent to which valid and fork blocks share the same transactions. Table II shows the overlap in percentage between the valid and fork blocks while considering different peer selection strategies.

Table II
OVERLAP BETWEEN VALID AND FORK BLOCK

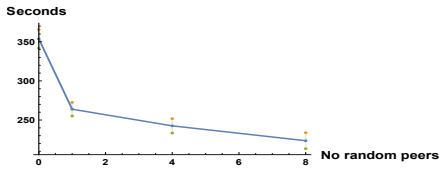
Peer selection strategies	Overlap valid vs fork block $((\mu, \sigma))$
Normal	(88%, 6%)
Mixed	(90%, 5%)
Random	(90%, 3%)

The mixed and normal peer selection strategies produce four to six fork blocks, where 90% of the transactions are the same, but the rest 10% are unique transactions which will be forced to return to the backlog for more waiting time. Similarly, for the random-based strategy, the valid and fork blocks share 90% of the transactions, but the remaining wait more time to be added to the chain.

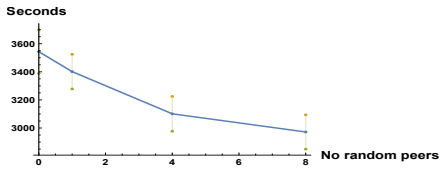
Overall, the peer selection strategy impacts the number of fork occurrences, mainly due to that different strategies give different propagation delays. This section has also showed that fork occurrence can affect the performance significantly. For instance, some of the transactions have had to wait more than 6600 seconds, which is 3000 more seconds of waiting time. This means some transactions have had to wait, on average, 11 valid block generation times.

XII. Discussion

1) Proposed approaches: The P2P formation strategies are essential in propagating information between participating nodes. In this work, we showed that peer selection strategies affect the overall performance of Bitcoin. There have been some research works proposing schemes and methods to reduce the propagation delay in Bitcoin. These proposals focus on either introducing a compact block [23][27] or having some relay nodes [33][34] in the middle to provide a pipeline to push



(a) Propagation time



(b) Confirmation time

Figure 19. Average transaction propagation and confirmation times : number of peers is eight with high (6 t/min) intensity generation rate λ.

more updates to the other nodes. The compact block may introduce better performance in propagating the information based on the available bandwidth between participating nodes [27]. However, this method focuses on how to propagate blocks in the network than how to propagate transactions. Based on our observation, nodes may have a different number of arrivals at the backlog waiting. The compact block method has to push more than half of the block content in such cases.

Using relay nodes to reduce propagation delay is another method proposed by researchers. This method relies on the relay nodes having a higher number of peer nodes from the network, enabling pushing more updates in the network. The main challenge in this approach is that the relay nodes become a security breach or vulnerability point. Attacking these nodes or taking control gives extra incentive to earn more or disrupt the overall activity in the network.

The best strategy to improve the propagation time is perhaps to improve the communication protocol. The protocol spends significant time validating and updating the same transaction. Furthermore, reducing the peer to peer network diameter by having peers other than the nearest may improve. The random strategies investigated in this paper are simple examples. For instance, Fig. 19 shows a specific use case, where the impact of adding more random peers on the performance is provided. The x-axis shows the number of random peers selected, and the y-axis indicates the propagation and confirmation times. Specifically, Fig. 19(a) and Fig. 19(b) show that adding random peers generally improves transaction propagation and confirmation times. In particular, Fig. 19(a) shows that adding one random peer significantly improves the propagation delay with a steep decline. However, this is not comparably visible in confirmation time (see Fig. 19(b)). One reason, as also implied by Fig. 19(a) and Fig. 19(b), is that the Bitcoin P2P protocol spends significant time in validating transactions, e.g.requiring six-block deep in the blockchain to confirm the transactions in the block, dominating the confirmation time.

2) Transaction propagation and confirmation times: The transaction propagation and confirmation times show some values higher than expected. This is because of the impact of the P2P formation strategies and P2P legacy relaying protocol. Some of the transactions have to return to mempool because of fork occurrence. For instance, the normal approach produces more forks than the other two approaches. In such cases, the transactions inside the fork block return to the backlog for pickup, of which some will be added to the new recent block, but others may wait for future block generation events. In addition to this, the processing capacity of the Raspberry Pi devices may contribute to some extent. Although we analyzed to observe the total usage, the Bitcoin, on average, in each device uses 114% CPU and 16% RAM. It is worth highlighting that the Raspberry Pi used has 64 bit quad-core Cortex-A53 and Cortex-A72, which is good enough to handle the traffic generated from Bitcoin and background processing.

3) Impact of temporary fork: The number of fork event occurrences has been reduced recently with the new Bitcoin core release [32]. However, the Bitcoin network is still not tested if it can handle high loads. Based on the current state where 3.3 to 7.2 transactions are processed per second, having arrivals at the mempool from 1700-2600 transactions waiting for pickup [3, 6, 12, 13]. The P2P network may handle processing and propagating updates with some acceptable performance index. However, when we pushed the load to 5500 to 6000 transactions at the backlog, the performance reduced significantly from propagating transactions in 10 seconds into 250-350 seconds. It also impacted the number of fork block occurrences in the network, making some transactions wait more than the expected confirmation time. For instance, for the normal peer formation strategy, the number of prude branches is higher because each node validates new arrivals before propagating to the neighbor nodes. In such cases, more delays happen in the network than having a few random peer links. This shows that the P2P network protocol requires improvement and research to improve its capacity.

XIII. Conclusion

In this paper, we analyzed the impact of peer formation strategies, arrival rate, and the number of peers on the overall performance of the technology. Specifically, we developed a testbed to mimic the Bitcoin P2P network, which enabled us to conduct a comprehensive investigation and gain deep insight into the impact of the underlying P2P network on the performance. The analysis shows that the transaction validation and propagation can take longer than expected, even with a low arrival rate and a high number of connected nodes. In addition, while the peer formation strategy currently adopted by the Bitcoin community is highly reliable in finding peers with low latency response, it does not give the best system performance in terms of propagation and confirmation times and fork rate. Considering a few random nodes in peer selection can improve the performance. These results indicate that the normal peer formation strategy alone may not bring optimal solutions. In addition, these results also imply that

improving the P2P communication protocol, including peer selection and the P2P network topology, has a great potential in improving the performance, including transaction confirmation time.

REFERENCES

[1] Lina Alsahan, Noureddine Lasla, and Mohamed Abdallah. "Local Bitcoin Network Simulator for Performance Evaluation using Lightweight Virtualization". In: *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*. 2020, pp. 355–360.

[2] Ryohei Banno and Kazuyuki Shudo. "Simulating a Blockchain Network with SimBlock". In: *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. 2019, pp. 3–4.

[3] Bitaps. *Today bitcoin blocks*. URL: https : / / bitaps . com / blocks. (accessed: 01.07.2020).

[4] Bitcoin. *Bitcoin Core release 0.21.0*. URL: https://bitcoincore.org/en/releases/0.21.0/. (accessed: 01.07.2020).

[5] Bitcoin. *Bitcoin developer-guide*. URL: https://btcinformation.org/en/developer-guide#peer-discovery. (accessed: 01.07.2020).

[6] Blockstream.info. *Recent Transactions and blocks*. URL: https : / / blockstream.info/tx/recent. (accessed: 01.07.2020).

[7] Christian Decker and Roger Wattenhofer. "Information propagation in the Bitcoin network". In: *IEEE P2P 2013 Proceedings*. 2013, pp. 1–10.

[8] Varun Deshpande, Hakim Badis, and Laurent George. "BTCmap: Mapping Bitcoin Peer-to-Peer Network Topology". In: *2018 IFIP/IEEE International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN)*. 2018, pp. 1–6.

[9] Joan Antoni Donet Donet, Cristina Pérez-Solà, and Jordi Herrera-Joancomartí. "The Bitcoin P2P Network". In: *Financial Cryptography and Data Security*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 87–102.

[10] Jean-Philippe Eisenbarth, Thibault Cholez, and Olivier Perrin. "A Comprehensive Study of the Bitcoin P2P Network". In: *2021 3rd Conference on Blockchain Research Applications for Innovative Networks and Services (BRAINS)*. 2021, pp. 105–112.

[11] Meryam Essaid, Sejin Park, and Hongteak Ju. "Visualising Bitcoin's Dynamic P2P Network Topoogy and Performance". In: *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*. 2019, pp. 141–145.

[12] Explorer. *Blockchain Explorer*. URL: https://www.blockchain.com/explorer. (accessed: 01.07.2020).

[13] Btc Block Explorere. *Block Explorer*. URL: https : / / btc . com/. (accessed: 01.07.2020).

[14] Muntadher Fadhil, Gareth Owen, and Mo Adda. "Bitcoin Network Measurements for Simulation Validation and Parameterisation". In: May 2016.

[15] Muntadher Fadhil, Gareth Owenson, and Mo Adda. "Locality based approach to improve propagation delay on the Bitcoin peer-to-peer network". In: *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. 2017, pp. 556–559.

[16] Befekadu G. Gebraselase, Bjarne E. Helvik, and Yuming Jiang. "An Analysis of Transaction Handling in Bitcoin". In: *2021 IEEE International Conference on Smart Data Services (SMDS)*. 2021, pp. 162–172.

[17] Befekadu G. Gebraselase, Bjarne E. Helvik, and Yuming Jiang. "Transaction Characteristics of Bitcoin". In: *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 2021, pp. 544–550.

[18] Johannes Göbel et al. "Bitcoin Blockchain Dynamics: The Selfish-Mine Strategy in the Presence of Propagation Delay". In: *Performance Evaluation* 104 (May 2015).

[19] Stephen Hemminger et al. "Network emulation with NetEm". In: *Linux conf au*. Vol. 5. Citeseer. 2005, p. 2005.

[20] Muhammad Imran, Abas Md Said, and Halabi Hasbullah. "A survey of simulators, emulators and testbeds for wireless sensor networks". In: *2010 International Symposium on Information Technology*. Vol. 2. 2010, pp. 897–902.

[21] Audrius Jurgelionis et al. "An Empirical Study of NetEm Network Emulation Functionalities". In: *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*. 2011, pp. 1–6.

[22] Hidehiro Kanemitsu and Hidenori Nakazato. "KadRTT: Routing with network proximity and uniform ID arrangement in Kademlia". In: *2021 IFIP Networking Conference (IFIP Networking)*. 2021, pp. 1–6.

[23] Aeri Kim et al. "Analysis of Compact Block Propagation Delay in Bitcoin Network". In: *2021 22nd Asia-Pacific Network Operations and Management Symposium (APNOMS)*. 2021, pp. 313–318.

[24] Quan-Lin Li, Jing-Yu Ma, and Yan-Xia Chang. "Blockchain Queue Theory". In: *Computational Data and Social Networks*. Ed. by Xuemin Chen et al. Cham: Springer International Publishing, 2018, pp. 25–40.

[25] Andrew K. Miller et al. "Discovering Bitcoin ' s Public Topology and Influential Nodes". In: 2015.

[26] D.L. Mills. "Internet time synchronization: the network time protocol". In: *IEEE Transactions on Communications* 39.10 (1991), pp. 1482–1493.

[27] Jelena Mišić, Vojislav Misic, and Xiaolin Chang. "On the Benefits of Compact Blocks in Bitcoin". In: Feb. 2020.

[28] Jelena Mišić et al. "Modeling of Bitcoin's Blockchain Delivery Network". In: *IEEE Transactions on Network Science and Engineering* 7.3 (2020), pp. 1368–1381. DOI: 10.1109/TNSE.2019.2928716.

[29] Saeideh G. Motlagh, Jelena Mišić, and Vojislav B. Mišić. "Impact of Node Churn in the Bitcoin Network". In: *IEEE Transactions on Network Science and Engineering* 7.3 (2020), pp. 2104–2113.

[30] Satoshi Nakamoto. "Bitcoin: A peer-to-peer electronic cash system". In: *Decentralized Business Review* (2008), p. 21260.

[31] Till Neudecker, Philipp Andelfinger, and Hannes Hartenstein. "Timing Analysis for Inferring the Topology of the Bitcoin Peer-to-Peer Network". In: *2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress*. 2016, pp. 358–367.

[32] Till Neudecker and Hannes Hartenstein. "Short Paper: An Empirical Analysis of Blockchain Forks in Bitcoin". In: Sept. 2019, pp. 84–92.

[33] Kai Otsuki, Ryohei Banno, and Kazuyuki Shudo. "Quantitatively Analyzing Relay Networks in Bitcoin". In: *2020 IEEE International Conference on Blockchain (Blockchain)*. 2020, pp. 214–220.

[34] Kai Otsuki et al. "Effects of a Simple Relay Network on the Bitcoin Network". In: Aug. 2019, pp. 41–46.

[35] N D Patel, B M Mehtre, and Rajeev Wankar. "Simulators, Emulators, and Test-beds for Internet of Things: A Comparison". In: *2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. 2019, pp. 139–145.

[36] Carsten Rieck. "An Approach to Primary NTP by Using the LINUX Kernel". In: *2007 IEEE International Frequency Control Symposium Joint with the 21st European Frequency and Time Forum*. 2007, pp. 873–876.

[37] Hirotsugu Seike, Yasukazu Aoki, and Noboru Koshizuka. "Fork Rate-Based Analysis of the Longest Chain Growth Time Interval of a PoW Blockchain". In: *2019 IEEE International Conference on Blockchain (Blockchain)*. 2019, pp. 253–260.

[38] Yahya Shahsavari, Kaiwen Zhang, and Chamseddine Talhi. "Performance Modeling and Analysis of the Bitcoin Inventory Protocol". In: Apr. 2019. DOI: 10.1109/DAPPCON.2019.00019.

[39] Yahya Shahsavari, Kaiwen Zhang, and Chamseddine Talhi. "A Theoretical Model for Block Propagation Analysis in Bitcoin Network". In: *IEEE Transactions on Engineering Management* (2020), pp. 1–18.

[40] Jun.Kawahara Shoji.Kasahara. "Effect of Bitcoin fee on transaction-confirmation process". In: *Journal of Industrial and Management Optimization* 15.1547 (2019), p. 365.

[41] Amool Sudhan and Manisha J Nene. "Peer Selection Techniques for Enhanced Transaction Propagation in Bitcoin Peer-to-Peer Network". In: *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*. 2018, pp. 679–684.

[42] Taotao Wang et al. "Ethna: Analyzing the Underlying Peer-to-Peer Network of Ethereum Blockchain". In: *IEEE Transactions on Network Science and Engineering* 8.3 (2021), pp. 2131–2146.

[43] *What is VPP?* https://wiki.fd.io/view/VPP/What_is_VPP?. Accessed: 2021-06-10.

Befekadu Gezaheng Gebraselase received a B.Sc. degree in computer science from Addis Ababa University, Ethiopia, in 2012, an M.Sc. degree in computer science from the University of Milan, Italy, in 2018, and currently pursuing a Ph.D. degree in computer science and technology from the Norwegian University of Science and Technology, Trondheim, Norway, since 2018. From 2012 to 2016, he worked with Abyssinia bank, Ethiopia, as a Senior network engineer. He has published seven scientific conferences and journal papers. His research interests include Blockchains, vehicular network, fog computing, UAV network, IoT, and 5G/6G wireless network, and machine learning.

Bjarne E. Helvik (Life Senior Member, IEEE) received his Siv.ing. degree (MSc in technology) and the Dr. Techn. degree from the Norwegian Institute of Technology (NTH), Trondheim, Norway, in 1975 and 1982, respectively. He has since 1997 been Professor at the Norwegian University of Science and Technology (NTNU), the Department of Telematics and Department of information Security and Communication Technology, since 2022 as Emeritus. In the period 2009 – 2017, he has been Vice Dean with responsibility for research at the Faculty of Information Technology and Electrical Engineering at NTNU. He has previously held various positions at ELAB and SINTEF Telecom and Informatics. In the period 1988-1997 he was appointed as Adjunct Professor at the Department of Computer Engineering and Telematics at NTH. During 2003 – 2012 Principal investigator at the Norwegian Centre of Excellence Q2S - the Centre for Quantifiable Quality of Service and was in 2020 – 2021 Principal investigator at the Centre for Research based Innovation NORCICS - Norwegian Center for Cybersecurity in Critical Sectors. His field of interests includes QoS, dependability modelling, measurements, analysis and simulation, fault-tolerant computing systems and survivable networks, as well as related system architectural issues. His current research is on ensuring dependability in services provided by multi-domain, virtualised ICT systems, with activities focusing on 5G++ and SmartGrids.

Yuming Jiang (Senior Member, IEEE) received the B.Sc. degree from Peking University and the Ph.D. degree from the National University of Singapore. From 1996 to 1997, he was with Motorola, Beijing, China, and the Institute for Infocomm Research (I2R), Singapore, from 2001 to 2003. He has been a Professor with the Norwegian University of Science and Technology, Trondheim, Norway, since 2005. He has authored the book entitled Stochastic Network Calculus. His research interests are the provision, analysis, and management of quality of service guarantees. He was a Co-Chair of IEEE Globecom 2005—General Conference Symposium, a TPC Co-Chair of 67th IEEE Vehicular Technology Conference (VTC) 2008, the General Chair of IFIP Networking 2014 Conference, the Chair of the 2018 International Workshop on Network Calculus and Applications, and a TPC Co-Chair of the 32nd International Teletraffic Congress (ITC32) in 2020.

Paper E:

Befekadu G. Gebraselase, Bjarne Emil Helvik, and Yuming Jiang. 2019. Suitability of Blockchains to Enable and Support Networking Functions: State of Art. In Proceedings of the 2019 4th International Conference on Cloud Computing and Internet of Things (CCIOT 2019). Association for Computing Machinery, New York, NY, USA, 110–119. https://doi.org/10.1145/3361821.3361838

Suitability of Blockchains to Enable and Support Networking Functions: State of Art

Befekadu G. Gebraselase
NTNU – Norwegian Science and Technology
befekadu.gebraselase@ntnu.no

Bjarne Emil Helvik
NTNU – Norwegian Science and Technology
bjarne@ntnu.no

Yuming Jiang
NTNU – Norwegian Science and Technology
yuming.jiang@ntnu.no

ABSTRACT

The underlying network infrastructure faces challenges from addressing maintenance, security, performance, and scalability to make the network more reliable and stable. Software-defined networking, blockchain, and network function virtualization were proposed and realized to address such issues in both academic and industry wise. This paper analyzes and summarizes works from implementing different categories of blockchains as an element or enabler of network functions to resolve the limitation. *Blockchain as a network function* has been proposed to give support to the underlying network infrastructure to provide services that have less lag, are more cost-effective, have better performance, guarantee security between participating parties, and protect the privacy of the users. This paper provides a review of recent work that makes use of blockchain to address such networking related challenges and the possible setbacks in the proposal.

Keywords

Blockchain; Performance; Security; Network Function

CCS Concepts

• **Networks~Peer-to-peer networks** • **Networks~Mobile ad hoc networks** • **Security and Privacy ~ Mobile and wireless security**

1. INTRODUCTION

Blockchain [26] is a distributed ledger technology that allows information to be distributed. It enables the data not to be centralized or controlled by a single party. Blockchain allows the involved parties to communicate and exchange in a peer-to-peer (P2P) fashion through which distributed decisions are performed by the majority rather than by a centralized authority, [29]. As the word expresses, blockchain is a chain of blocks (records). Each block has a pointer to the previous block (previous hash), nonce, and transaction list, as illustrated in Figure 1. Having the cryptographic hash of the last block makes it hard to temper or reverse the current transaction. Blockchain has been explored/exploited in a variety of fields of studies, such *as a network function in networking*, to build new medical information platforms in medicine, and for money transfer in business, identity management in security, and voting systems in social science. In networking, all current connectionless networks require network-unique addresses, and in all known systems the uniqueness is enforced by some centralized entity, e.g., the IEEE sells MAC addresses, Internet Assigned Numbers Authority (IANA) and the Regional Internet Registry (RIRs) allocate IP addresses, ICANN and the TLDs provide URLs. These entities control related activities using a centralized way. With the current progress in the number of nodes that a network supports and the number of new organizations that emerge, centralized control will reduce the flexibility and quality of service delivered to the users and may become dictatorial since all the control power is from some specific entity. Besides, to add a new network service, we often end up purchasing a dedicated network element that satisfies the service specifications. To remove this dependency between network functions and hardware proprietary vendors, innovative technologies have been proposed. They include software-defined networking, network function virtualization, and blockchain.

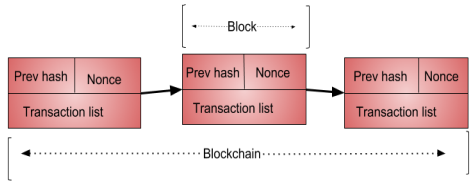


Figure 1. Typical structure of blockchain

Blockchains have properties that could change how the current network infrastructure works: As a distributed ledger, a blockchain can hardly be modified or controlled by a single or group of people or organizations; additionally, while it removes the intermediatory between parties, it still can guarantee trust between participating nodes. These promising properties can be applied to network functions or services that are currently provided only by trusted third-party brokers or using inefficient distributed approaches, which are found in network control, management and security services including AAA (authentication, authorization, and accounting), confidentiality, privacy, integrity, and provenance. In the literature, several such blockchain applications have emerged, where blockchains are exploited to enable, support, or enhance the desired network functions or services. In this paper, we classify and summarize them.

The main contributions of this paper are:

- We review the state of the art of blockchains acting as a network function and possible setbacks.
- We presented different architectures based on literature reviews and more insights.
- We presented nine different areas where blockchain claimed to solve a problem and possible limitation.

- We explore the applicability of blockchains bringing robust and reliable network infrastructures.

The remainder of this article organized as follows, Section 2 gives a short introduction and discusses optional consensus protocols. After that, Section 3 explains blockchains as an element of network functions to guarantee security between participating parties in different use cases: a cognitive cellular network, mobile communication, and 5G. Section 4 provided blockchain as an element of network function to enhance network control and management in various instances: wireless mesh network, Internet of things, and roadside traffic support. Section 5 addresses blockchain as an element of network function to increase the security in network protocols: named data networking and border gateway protocol. Finally, Section 6 gathers the main conclusions obtained and make a future proposal.

2. BLOCKCHAINS IN BRIEF

This section gives a brief introduction to some aspects of blockchains that are necessary for the remainder of the paper. Readers familiar with these may skip the section.

2.1 Blockchain Catagories

There are three types of blockchains public, private, and consortium. The division of the classification of blockchains only relies based on their characteristics. In public blockchain, the infrastructure is available for any users or nodes to join the network. The participating nodes need to download the records to take part in transaction or mining. The public availability of technology attracted popularity and accessibility. The flexibility and convenience of technology face significant challenges from scalability, latency, and performance. For instance, we can consider Bitcoin, one of the great electronic currency transaction. The total value of the currency up-to-date is close to USD 156 billion. Bitcoin faces challenges in increasing throughput capacity. The number of transactions supported by Bitcoin is not good enough to consider it in demanding networking.

Private blockchains are other kinds of blockchains controlled by private organization or communities. In such cases, the main challenges like performance, and latency are not the primary factors as in public blockchains. Access control in private blockchains implemented in different ways. It can be an independent authorizing system, or a set of rules to meet before joining. In private blockchains, it is easy to manage the consensus and membership services since all the nodes in the network are well known. Such alignments enable private blockchain owners or communities to plug and play functions. These properties make private blockchains more suitable for developing applications for many purposes. Developing different forms for different use allows for enhancing pure and easy access.

Consortium blockchain provides almost similar benefits as private blockchains. The main difference lies in performing validation of the transactions. In private blockchain, a single organization or company will be responsible for deciding which node can join the network. Additionally, what kind of pre-requirement must meet by the node. However, Consortium blockchains have a group of nodes or leaders that will decide for the whole network. These make it suitable for collaboration between different company or organizations. These also add enhancing security features of the public blockchain and allowing for more control over the network. The most common consortium blockchains examples are Quorum, Hyperledger, and Corda.

2.2 Consensus Protocols

In blockchains technology, nodes need to connect in peer to peer fashion and update all the modifications. If the updates are adding new records or amendments, then all the participating nodes receive the notification. Even though different organization implemented their version of consensus algorithms, the primary goal of consensus algorithms is to provide nodes to communicate and to offer validated set to add to the ledger. The most common consensus algorithms are Proof of Work, Proof of stake, Deligated proof of stake, Practical Byzantine fault tolerance, and Ripple.

2.2.1 Proof of work (POW)

POW is one of the consensus algorithms used by public blockchains like Bitcoin and Ethereum. Proof of work leads node with high resources use and computation power a more chance to solve a mathematical puzzle. By doing so, the node will earn some extra benefits. This method has exploited for 51 percent attack [25]. Relying on the computation power of nodes brings limitation on power consumption and resource use. Additionally, as the number of participating nodes rises scalability and latency increases together. Because of such significant limitations, most researchers are pointing out that practical Byzantine fault tolerance has better resource use, as illustrated Table I.

2.2.2 Practical Byzantine fault tolerance (PBFT)

PBFT is a consensus algorithm inspired by majority voting. The primary objective is reaching in consensus between distributed nodes with or without the presence of malicious nodes that sends wrong information. All the nodes communicate to one another heavily to guarantee the transaction is not falsify and to come up an agreement through majority voting. This technique could be a useful a consensus protocol when the number of nodes is small but if the number of participating devices increases then it will be hard to reach a consensus since all the nodes should talk and update every time. Additionally, it could be easily attacked by a single entity with a vast number of nodes.

Table I. CONSENSUS ALGORITHMS COMPARISON

Cases	**POW [4] [26]**	**PBFT [4] [9] [25]**	**POS [4] [35] [40]**	**DPOS [4] [26] [40]**	**Ripple [4] [25] [26] [40]**
Limitations	Energy consumption	Scalability	Unbalanced distribution	decentralization for speed and scalability	Highly Centralized
Energy Efficient	No	Yes	Yes	Yes	Yes
Permission	No	No	No	Yes	No
Adversary Tolerance	25%	33%	Unknown	Less than 20%	20%
Transaction Per Seconds	7-10	10-20	7	unknown	1500

Even if there are some limitations, these techniques still considered in different kind of blockchains, e.g., Hyperledger [9].

2.2.3 Proof of Stake (POS)

POS is also a consensus algorithm like POW. These algorithms designed to overcome the disadvantages of POW high energy consumption, as showed in table I row two. This algorithm is more deterministic in ways that the node that supposes to make the mining is the one which holds more wealth or stake. Although proof of stake developed to replace POW, this method has more limitations. For instance, a node can create a transaction that it can reverse later, the more wealth hold, the more chance to earn more. A node can create a secret channel for cheating. To remove these limitations of vulnerability and the richer get more prosperous concept new consensus protocol emerged: delegated proof of stake. The most known blockchain applications to use POS method are Peercoin, blackcoin, and NXT [35].

2.2.4 Delegated proof of stake (DPOS)

DPOS is a similar consensus algorithm like POS. This method adds flexibility by including delegates. The delegates take part in choosing the block size, transactions fee, and the amount of payment the witness should pay. Each stakeholder has the right to take part in voting for witness but allowed to vote only once for a witness at a time. The group of witnesses will be responsible for generating and adding a transaction to the blockchains. They earn rewards for their effort. The most significant enhancement from proof of work is a reduction in energy consumption. However, since the current underlying network infrastructure will not allow too many validators to take part, achieving the devolution will be a difficult task. Although, such limitation did not stop this algorithm from used by BitShare, Nano, Lisk, and more [23].

2.2.5 Ripple protocol consensus algorithms (RPCA)

RPCA is a method implemented outside of using blockchain technology [23]. The primary goal of the algorithms is to reach consensus between the participating entities. It helps to maintain the correctness and agreement of the network. Once consensus achieved, the current ledger considered "closed" and becomes the last-closed ledger. This method got much criticism because of most of the coin close to 61% are already mined and controlled by Ripple Lab. It is centralized [23], and the developers have more control over when and how many coins should be released or not.

3. BLOCKCHAINS TO ENABLE NETWORK-BASED SECURITY SERVICES

The amount of traffic generated by social networking takes second place after video streaming. The increasing demand in cellular network forces the development of higher radio spectrum. It will enable dynamic spectrum access that leads users to seek secondary access to many carriers. To make the access enabled more personal data must share with carries. The standard protocol AAA has limitations to protect the privacy of the users. The shared information needs protection from authorized, unauthenticated, and unauthenticated access. Moreover, the performance of the AAA protocol affected by network hops, latency, and jitter. Khashayar et al. [19] proposed an algorithm that uses a public blockchain to allocates the spectrum. They managed to use virtual currency as a payment mechanism. By including a public blockchain, they offered a fair opportunity for primary users to take part in service verification. Raju et al. [31] implemented private blockchains in cellular cognitive networks. By doing so, they manage to identify and measure the credibility of the user.

A good example is Hyperledger that uses PFTB to reach consensus between participating parties. From in Table I, this protocol cannot reach all the available nodes if they distributed. Leverage the property of the private blockchain could be a good option.

Table II. BLOCKCHAIN APPLICATIONS

Cases	**VN [7] [27]**	**CCN [19] [22]**	**MC [20] [28]**	**BGP [2] [31]**	**WMN [1] [32]**	**5G [10] [20]**	**IOT [3] [29] [39]**
Scope	Incident propagation, Transactions	Identity Management	Route Announcements, Transactions	Route Announcements	Route Announcements, Database	Route Announcements, Transactions	Software updates, supply-chain transactions
Addressed Issues	Security, Performance, Scalability	Privacy, Performance	Privacy, Performance	Security	Privacy, Performance	Performance, Security	Scalability, Performance
Unaddressed Issues	Time critical	Scalability	Time critical	Scalability, Performance	Security, portability	Scalability, Performance, Latency	Latency
Implementation	Ethereum	Private Blockchain	Ethereum	BGPcoin	Bitcoin	Private Blockchain	Slock.It, Filecoin
Testbed	Simulation	Simulation	Simulation	Simulation	Simulation, Live system	Simulation	Simulation
Limitations	Power Consumption, Maintenance, Latency, Security	Power Consumption, Maintenance	Power Consumption, Maintenance, Scalability, Monitoring/ Controlling	Performance, Maintenance, Latency	Network congestions, spectrum limitation, bandwidth consumption	Maintenance cost, latency	Power Consumption, resource utilization, transparency

3.1 Authorization of Mobile Communication Services (MC)

The service level agreement prepared by the service providers is unfair and undistributed. These service-level objects developed to increase the benefit of the company. It will make the amount of payout per individuals to become similar while the number of service usages is varying. Most of our day to day activities involve using different cellular network technologies like Wifi, WiMax, and 5G. The service level agreement provided by the providers does not consider per usage rather per income. The current service level agreement mechanisms lack clarity, integrity, visibility, and maintainability. Kiyomoto et al. [28] proposed blockchain-based authorization architecture to separate communication services from billing services. The architecture has central gateway servers. Blockchains are used to make authentication and authorizations. They suggested that users can see the service level agreement and change it according to consumptions. These will enable users or customers to trust and use the available infrastructure. There is always a tradeoff when realizing a new technology. Most of the information transmitted in mobile communication is time-critical and urgent. The currently available blockchain technologies are not suitable to play in a time-critical application. The main reason is consensus protocols tasks bring latency to the system. For instance, implementing a public blockchain will bring latency in the communications channels. While considering private blockchain will bring scalability issues. Moreover, the end layer devices will be forced to take part in tasks related to mining. The main factors we should consider installing blockchains in mobile communication are power consumption, resource use, maintenance, monitoring/controlling, and latencies.

3.2 5G: Blockchain-based Trusted Authentication

Starting 3G network architecture divided into a baseband unit and remote radio unit. The division gives more flexibility to the core network to control and manage route exchange between sub-networks. But this also put too much load to the core network to control the security issues of all sub-network. Soon, 5G will take over the cellular network. All the connected terminals will generate a massive amount of traffic to the core network. Blockchains can provide security through the cryptographic hash. Yanling et al. [38] propose a software-defined networking solutions. The software-defend networking controller manages traffics generated. It forwards flows tables routes to the authentication and database server. They realized a Dijkstra algorithm to calculate many routes for different media types. The authentication and database server connected to the centralized SDN controller. These may bring to a single point of failure and miss-configuration. The Dijkstra algorithm to calculate many routes has a limitation on resource use. It is a greedy approach that looks for the best routes through blind search. In such cases, implementing blockchain technologies will give more support to the technique by providing bookkeeping of all the possible route paths as a table of reference. Yang et al. [10] proposed blockchain-based trust authentication (BTA) architecture in C-RoFN. This architecture enables to authenticate network access with the user. It also helps to authenticate the network operator in the access area. By using the architecture, it is possible to reduce network connection cost and enhance the radio frequency. The proposal removes unified authentication in a core network and brings decentralized agreements. The virtualized edged layer take part as a middleware to process requests in data pre-processing, as illustrated in Figure 2. It also involves in aggregation, security, and privacy enhancement using blockchain technology.

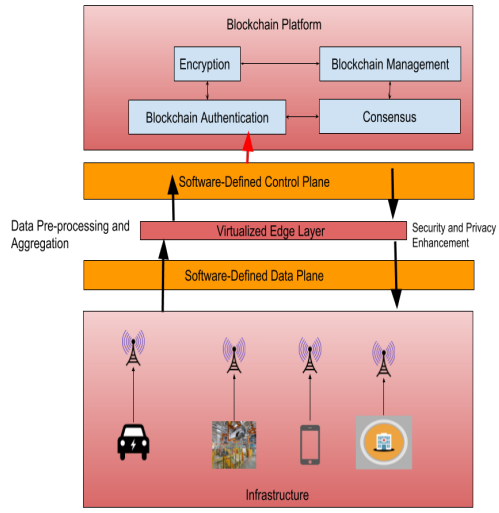


Figure 2. Blockchain-based Trusted Authentication (Edge-Enabled)

3.3 5G: Blockchain-based Network Slice

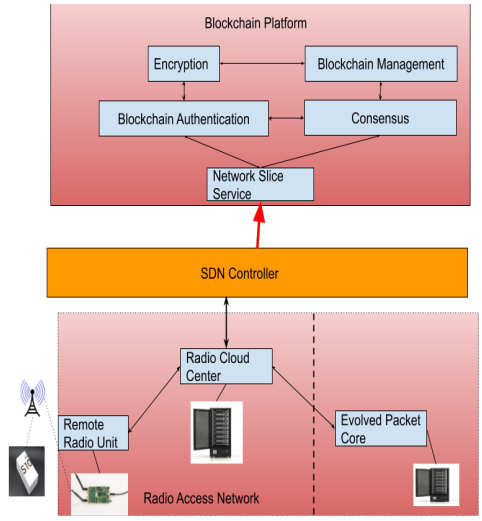


Figure 3. Blockchain-based Slice orchestration

Network slicing is one of the significant enablers in 5G services. It enables the facility to compose logical networks over shared physical infrastructures [13]. 5G network architecture is being defined by 3GPP to support connectivity and service deployments. These properties allow the service to include network function virtualization and software-defined networking. Different network vendors and researchers proposed slicing architecture that utilizes NFV and SDN. The main disadvantages of previous architectures are not aware of how to provide slice isolation and sharing. In this work, we are summarising proposals that use blockchains as a use case to realize network slicing. Blockchains have significant properties guaranteeing security between participating parties. Figure 3 shows one of the architectures proposed to enhance network slice security in 5G. Jere et al. [38] proposed blockchain-based slice leasing ledger to reduce service creation time. They offered an architecture that enables manufacturers to get slice more efficient ways. They utilize NFV and SDN for 5G network slicing and smart contract for Slice Leasing Ledger.

For services like 5G, the infrastructure requires fast access and high performance. The service needs to be fast enough to deliver end to end transport within 1-5 milliseconds. The availability of the infrastructure must provide low downtimes close to 5 minutes per year. The maintenance cost will be high since the blockchain handles most of the security part. Based on the current blockchains standards from private to the public are very far from reaching such specifications.

Table III. TYPES OF SYSTEMS

Cases	Centralized	Decentralized	Distributed
Pros	Easy to develop and maintain	High availability	No intermediator
Cons	Single point of Failure	Difficult to maintenance	Difficult to achieve consensus
Example	Microsoft passport	Blockchain	Multiplayer online game

3.4 Blockchain-based Security for Software-defined Networking

Network function is a defined functional block of network infrastructure. The infrastructure has a well-defined interface and essential behavior. Some of network function examples are routing, switching, and network broking monitoring. To add a new network service, we end up purchasing a dedicated network element that satisfies the service specifications. To remove the dependency between network function and vendor proprietary different techniques are proposed. These methods include network function virtualization and software-defined networking. Software-defined networking gives more flexibility by splitting the control plane and data plane. The control plane is responsible for handling routing and security tasks but also makes it more vulnerable to security. To address this limitation, blockchain proposed to give support to the control plane. The integrated blockchain enables security through record keeping and distributed ledger [30]. The control planes will have the same copy of records and security log entries that control the bridge in one controller will not affect the rest.

The main limitation on such realization is that the control plane tasked to perform blockchain jobs. Other than this, the main restriction comes from realizing and integrating two technologies. The main contribution of software-defined networking relies on the internet of things. In such low power devices integrating two technologies requires resource consumption reduces performance. Besides to all these facts, software define-networking has more problem by itself like software aging. Adding blockchain technologies in software-defined networking must realize the domain of implementation.

3.5 Routing Payment on the Lightning Network

Blockchain is one of the technologies that transform the current way of the transactions without a third-party. There are some challenges to solve first, like scalability, performance, and latency. For instance, in bitcoin, the main problem arises with the block size. If the block size increased from 1Mb to 32Mb then loses the idea of decentralization. The full nodes will hold a vast amount of the transaction, plus all the request will be redirected to the full node that contains the longest chain. However, if the size remains the same, then the scalability still will be an open problem to address. To solve this challenge, payment routing on the lightning network proposed [7]. The lightning network is an overlay network between peer to peer communication in the underlying network infrastructure. This protocol works on the top of the blockchain. This method has similar properties like off-chain blockchain [8] [16]. Off-chains is one of the methods implemented by the blockchain community to handle transaction between two parties. It makes a transaction until a certain amount satisfied and agreed. Jourenko et al. [8] summarized all the off-chain transaction taxonomies. They presented the necessary components that play a significant role in the scalability of cryptocurrency. These include a mechanizes to create a network on the opened channel and way of managing the network. By using a lightning network, for two nodes in the network to exchange message, they open a channel to transfer the transaction like off-chain blockchain. However, the main difference arises when the number of nodes to communicate is more than two nodes. In such cases, the sending node sends the message through the intermediate node by appending a secret password. Since all the participating parties must gain a benefit so that they proposed routing payment on the lightning network, for instance, let say the peer connection between A and B, B to C established. However, we want to make the transaction between A to C, and now we can put away of payment to each node forwarding the packet to a different station. When A makes a forward from itself to B charges, some payment and B to C charges some Payment as Well. In the end, the receiver Node C uses the password to decrypt the payment and complete the transaction by closing the channel. By using the Lightning network, the performance of the network increase. The scalability also achieved without the consideration of the block size and bandwidth capacity.

Implementing a virtual network over the blockchain could bring better performance, but, losing a token between parties could deliver inconsistency and unreliable services. Additionally, Un honest party can try to cheat by not forwarding the service or could change the timestamp of the current transaction [15]. Finally, a malicious user or node can learn the pattern of the communication and figured out the encryption keys.

4. BLOCKCHAINS TO IMPROVE NETWORK CONTROL AND MANAGEMENT

4.1 Wireless Mesh Network

Developing a network that utilizes fewer expenses and provides community services are essential. These idea or concept suggested by researchers and companies for the last two decades. Developing such an environment requires more financial support. It also requires a system that generates the transaction, and a controller that manages the network traffic usage of each region. One of the best ways to have such community services is to develop a wireless mesh network. The network will have a property to divide the nodes based on geo-location. Each different subnetwork need to find a way to trust each other's activities. These activities include transactions, the number of nodes they support, and the amount of traffic generated or used. A wireless mesh network considered as one of the future community services. It will connect many sub-networks to achieve cost reduction. However, building trust between different sub-networks could be a difficult task. Including blockchain, implementation could bring confidence between the participating sub-networks. AKabbinale et al. [1] Proposed blockchain for economically sustainable wireless mesh networks. There is one of the works offered to enable complete transparency and accountability for investment and revenue. It also helps other forms of economic enjoy sharing of network traffic, content, and services. The system keeps the records of each sub-networks on deployed blockchain technologies. Blockchain technologies deployed on the access network layer. These enable blockchains to have enough information about several links, nodes, maintenance, and consumption of the network resources. This information helps in determining the amount of budget, implementation of resource use, and to keep records of transactions.

The main disadvantages of integrating blockchains in wireless mesh-networks are network congestion. In a wireless mesh network, the amount of system generated by the subnetworks increases by the number of nodes. These will cause spectrum limitation, bandwidth and CPU consumption, and lack of interoperability [32]. In such implementation, wireless gateway routers will have more responsibility to perform. These duties include adding a transaction, mining blocks, and time-stamping everything. It could cause traffic congestion and overload. For instance, if the battery of one of the gateway routers failed or shutdown, this creates an overhead to the network. Additionally, users' data travels through different wireless hops that cause privacy concern. With high capital and maintenance capacity, such deployment will remove intermediary providers — for instance, Wi-fi, phone carriers, and middleman between organization.

4.2 Internet of Things

Internet of things is the network of Interconnected devices. The connection of devices either in heterogeneous or homogeneous environment faces some challenges. These challenges include transparency, audibility, conflict of identity, and forks [5]. Several ways of managing devices considered by companies centralized, decentralized, and distributed. The comparison between centralized, decentralized, and distributed systems are illustrated in Table III. It is very challenging to reach all the available nodes from the centralized system [3]. The main difficulties can be a failure from the server-side or client-side; either way, it is hard to manage it centrally. However, decentralized means of controlling the devices face performance issues. By including blockchains and smart contract [18], these issues can be removed. Then, activating the smart-contract to update the firmware any time it detects the latest version. Blockchain technologies included as middleware between the network and application layer. By using blockchain, it is possible to control and manage devices inside in the same ecosystem. Given that IoT devices are connected fully or partial, they are susceptible to an attacker. It is vital to secure update patches and communications. Blockchains, on the other hand, bring security through public key stored in the blockchain platform and private key stored in IoT devices [12].

Blockchains implementations in the internet of things are maintaining the balance as a middleman. They provide services between the network and the application layer. As proposed in Figure 4, these technologies should consider the capacities of end layer devices. The devices supported by the internet of things throughput capacity is different in each layer. The leading development of blockchain technologies is to act as a distributed ledger. This ledger includes security through a cryptographic hash of previous blocks. It will also result in a low susceptibility to manipulation and forgery by malicious participants [21]. Implementing such technology in monitoring and managing the network traffic need research works from academic and industry. Most works of blockchain in the internet of things are acting as a middleware between application and network layer. These tasks include hiding heterogeneity of hardware, operating systems, and protocols [34].

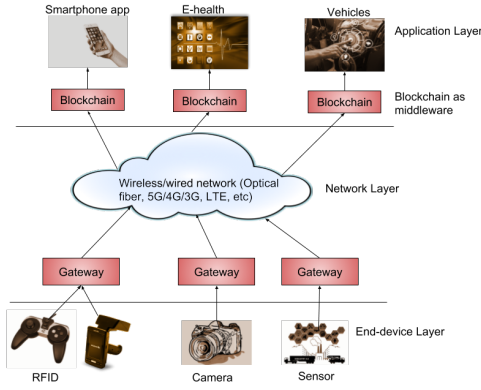


Figure 4. Three-layer of internet of things architecture with blockchain as middleware

Adding such a job in blockchain application over low power and computation devices is not adequate. Blockchains in the internet of things take part in providing uniform, and high-level interfaces, reusable, and portable applications. These requirements need a set of standard services that cut duplication of efforts. Merging all the properties of middleware into either public or private

blockchain must consider the deployment environments. Implementation of blockchains in the Internet of things faces challenges from storage, computation capacity. As illustrated in Figure 3, the end-device layer comprises sensors and low-power embedded platforms. In such devices, blockchain demands synchronization between participating devices. These will need enough bandwidth and computations power, which is very hard to guarantee. In low-power devices, the size of the memory is close to the 10kb and storage capacity of 100kb, but blockchain platforms demand GBs of memory. Other than this, the heterogeneity of devices also plays critical impacts on the performance. Because all the tools not manufactured to perform computation power, it is challenging to integrate devices.

In this part, we only tried to address a high-level limitation of considering blockchain in the internet of things since the consideration of blockchain in this area is increasing so we believe an independent work would satisfy the reader.

4.3 Road Traffic Support

Vehicle to vehicles communication reduces traffic incidents, jams, and pathways blocks. These properties improve day to day activities. However, making the information exchange between vehicle to vehicles raises privacy and security. The communication between cars managed by the centralized system. This system not only has a weakness of vulnerability for a single point of failure. However, it has scalability and performance limitation on reaching all the vehicles that are on mobility. It is also difficult to support a different kind of vehicular networks. Blockchains are considered to remove such challenges. Some of the contributions include trust management in Vehicle to vehicle communication. Smart Vehicles communication and cars to charge stations connection.

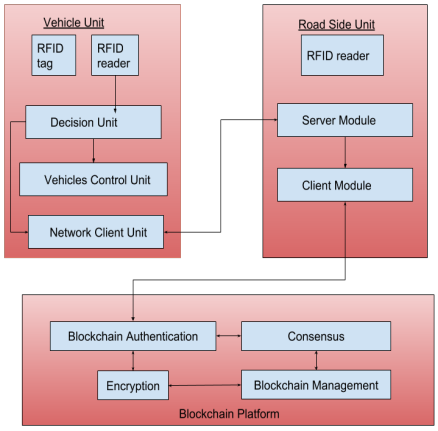


Figure 5. Road Traffic Support relationship between vehicles to roadside unit

4.3.1 Trust Management in Vehicles Network

In the centralized trust management system, the cars and roadside unit (RSU) connected to the central server. The central server provides the rating information. Based on the current progress, it calculates and stores trust values of vehicles to vehicles communication. The centralized system faces a high amount of request and high latency. The main factors the number of intelligent vehicles are increasing very fast.

In contrast, a decentralized system can cope up with the growth rate of intelligent vehicles; however, since the collected information stored on RSU, which will make it less consistent and incomplete. Yang et al. [39] proposed private blockchain-based decentralized trust management system. It provides a distributed ledger that is hard to temper (malicious vehicles could easily be discarded from the system). The RSU is responsible for collecting rating information and trust value management. The cars, at the same time, will manage traffic-related events. They send a warning message to other vehicles. The communication protocol can be using vehicles to vehicles communication standards (Long-term Evolution Vehicles to Vehicles (LTE-V2V), Dedicated short-range Communication (DSRC), and Vehicle Ad Hoc network (VANET)).

The main setbacks of the previous approaches come from RSU server and decision unit of vehicles. As shown in Figure 5, the RSU server module becomes overloaded by collecting traffic-related events. The decision unit of the cars will be responsible for sending warning messages and guaranteeing the arrival. Additionally, the decision unit of the vehicle will be to overload in the case a large amount of traffic propagation. The processing capacity of the server module and decision unit must be considered as a critical point in the implementation.

4.3.2 Communication of Smart Vehicles

Vehicle Ad Hoc network (VANET) is one of the vehicles to a vehicle's communication standard [27]. The transferring or sending a message to another vehicle requires identifying the source, plate number, and the identity of the owner. On such occasions, vehicles owner loses interest to broadcast any incidents, jams or congestions. To solve privacy and motivation to publicize the different event mechanisms proposed: Threshold Authentication [14], Credit Network with Blockchain, and a privacy-preserving blockchain based incentive announcement network. Threshold Authentication, if the number of vehicles that confirmed the message is higher than the threshold value, then the message considered honest and valid. This methodology has two limitations if the number of malicious nodes or vehicles are higher than the number of correct nodes then the news tempered, the privacy of the event broadcaster including the owner identity is not secured. In Credit network with blockchain, each node has a point regarding their reputation so that to find the dishonest node is very easy. The only disadvantage is that it is simple to trace the coins through the public key. However, it is tough to trace the transactions that make it less reliable.

However, L. Li [22] designed privacy-preserving vehicles announcement protocol on a blockchain network. It maintains the reliability and anonymity of the messages. To increase the motives of the users have accounts at different addresses. Where they collect the coin, they gained for providing or announcing the events to the neighbors who need it.

4.3.3 Electric Vehicles and Charging Stations

The international energy agency forecasted the number of electric vehicles would reach 125 million by 2030 and increased by 57 percent in 2017. The domination of electric cars also brought another attention, which is charging stations. The main problem arises from the amount of time the vehicles owners need to recharge. If electric cars want to reload from the charge station, the owner needs to reload more often than oil gas, which means the owner needs to pay for the transaction each time. Such

demands make the Bitcoin community to developed the off-chain transaction. In the off-chain operation, the transaction fee is charged only to open and close the channel. To even reduce the cost of the transaction bitcoin-based payment network proposed. E Erdin et al. [6] created a virtual topology payment channel network. They managed to cut the transaction fee by allowing vehicles to recharge from any one of the stations. The stations are connected to the virtual interface on the top of the blockchain. The channel is open between the two participating parties may bring inconsistency. To overcome the limitation, the new method evolved. The lightning network is an overlay network between peer to peer communication. It works on the top of the blockchain and has similar properties like off-chain blockchain. For two nodes in the system to exchange message, they open a channel to transfer the transaction like off-chain blockchain. However, the main difference arises when the number of nodes to communicate is more than two nodes. In such cases, the sending node sends the message through the intermediate node by appending a secret password that is well known by the receiver since all the participating parties must gain a benefit so that they proposed routing payment on the lightning network.

The implementation of blockchains in roadside traffic management brings advantages to transport management. It will reduce traffic jams, incidents, and enhance road management system. The deployment of blockchains should consider power consumption, latency, maintenance, and security. The RSU server and client modules take part in performing blockchains tasks. The tasks to mining block, time-stamping, and adding transactions. Such responsibilities help the road management system but add more overload to RSU unit. For example, in the case of the public blockchain, the consensus protocol is POW, which consumes too much power. The number of transactions per seconds is less than the amount of throughput needed by a traffic management system. Besides that, by implementing Etherium or Bitcoin on the roadside traffic management system will affect the cost of maintenance. The battery life of RSU, and network instability, while the removal of the RSU unit that holds the longest chain, another limitation to be addressed before implementations.

5. BLOCKCHAINS TO ENHANCE SECURITY IN NETWORK PROTOCOL

5.1 Border Gateway Protocol

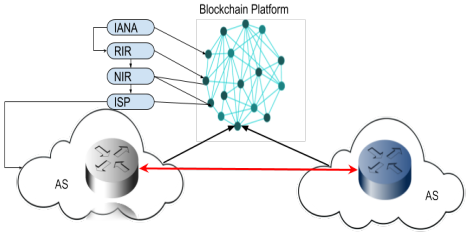


Figure 6. Architecture of BGP with blockchain resource management.

BGP is the only implemented protocol to exchange routing information between two different autonomous systems. It designed without the intention of possible attackers like prefix/sub prefix hijacks. The Internet Engineering Task Force (IETF) Secure inter-domain routing (SIDR) proposed Resource Public Key Infrastructure (RPKI). It works on centralized authorities. This system has a high chance of miss-configuration and compromised RPKI authorities. So, other techniques proposed by the researchers on how to handle such limitation on RPKI. Appending the transparency log to alarm the changes on RPKI and adding inanimate objects to realize the revocation. However, even adding such a method will not guarantee if the malicious authorities delete or modify objects. Besides that, to respond to such activities based on the alarm system takes time.

Moreover, revocation in RPKI requires complicated collaboration with resource certificate issuers. Jun et al. [17] present an Expectation Exchange and Enforcement mechanism. It defines policies between autonomous systems such that any independent system may enforce such policies. Kumar and Crowcroft [2] proposed security of distance-vector related routing protocol through digital signature. They performed loop detection in pathfinding to verify the selected route's path information is correct. Xing, Q. Wang, B. Wang [31] propose public blockchain-based internet number resource authority and BGP security, which implemented a blockchain application that provides temper-resilence and transparent internet routing registry plus origin repository and governance infrastructure for BGP security. Additionally, they developed a lightweight framework on blockchain to replace RPKI authentication based on origin.

Blockchains considerations in BGP has enhanced the security of prefix and subfix hijacks. The implementation of blockchains in BGP needs to consider performance and latency. Blockchains are designed to acts as a distributed ledger between participating parties IANA, RIR, ISP, and NIR as shown in Figure 6. In such deployment, security must be given a higher priority since a single break can cause global attacks. For instance, based on the current state of the art, public blockchains have route announcement capacity of 10 to 20, which is less throughput for BGP.

Furthermore, public blockchains like Bitcoin ability to generate new route blocks takes 10 minutes that is not enough. If we prefer to put in place either private or consortium blockchains, the autonomous node gets responsible for management. These tasks include organizing, access control, and resource management. The current BGP protocols take 30 minutes to propagate new routes, but if blockchains considered, then it may go beyond. Adding a new independent system needs to download whole records in which cases may take weeks or more depending on bandwidth.

5.2 Named Data Networking

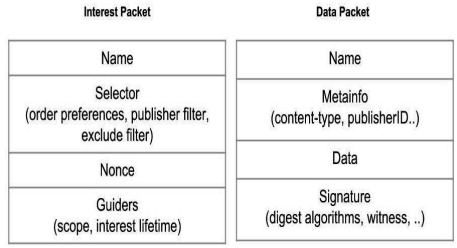


Figure 7. Named Data Networking architecture

The Internet is as the massive interconnection of computers or nodes. The information exchange between participating nodes is done using TCP/IP. It provides reliable and delivery guarantee services. Although some would agree that TCP/IP has some limitation on securing network flow, it has no particular way to broadcast messages to some specified group. So, implementing blockchain on it considered not adequate. Mohammad et al. [24] proposed policy-based security module in TCP/IP stack and policies include security policy in the application layer. Security control and data security layer in the transport layer. Hao-yu et al. [34] addressed the issues related to TCP/IP agreement has some specific security bugs. They analyzed the limitation of the protocol in performing an agreement. The parameters are unreliable identity authentication, information divulging not prevented, and weak protection against data integrity. The proposed possible counter solutions. Jin et al. [33] suggested another protocol to exchange message between different parties on the Internet Named Data Networking (NDN) [33]. It is different from the defacto protocol TCP/IP in architecture and concept-wise. NDN architecture has two communication unit: interest packet and the data packet, as illustrated in Figure 7. Interest packet is the named representative or description string of the data packet, the relationship between the interest packet and data packet is always one to one. If a requester wants to get the data packet needs to send the request by presenting the interest packet. As each of the packet transition can be traced and no IP addressing, then malicious nodes or system could be easily traced out.

NDN has two essential properties: it works on the content of the data and allows multicasting. This system behavior makes it more appropriate for data transmission. Jin et al. [33] proposed a bitcoin blockchain decentralized system over NDN. For a new node to take part in the network needs to download the whole record. While the miner continues working on the current transaction. After that, the miners broadcast the new update to the system. In the end, the listeners check the correctness of the block then update their local blockchain up to date. Since the nodes in NDN can send the message to a collection of groups at a time will increase the performance of the network. Some researchers suggest that implementing blockchain technology in the current internet protocol TCP/IP is wrong decisions even so deploying it on Named data networking (NDN) could bring better performance and provide less latency service [33]. However, changing TCP/IP prefixes to named URLs will take a considerable amount of times. Besides that, NDN has unsolved problems: how to manage naming, routing, security, and application development.

6. CONCLUSION

The main aims of including blockchains in the networking infrastructure are to enhance security, to increase performance, to reduce latency, and to build trust between participating parties. In this paper, we presented the nine different areas where blockchain claimed to solve the challenges. Blockchains are making an impact in various domains, including networking. Table II demonstrates the contribution of blockchain as an element of network function in networking. As we can see from Table II, most of the contributions are to guarantee security and performance. Although there are contributions to support the performance of the network environment, most of them lack considering the limitations. The development phase requires the considerations of power consumption by the miners. The time it takes for the miners to finish and propagate the update is also another factor. The underlying network infrastructure complexity differs in different conditions and environment. Performing a test case only in simulation and modeling reduces the contributions. The proposed architecture and deployment in mobile and cellular network lack more research works. In mobile and cellular network, the main challenges come from resource use. The nodes in the mobile and cellular network have small capacity comparing to what needed in the blockchain. Finally, time-critical matters are vital in networking. So, in such an environment considering blockchain to provide network function delay the services. Besides, if the development of the lightweight framework that considers all the limitation introduced by Table II, then realizing blockchains as a network function brings more advantages to the current network infrastructure.

7. REFERENCES

[1] Aniruddh Rao Kabbinale, Emmanouil Dimogerontakis, Mennan Selimi, Anwaar Ali, Leandro Navarro, Arjuna Sathiaseelan, "Blockchain for Economically Sustainable Wireless Mesh Networks," 2018.

[2] B. Kumar, and J. Crowcroft. "Integrating Security in Inter-Domain Routing Protocols" ACM Computer Commun. Review, pages 36 51, 1993.

[3] Boudguiga Aymen, Bouzerna Nabil, Granboulan Louis, Olivereau Alexis, Quesnel Flavien, Roger Anthony, and Sirdey Renaud, "Towards Better Availability and Accountability for IoT Updates by means of a Blockchain," 2017

[4] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei and C. Qijun, "A review on consensus algorithm of blockchain," 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, 2017, pp. 2567-2572.

[5] Das, Manik Lal, "Privacy and Security Challenges in Internet of Things," Distributed Computing and Internet Technology, 2015.

[6] E Erdin, M Cebe, K Akkaya, S Solak, E Bulut, S Uluagac, "Building a Private Bitcoin-based Payment Network among Electric Vehicles and Charging Stations," 2018

[7] Giovanni Di Stasi, Stefano Alallone, Roberto Canonico, Giorgio Ventre, Routing Payments on the Lightning Network, Napoli, 2018.

[8] Gudgeon, Lewis et al. "SoK: Off The Chain Transactions," IACR Cryptology ePrint Archive 2019

[9] H. Sukhwani, J. M. Martínez, X. Chang, K. S. Trivedi, and A. Rindos, "Performance Modeling of PBFT Consensus Process for Permissioned Blockchain Network (Hyperledger Fabric)," 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS), Hong Kong, 2017, pp. 253-255.

[10] H. Yang, H. Zheng, J. Zhang, Y. Wu, Y. Lee and Y. Ji, "Blockchain-based trusted authentication in cloud radio over fiber network for 5G," 2017 16th International Conference on Optical Communications and Networks (ICOCN), Wuzhen, 2017, pp. 1-3

[11] Ho, G., Leung, D., Mishra, P., Hosseini, A., Song, D. and Wagner, D., "Smart locks: Lessons for securing commodity internet of things devices.," in Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security

[12] Huh Seyoung, Cho Sangrae, and Kim Soohyung, "Managing IoT devices using blockchain platform," 2017

[13] J. Backman, S. Yrjölä, K. Valtanen, and O. Mämmelä, "Blockchain network slice broker in 5G: Slice leasing in a factory of the future use case," 2017 Internet of Things Business Models, Users, and Networks, Copenhagen, 2017, pp. 1-8

[14] J. Shao, R. Lu, X. Lin and C. Zuo, "New threshold anonymous authentication for VANETs," 2015 IEEE/CIC International Conference on Communications in China (ICCC), Shenzhen, 2015, pp. 1-6.

[15] Jordi Herrera-Joancomartí et al. “On the Difficulty of Hiding the Balanceof Lightning Network Channels,” IACR ePrint Archive 2019

[16] Jourenko Maxim, Larangeira Mario, Kurazumi Kanta, and Tanaka Keisuke, "SoK: A Taxonomy for Layer-2 Scalability Related Protocols for Cryptocurrencies," 2019

[17] Jun Li, J. Stein, Mingwei Zhang, and O. Maennel, "An expectation-based approach to policy-based security of the Border Gateway Protocol," 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), San Francisco, CA, 2016, pp. 340-345.

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

[19] K. Kotobi and Sven G. Bilén, "Blockchain-enabled spectrum access in cognitive radio networks," 2017 Wireless Telecommunications Symposium (WTS), Chicago, IL, 2017, pp. 1-6.

[20] K. Valtanen, J. Backman and S. Yrjölä, "Creating value through blockchain powered resource configurations: Analysis of 5G network slice brokering case," 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Barcelona, 2018, pp. 185-190.

[21] Kshetri Nir, "Can Blockchain Strengthen the Internet of Things?," IT Professional, 2017

[22] L. Li et al., "CreditCoin: A Privacy-Preserving Blockchain-Based Incentive Announcement Network for Communications of Smart Vehicles," in IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 7, pp. 2204-2220, July 2018

[23] L. S. Sankar, M. Sindhu, and M. Sethumadhavan, "Survey of consensus protocols on blockchain applications," 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, 2017, pp. 1-5.

[24] M. Al-Jarrah and A. R. Tamimi, "A Thin Security Layer Protocol over IP Protocol on TCP/IP Suite for Security Enhancement," 2006 Innovations in Information Technology, Dubai, 2006, pp. 1-5.

[25] N. Bozic, G. Pujolle and S. Secci, "A tutorial on blockchain and applications to secure network control-plane," 2016 3rd Smart Cloud Networks & Systems (SCNS), Dubai, 2016, pp. 1-8.

[26] Omar Dib, Kei-Leo Brousmiche, Antoine Durand, Eric Thea, Elyes Ben Hamida, "Consortium Blockchains: Overview, Applications, and Challenges," International Conference on Wireless and Mobile Communications, ICWMC, Nice, June 2017.

[27] S. Harrabi, W. Chainbi, and K. Ghedira, "A multi-agent proactive routing protocol for Vehicular Ad-Hoc Networks," The 2014 International Symposium on Networks, Computers, and Communications, Hammamet, 2014, pp. 1-6.

[28] S. Kiyomoto, A. Basu, S. Rahman, and S. Ruj, "On blockchain-based authorization architecture for beyond-5G mobile services," 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), Cambridge, 2017, pp. 136-141.

[29] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.

[30] S. R. Basnet and S. Shakya, "BSS: Blockchain security over software defined network," 2017 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, 2017

[31] S. Raju, S. Boddepalli, S. Gampa, Q. Yan and J. S. Deogun, "Identity management using blockchain for cognitive cellular networks," 2017 IEEE International Conference on Communications (ICC), Paris, 2017, pp. 1-6.

[32] Selimi Mennan, Rao Aniruddh, Ali Anwaar, Navarro Leandro, Sathiaseelan Arjuna, "Towards Blockchain-enabled Wireless Mesh Networks," 2018

[33] Tong Jin, Xiang Zhang, Yirui Liu, Kai Lei, "BlockNDN: A bitcoin blockchain decentralized system over named data networking," 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 75-80, 2017

[34] W. Hao-yu, C. Hui-Zhi, Z. Xu, J. Chao-jun and J. Xiao-Juan, "The Security and Promotion Method of Transport Layer of TCP/IP Agreement," 2010 Second International Conference on Information Technology and Computer Science, Kiev, 2010, pp. 513-517.

[35] W. Y. Maung Maung Thin, N. Dong, G. Bai, and J. S. Dong, "Formal Analysis of a Proof-of-Stake Blockchain," 2018 23rd International Conference on Engineering of Complex Computer Systems (ICECCS), Melbourne, Australia, 2018, pp. 197-200.

[36] Wang Xu, Zha Xuan, Ni Wei, Liu Ren, Guo Y, Niu Xinxin, and Zheng Kangfeng, "Survey on Blockchain for Internet of Things," Computer Communications, 2019

[37] Xing, Q.; Wang, B.; Wang, X. "BGPcoin: Blockchain-Based Internet Number Resource Authority and BGP Security Solution," Symmetry 2018, 10, 408.

[38] Y. Zhao and X. Zhang, "New media identity authentication and traffic optimization in a 5G network," 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, 2017, pp. 1331-1334.

[39] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, Blockchain-based Decentralized Trust Management in Vehicular Networks, in IEEE Internet of Things Journal & 10.11.2018.

[40] Zheng, Zibin & Xie, Shaoan & Dai, Hong-Ning & Chen, Xiangping & Wang, Huaimin, "Blockchain challenges and opportunities: A survey" International Journal of Web and Grid Services, 2018.

Paper F:
B. G. Gebraselase, "Blockchain-Based Information Management for Network Slicing," 2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE), 2021, pp. 555-559, doi: 10.1109/ICCIKE51210.2021.9410755.

Blockchain-Based Information Management for Network Slicing

Befekadu G. Gebraselase
Department of Information Security and Communication Technology
NTNU, Norwegian University of Science and Technology, Trondheim, Norway
{befekadu.gebraselase}@ntnu.no

***Abstract*—Network slicing is the crucial enabler of the new emerging 5G and beyond network generations. It facilitates the facility to compose logical networks over shared physical infrastructures. From the implementation perspective of view, slice isolations and sharing become very challenging. It introduces challenges to provide secure information to the subscribers and enables users to modify and configure the registrations while following the service level agreement. To this aim, we introduce a blockchain as a service, in which the distributed ledger technologies provide security and accessibility to the end-users while removing a third-party involvement. Additionally, it allows tenants and subscribers to manage the slice information as necessary without violating the agreement. The primary advantage of including blockchain in architecture is using it to slice isolation and sharing.**

***Index Terms*—Network slicing, blockchain, slice isolation and sharing, information management**

I. Introduction

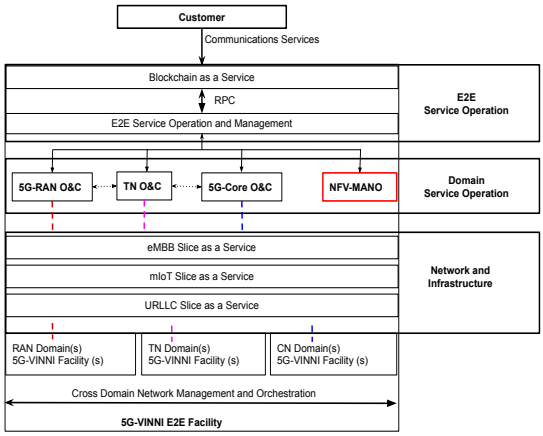


Fig. 1. 5G network slice architecture [19]

The fifth network (5G) generation system is expected to provide complex use cases and requirements. Some of these services' key performance indexes are higher throughput, low-latency, ultra-high reliability, higher connectivity density, improved energy consumption, greater capacity, and higher mobility range [2]. Such requirements make it more desirable enhancement relative to previous versions, while to achieve such broad potentials, the 5G depends on the logical partitions of the shared network infrastructure while ensuring the various applications from interfacing each other [14]. Slice isolation and sharing starting from the tenants/users level up to the infrastructure level while sharing common key elements like network functions, network slice instances, and virtual links [16]. To achieve slice isolation and sharing, we can leverage technologies that are evolving: blockchain [3], software-defined-networking (SDN) [5], network function virtualization (NFV) [8], and cloud computing [14] may help to some extent.

The Europian Telecommunication Institute (ETSI) has developed a network function virtualization manager and orchestrator (NFV-MANO) for life cycle management (LCM) of network function virtualization infrastructure (NFVI) and virtual network functions (VNF) or virtual links (VL) for a network service/slice. The current ETSI NVF-MANO framework aims to provide multi-tenant multi-network services while the network services management and control will be logically centralized. The ETSI NVF-MANO specifications define the functionality and building block of each component but not how to deploy each component or how it can be realized in any use cases [7], which gives a wide range of options to leverage different technological advantages to secure tenant/users information. Blockchains have been used for such demands in different fields, including networking. Based on this fact, we are proposing blockchain as an information management component that can be integrated as a service [9] to address the requirements mentioned above by leveraging the network slicing architecture available from an EU H2020 project 5G-VINNII [19], as proposed in Fig. 1.

Services provided by a multitenancy system require to incorporates a secure channel between participating parties [7]. Most importantly, since the tenants share the same infrastructures, it must apply the data's privacy and confidentiality. Additionally, based on the 5G specifications, a tenant may have the right to lease the slice for other tenants/users, bringing security issues while one or more tenants can access others' data or resources [3]. Further, service providers may store tenant-related information in the same database for the sake of cost-reductions or improper management that may lead to removing tenant information that may also affect others [1]. These facts imply that we need an information management

platform that provides privacy, security, and confidentiality without any third party involvement. This is where blockchain comes, a distributed ledger technology that provides all the above-stated benefits.

II. Preliminary Network Slice and Blockchain Concept

A. Network slice

The shared network infrastructure makes isolations of the logical partitions a must fulfilled requirement. Ensuring the network slice isolation from interfacing with each other incorporates performance, resiliency, security, privacy, and management level isolations. A network slice consists of three logical layers: the service instance layer (SIL), where end-users service supported, network slice instance layer (NSI), which provides network function for the upper services and resource layer (RL) that provides virtual or logical required resources for two upper layers. Fig. 2 illustrates the overview of conceptual slice management, where the blockchain is used as information management. In each slice, we can see some client blockchain application instances in which they cooperate to provide a report, monitor, validate, and verify integrity and authenticity.

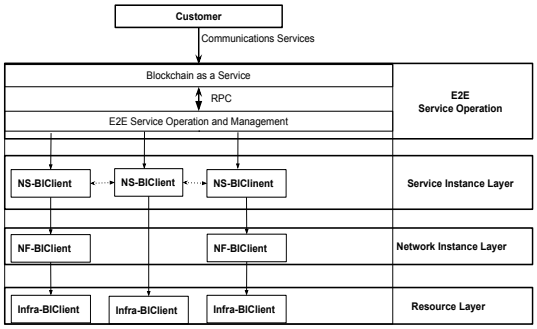


Fig. 2. Overview of conceptual slice management [12]

1) Blockchain: Blockchain is a distributed ledger technology that enables parties to exchange information/transactions while removing a third-party by making majority based decisions. With the help of consensus protocols, it provides anonymous, autonomous, privacy-protected, and secured communications between exchanging parties [6]. These properties attracted attention from various domains to include blockchains in 5G network slicing. For instance, some recent works include blockchain as a network slice broker [11] and blockchain as a network slice broker and leasing [3]. This article introduced blockchain as a service that enables the 5G system to interact with the tenant/users in a more accessible but secure way.

Fig. 2 illustrates the blockchain-based information management architecture to achieve isolation among network slice instances at the management level in which the blockchain provides multi-tenancy support. The blockchain is a service [9] that can be deployed in the cloud, virtual machine, or any physical machine in which it has a virtual or direct link to the end-to-end service operation and management. As such, the ledger provides tenants and users access to manage the slice provided. Blockchain-as-a-service (BaaS) can leverage cloud computing to provide the flexibility of plug and play and blockchain to provides security through consensus protocols and cryptography [15]. It allows users/tenants to leverage cloud-based solutions to build, host, and manage their slice information according to the agreement between the service provider and the tenants.

III. Blockchain-Based Slice Architecture

The blockchain-based information management concept is proposed as an extension on the ETSI NVF-MANO model [8], specifically inspired by the work EU H2020 project 5G-VINNII [19]. We extended the 5G-VINNII architectural proposal by introducing blockchain as a service to act as a middleware between the facility and the tenant by integrating a software-defined controller (SDC).

The blockchain provides the tenant with the ability to manage and orchestrate the slice provided. The ledger can be exposed to the tenant/users as an access point to configure the overall system's communication demands. Simultaneously, the blockchain uses a software-defined controller to collect and gather information from the end to end service operation manager. Fig. 1 and 2 shows the interconnections between the blockchain and service orchestrator and manager through either remote procedure call (RPC) or virtual link (VL).

Fig. 3 illustrates the conceptual architecture of blockchain-based information management, in which the two tenants were used to demonstrate how the deployment would bring slice isolation while sharing the infrastructure.

VIM is responsible for managing the NFVI per domain; depending on the tenants' layout, the number of VIMs' also changes. Each VIM is accountable for managing the independent NFVI, where it controls virtual resources' life cycles in its domain. For instance, Fig. 3(a) and 3(b) have an independent VIM, where each is responsible for keeping inventory of VMs associated with physical resources and performance and fault management of the resources. Additionally, it exposes physical and virtual resources to other management systems, like blockchain vis BClient APIs.

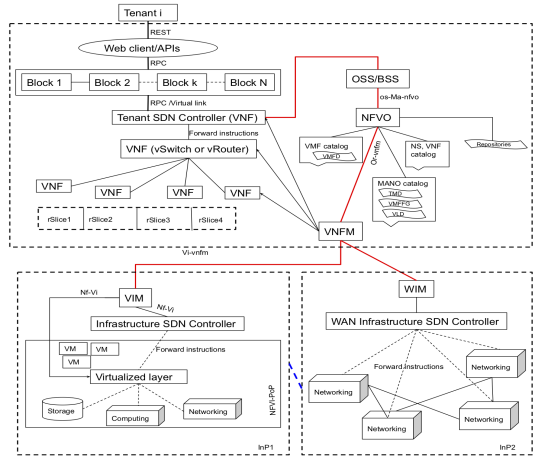
The VNFM manages VNFs, in which the relationship between the manager and the network functions can be 1-to-1 or 1-to-many. As such, it is responsible for creating, maintains, and terminates the virtual network functions. Like VIM, this functional unit is also responsible for the fault, configuration, accounting, performance, and security management of VNFs. Fig. 3(a) shows the VNFM used to control and manage the tenant-SDN controller, responsible for providing transparency between blockchain and tenants. Other than previously mentioned responsibilities, the tenant-SDN controller is also responsible for scales up/scales down VNFs based on the slice requirement, such as CPU usage.

NFVO is responsible for service and resource orchestration, which coordinates, authorizes, releases, and engages NFVI resources among different PoPs or within one PoP. For instance, the two tenants demonstrated in Fig. 3(a) and 3(b) have independent PoP where the NFVO manages the communication. WAN infrastructure manager (WIM) is responsible for enabling connectivity between NVFI-PoPs. Other than this, NFVO creates end-to-end service between different VNFs by coordinating with the respective VNFMs. To provide such functionality, it contains a repository that holds different information about NFV-MANO. A VNF Catalog is a repository of VNFDs, a deployment template that specifies a VNF in terms of its deployment and operational.

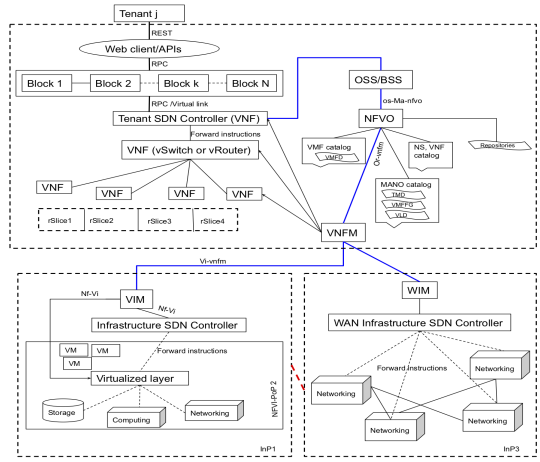
Additionally, NS Catalog holds valuable Network services, and a description of their connectivity through VLs is stored in NS Catalog for future use, and NFV Catalog contains details about NSIs and related VNF Instances. Further, the NFVI Catalog a repository of NFVI resources utilized to establish NFV services. Infrastructure SDN is responsible for managing and controlling the connectivity among the virtualization containers.

OSS/BSS is a collection of applications that a service provider uses to operate its business. The most common functionality includes service assurance, which enables tracking service performance and ensuring customer service-level agreements (SLA) are met [12], service catalog that provides the system administrators what product is offered to the customers and how much resources are dedicated. Additionally, service management is responsible for orchestrating the fulfillment process while keeping the customer-facing entity about progress, changes, or delivery issues.

The blockchain acts as a middleware that enables tenants/users to interact with the system securely. The ledger will validate the tenant/users' request. After that, it will be changed into a network slice request, which will be forwarded to the tenant SDN. The tenant SDN controller interacts with NFV-MANO and the infrastructure SDN directly or indirectly to provide the end-to-end slice. In similar cases, when the user's request is different from slice subscriptions. However, related information like checking the current state of access, subscription list, or usage amount, and more, the blockchain interacts with the two SDN controllers that the NFV-MANO manages to collect information about different technology domains information.



(a) Tenant i slice deployment



(b) Tenant j slice deployment

Fig. 3. Network Slicing deployment applying NFV and Blockchain concepts to achieve isolation

IV. Information Management

Blockchain synchronizes in a peer-to-peer fashion where each node is independent. It is a distributed ledger technology that enables participating parties to exchange securely and transparently. The consensus protocols and cryptographic hashing mechanisms will allow the ledger to make majority based decisions while removing the middleman, making it a technological advancement relative to the traditional database that brings decentralized-based decision making and distributed data control and storage. This property makes it more suitable for acquiring information, organize and develop data, and distribute data across parties.

The emerging 5G and beyond system generates massive data traffic, which requires mechanisms to manage the data. For instance, as the number of tenants increases, then slice per user also grows exponentially. It may generate a problem in proper data management since all the customers may use the same or similar resources. This is where blockchain becomes crucial; it gives the 5G and beyond system a platform that reduces the hurdles.

A. Acquire Information

Tenant/users interact with the 5G system through the ledger. The ledger performs a registration and identity management

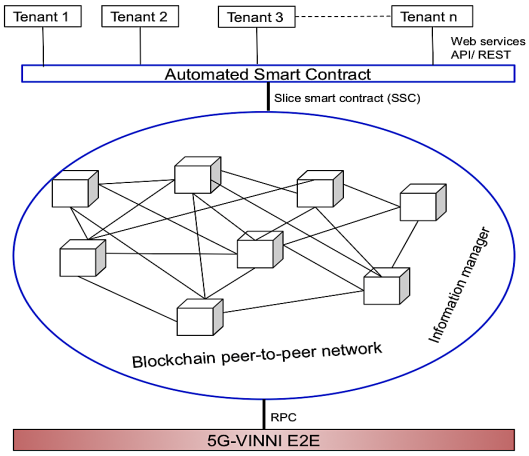


Fig. 4. Blockchain smart contract

[3] of the new or old customers. At the same time, when the new arrival requests are requesting a subslice/slice, then the ledger convert the slice smart contract (SSC) request into a slice request, as shown in Fig. 4. After validation by the blockchain node, this request is forwarded to the E2E service operation and manager for further operations. Finally, when the request reserved the resource need in all the technology domains, the ledger stores its information.

On the other hand, when customers use the ledger to get some details of the subscription, then, ledger provides the necessary details according to the predefined agreement. At the same time, such information also is recorded in a log file for future accountability. As a result, the ledger stores information about the agreement, slice information, access details, and subscribers' usage, which will make the full database detailed.

B. Organize and store

One of the critical challenges of multitenancy services is the privileged access may compromise other information while sharing the same storage area. Even though blockchain can provide permission-level access, it is crucial to separate and organize record-keeping to reduce the conflict between slice/subscribers. Blockchain stores transactions in a block, where blocks are linked through a cryptographic hash make it hard to rearrange the blocks accordingly. However, it is still possible to run multiple instances of the ledger to track information and store it in privileged-level, for instance, independent records for an administrative domain to control what is going on in the network.

C. Distribute information

The ledger provides novel trust mechanisms through consensus and cryptographic hash. New records propagate to the neighbors' nodes while the receiving ends validate the transaction/information insides are valid. In two different tenant cases, the ledger pushes records to the other tenant when information sharing is required, bringing better services.

D. Develop information

With the heterogeneity of use cases and the customers' requirements, the 5G network system requires extra data-analytics to cope with the day-to-day demand. With the ledger holding valuable information about the services and users' consumptions, it will be easier for system administrators and service providers to control service operation and management. For instance, InP can generate a report to see the utilization and consumption of the infrastructure. At the same time, MNO can also overview resource usage and area regions that require extra facilities to enhance the service.

V. Use-case

The 5G network is expected to support various communications services, such as eMBB, IoT, and URLLC. Fig. 1 illustrates these services provided as a service. This increasing flexibility of the networks to support services with diverse requirements may present operational and management challenges [18]. Therefore, an information management system can collect network data, including service, slicing, and network function related data. In a later case, this can be used to perform analytics on network performance and service assurance.

The amount of data and the information exchange's sensitivity between the customer and service provider require a high confidentiality level. Let take two customers that use the system for two different services, customer i and j as represented by Fig. 3(a) and 3(b). Customer i is interested in downloading a multiple-image download and uploading it from the service provider, an eMBB service. Based on our proposed architecture, the customer needs to verify the access and then validate the request to meet the service-level-agreement signed by the two parties via BaaS. Simultaneously, the ledger collects all the information related to throughput and bandwidth consumption by the current user, which can generate a report when the privileged access requires it. Additionally, the ledger also allows the current user based on the privilege to access the service's information.

On the other hand, let us assume Customer j gets virtual-based clinical treatment, where the sensors attached to the body and house send some sensitive information that requires service with low latency and high reliability, a URLLC. In such cases, besides providing low latency services, the system must provide the information's integrity and confidentiality. Similarly, customer j will follow the same step as customer i, except that the BaaS provides a more secure channel.

As the above examples explain, BaaS is not involved in providing network management rather information management. The tenant, service providers, network operator, and customers can have a privileged level of access to the information collected while providing the first security and information management level. Nevertheless, it provides isolation in terms

of security by enabling data protection, privacy, and accountability. A possible deployment architecture is provided in Fig. 5, which shows a possible architectural deployment.

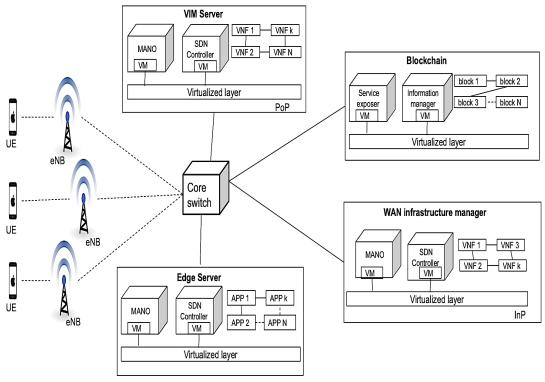


Fig. 5. Blockchain based information management architecture for 5G

VI. Discussion

The 5G network and beyond system shall allow the network operator to provide priority-based identification in the time of resource competition [18]. Nevertheless, at the time of writing this article, no work states how this can be done. Since blockchain is used as the validator's entry-level for slice creations, propriety-based resource utilization can be adjusted by providing privileged access. Additionally, policy management and control of service assurance are still undergoing works that require detailed inspections of each entity, in which the ledger may help to some extent by providing some information through Bclient in each layer of the slice. Information management of the 5G network slicing is growing slowly because of the dense requirement and use-cases; however, including blockchain may help.

VII. Conclusion

The blockchain-based information management for 5G network slicing conceptual architecture brings a new insight into managing the information while maintaining the information's confidentiality. The blockchain network provides access-level security, while the ledger maintains the bookkeeping. The proposed conceptual architectures provide privacy, confidentiality, and integrity of sensitive information. Additionally, the architectures removed the hustle to distribute sensitive information on a shared database while providing a privileged level of access, and it hides the interactions and dependencies between users.

In conclusion, we proposed a conceptual architecture while showing how each level can communicate the achieve the same goal. Blockchain is introduced to provide a safe and secure ledger that allows users to subscribe and utilize the services without external help; simultaneously, the service provider and network operators can monitor the activities. Such a conceptual proposal needs more work regarding security, performance, and dependability attributes that they may or may not bring to the system. We will consider the aforementioned fundamental attributes' impact on the 5G and beyond systems in future works.

References

[1] Wayne Brown, Vince Anderson, and Qing Tan. "Multitenancy - Security Risks and Countermeasures". In: Sept. 2012, pp. 7–13. ISBN: 978-1-4673-2331-4.

[2] Ibrahim Afolabi et al. "Towards 5G Network Slicing over Multiple-Domains". In: *IEICE Transactions on Communications* E100.B (May 2017).

[3] J. Backman et al. "Blockchain network slice broker in 5G: Slice leasing in factory of the future use case". In: *2017 Internet of Things Business Models, Users, and Networks*. 2017, pp. 1–8.

[4] Zbigniew Kotulski et al. "On end-to-end approach for slice isolation in 5G networks. Fundamental challenges". In: Sept. 2017, pp. 783–792.

[5] I. Afolabi et al. "Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions". In: *IEEE Communications Surveys Tutorials* 20.3 (2018), pp. 2429–2453.

[6] Qi Feng et al. "A survey on privacy protection in blockchain system". In: *Journal of Network and Computer Applications* 126 (Nov. 2018).

[7] A. J. Gonzalez et al. "Dependability of the NFV Orchestrator: State of the Art and Research Challenges". In: *IEEE Communications Surveys Tutorials* 20.4 (2018), pp. 3307–3329.

[8] GR NFV-EVE. "Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions". In: *IEEE Communications Surveys Tutorials* 20.3 (2018), pp. 2429–2453.

[9] J. Singh and J. D. Michels. "Blockchain as a Service (BaaS): Providers and Trust". In: *2018 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*. 2018, pp. 67–74.

[10] M. Garrich et al. "The Net2Plan-OpenStack Project: IT Resource Manager for Metropolitan SDN/NFV Ecosystems". In: *2019 Optical Fiber Communications Conference and Exhibition (OFC)*. 2019, pp. 1–3.

[11] B. Nour et al. "A Blockchain-Based Network Slice Broker for 5G Services". In: *IEEE Networking Letters* 1.3 (2019), pp. 99–102.

[12] M. Xie et al. "Towards Closed Loop 5G Service Assurance Architecture for Network Slices as a Service". In: *2019 European Conference on Networks and Communications (EuCNC)*. 2019, pp. 139–143.

[13] F. Z. Yousaf et al. "MANOaaS: A Multi-Tenant NFV MANO for 5G Network Slices". In: *IEEE Communications Magazine* 57.5 (2019), pp. 103–109.

[14] S. Zhang. "An Overview of Network Slicing for 5G". In: *IEEE Wireless Communications* 26.3 (2019), pp. 111–117.

[15] W. Zheng et al. "NutBaaS: A Blockchain-as-a-Service Platform". In: *IEEE Access* 7 (2019), pp. 134422–134433.

[16] Bjarne E. Helvik Andres J. Gonzalez Jose Ordonez-Lucena. "The Isolation Concept in the 5G Network Slicing". In: (May 2020).

[17] Yilma Girma Mamuye et al. "Benchmarking Open-Source NFV MANO Systems: OSM and ONAP". In: (Mar. 2020).

[18] 3GPP TS 28.530. "Management and orchestration; Concepts, use cases and requirements". In: *White papaer* (), pp. 1–32.

[19] 5GVINNI. *D1.2 Design of network slicing and supporting sub- systems v1*. URL: https://www.5g-vinni.eu. (accessed: 31.07.2020).

Part III
Not Included Paper

Paper G:
Befekadu G. Gebraselase, Charles Adrah, Tesfaye Zerihun, Bjarne E. Helvik, Poul Heegaard, "Blockchain Support For Time-Critical Self-Healing In Smart Distribution Grids," 2022 IEEE PES ISGT Europe, Serbia

Blockchain Support For Time-Critical Self-Healing In Smart Distribution Grids

Befekadu G. Gebraselase*, Charles M. Adrah*, Tesfaye Amare†, Bjarne E. Helvik* and Poul E. Heegaard*
*Department of Information Security and Communication Technology,
NTNU, Norwegian University of Science and Technology, Trondheim, Norway
†Sintef Energy AS, Trondheim, Norway
Email: {befekadu.gebraselase, charles.adrah, tesfaye.zerihun, bjarne, poul.heegaard}@ntnu.no*/@sintef.no†

***Abstract*—Smart distribution grids have new protection concepts known as fault self-healing whereby Intelligent Electronic Devices (IEDs) can automatically reconfigure the power circuits to isolate faults and restore power to the relevant sections. This is typically implemented with IEDs exchanging IEC 61850 Generic Object Oriented Substation Event (GOOSE) messages in a peer-to-peer communication network. However, a self-healing application may be faced by challenges of emerging cyber-physical security threats. These can result in disruption to the applications' operations thereby affecting the power system reliability. Blockchain is one technology that has been deployed in several applications to offer security and bookkeeping. In this paper, we propose a novel concept using blockchain as a second-tier security mechanism to support time-critical self-healing operations in smart distribution grids. We show through a simulation study the impact of our proposed architecture when compared with a normal self healing architecture. The results show that our proposed architecture can achieve significant savings in time spent in no-power state by portions of the grid during cyber-physical attacks.**

***Index Terms*—Smart Distribution Grid, Cybersecurity, Blockchain, IEC 61850, Self-healing**

I. Introduction

The transition from the traditional power grid to the smart grid has enabled more reliable, efficient, and secure services [1]. The traditional grid enables a unidirectional power flow from generation plants to the consumers, whiles the smart grid enables electricity and information exchange in both directions as well as the integration of distributed energy resources (DERs) [2]. The IEC 61850 standard for power utility automation defines the communication between Intelligent Electronic Devices (IEDs) within a substation as well as wide-area protection and control application services [3].

The fifth generation mobile network (5G) is defined over three types of connected services known as Enhanced mobile broadband (eMBB), Massive Machine Type Communication (mMTC), and Ultra-reliable low latency communications (URLLC) [4]. 5G URLLC services will support time-critical operations such as remote surgery, emergency response, autonomous driving, and smart grid with strict latency (1ms) and reliability (99.999%) requirements [5]. Hence, a 5G URLLC solution will be deployed for Smart Distribution Grid (SDG) applications bringing benefits of guaranteed quality of service, as well as reducing capital and operational expenses.

One such application is fault self-healing in SDGs. Fault self-healing refers to automatic control measures to eliminate or isolate the fault and restore service using modern communication, computer, automatic control and power electronics technologies [6]. Self healing, also known as Fault Location, Isolation, and Service Restoration (FLISR) is a key SDG application which is implemented using peer-to-peer (P2P) IEC 61850 Generic Object Oriented Substation Event (GOOSE) communication. FLISR enables utilities to significantly reduce outage time to the end customers and improve their distribution network reliability.

When GOOSE messages are used in such self-healing applications, communication between the IEDs is usually not encrypted due to performance reasons since the messages are time-critical [7]. This makes the information exchange between participating IEDs susceptible to man-in-the-middle attacks, denial-of-service (DoS), and repeat messages attacks [8]. Moreover, self-healing applications can involve multi-actors of producers, consumers and prosumers in the grid which may bring the challenge of trusting the information exchanges among the IEDs from these actors. Furthermore, there is no specification on how the messages exchanged between actors can be stored as immutable records and to be used in future investigations.

In this paper we propose a GOOSE and 5G based self-healing architecture utilizing blockchain to address the challenges of security and immutability of records. Our architecture uses blockchain as a second-tier security layer to validate time-critical messages in a smart grid FLISR application. As a second-tier security layer, our blockchain architecture does not affect the time-critical GOOSE message exchanges but can reverse actions of these time-critical messages when they are invalidated at some future time. The architecture also provides a secure decentralized bookkeeping that can be used to probe and track both internal and external actor activities.

The rest of the paper is organized as follows: Section II presents our proposed blockchain second-tier security archi-

This paper has been funded by Prodig - Power system protection and control in digital substations, (under KPN-project ENERGIX, 295034/E20), and CINELDI - Centre for intelligent electricity distribution, an 8 year Research Centre under the FME-scheme (Centre for Environment-friendly Energy Research, 257626/E20). The authors gratefully acknowledge the financial support from the Research Council of Norway, and the other partners of Prodig and CINELDI.

tecture for a self-healing application. In section III, we explain how blockchain information is organized and distributed in our architecture. In Section IV, we present a simulation evaluation of the proposed architecture. Finally, we give concluding remarks in Section V.

II. Blockchain-Based Second-Tier Security Architecture

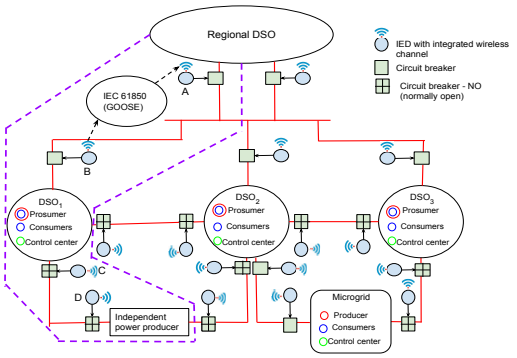


Fig. 1. Self-healing in a smart distribution grid

A. Smart Grid Use Case

Figure 1 presents a self-healing application in an SDG topology consisting of a Regional Distribution System Operator ($\mathtt{R-DSO}$), three Distribution System Operators ($\mathtt{DSO}_1$, $\mathtt{DSO}_2$ and $\mathtt{DSO}_3$), an Independent Power Producer (IPP) from a distributed energy resource, and a micro-grid. We assume the DSOs are independent with their own administrative domains. The IPP can be connected to $\mathtt{DSO}_1$ or $\mathtt{DSO}_2$ while the microgrid can be connected to the main grid through $\mathtt{DSO}_2$ or $\mathtt{DSO}_3$. All the feeder lines have circuit breakers and IEDs. The IEDs serve as control units for the circuit breakers in the feeder lines and are mainly used to localize fault outages.

The self-healing activities entail Fast fault clearing, Locate the fault, Isolation, Selectivity and Reconfiguration (FLISR). The IEDs use P2P communication to exchange GOOSE messages with the self-healing logic residing in the IED. FLISR operates autonomously without the need of a control centre. However, all actions taken during a self-healing carried out will be communicated immediately to the control centre which can be located at the $\mathtt{R-DSO}$, to keep the grid operation status up-to-date.

B. Architecture

Figure 2 shows the proposed architecture which combines the FLISR application and blockchain over a 5G communication system. 5G is introduced to provide the P2P communication mechanism among the interacting IEDs as well as to the control center. This can be realized by a virtual bus in 5G edge cloud. In this work, we considered that URLLC provides the network communication service. As such, the time-critical low latency requirement is guaranteed according to URLLC specifications [9]. We also assume a network slice that provides the URLLC service for the application traffic.

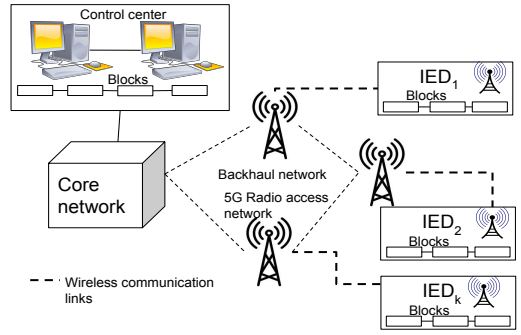


Fig. 2. Blockchain-based second-tier security architecture in self-healing smart distribution grid

In our proposal, the FLISR system still operates to autonomously isolate and restore faults as previously explained. However, whenever an event occurs such as a GOOSE message is published or is received by a subscribing IED, an independent corresponding blockchain transaction is also generated and broadcasted to the other IEDs in the network. This generated transaction propagates to a full node or miner to be validated or invalidated.

If an IED validates the transaction, it adds the transaction to its' valid log file. The self-healing action taken based on the GOOSE message from that now validated transaction is kept and maintained as true. Subsequently new blocks may be created from the valid transactions and added to the ledger for bookkeeping purposes. On the other hand, if an IED invalidates the transaction received, the self-healing action taken based on the GOOSE message from the invalidated transaction will then be discarded or reversed. The invalid transaction will be added and stored to its invalid log file.

The blockchain communication will run over the URLLC network slice, which will make the transaction propagation delays smaller than running on traditional networks. The block generation intensity depends on the event that leads to self-healing handling. The IED node generates a block of valid transactions inside when self-healing handling happens. Nevertheless, the block generation intensity can be adjusted to keep the bookkeeping with the GOOSE message delay requirement. When the IEDs create a block, they add transactions from the backlog to the block and push it to the neighbor nodes. We assume the IEDs to have computation and storage capacity to run blockchain nodes.

The block generation depends on the type of consensus protocol used. In this work, we consider Practical Byzantine Fault Tolerance (PBFT) which is a computationally-light consensus mechanism compared to other consensus protocols such as Proof-of-Work (PoW) and Proof-of-Stake (PoS) [10, 11]. CPU utilization of a node using PBFT was 20% and over 70% using

PoW [10]. In addition, PBFT can process more transactions in the order of 1000 transactions per second compared to PoW (2 transactions per second) and PoS (50 transactions per second) [10].

C. Interactions between self-healing and blockchain events

In this section, we demonstrate through sequence activities, the interactions between the self-healing events and the blockchain events in normal operation and when there is a malicious attack. We illustrate this using a simplified self-healing application involving $\mathtt{R-DSO}$, $\mathtt{DSO_1}$ and an IPP in Figure 1 (i.e., section is framed by violet line)

In the normal operation, $\mathtt{DSO_1}$ is fed power from the $\mathtt{R-DSO}$ (i.e., circuit breakers, $\mathtt{CB_A}$ and $\mathtt{CB_B}$ are closed). When a fault occurs on the feeder line between $\mathtt{R-DSO} \leftrightarrow \mathtt{DSO_1}$, the $\mathtt{IED_A}$ and $\mathtt{IED_B}$ communicate the event change (i.e., by publishing GOOSE messages) to $\mathtt{IED_C}$ and $\mathtt{IED_D}$. $\mathtt{CB_A}$ and $\mathtt{CB_B}$ become opened whiles $\mathtt{CB_C}$ and $\mathtt{CB_D}$ which are normally open will then close to allow power to be fed from the IPP. When the fault is cleared between $\mathtt{R-DSO} \leftrightarrow \mathtt{DSO_1}$, the event change is again communicated to $\mathtt{IED_C}$ and $\mathtt{IED_D}$ which then open $\mathtt{CB_C}$ and $\mathtt{CB_D}$. $\mathtt{CB_A}$ and $\mathtt{CB_B}$ also close and power feed is restored to $\mathtt{DSO_1}$ from the $\mathtt{R-DSO}$, as in the normal operation.

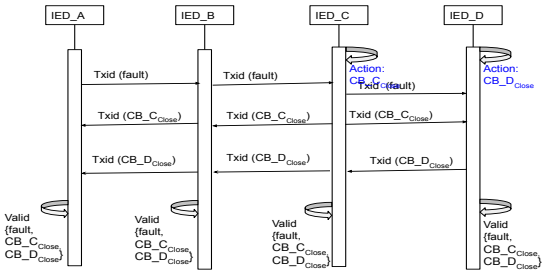


Fig. 3. Blockchain activity sequence for self-healing in normal operation

The activity sequence of self-healing with blockchain process is shown in Figure 3. At the instance when a fault occurs and $\mathtt{IED_A}$ and $\mathtt{IED_B}$ publish $\mathtt{GOOSE_{fault}}$ messages, blockchain transactions are also generated by $\mathtt{IED_A}$ and $\mathtt{IED_B}$. These transactions, named $\mathtt{Txid(GOOSE_{fault})}$, are broadcasted to all other IEDs to be validated. $\mathtt{IED_C}$ and $\mathtt{IED_C}$, will execute an action (.i.e, close $\mathtt{CB_C}$ and $\mathtt{CB_D}$), and will also at this instant generate blockchain transactions $\mathtt{Txid(CB-C_{Close})}$ and $\mathtt{Txid(CB-D_{Close})}$. Both transactions will reach all other IEDs in the network and be validated. The transactions having been validated as true will necessitate no further actions at this point. Note that similarly, the actions $\mathtt{CB-A_{Open}}$ and $\mathtt{CB-B_{Open}}$ can also generate blockchain transactions that can be independently validated.

In Figure 4, we show an activity sequence of self healing with blockchain process under malicious attack. Here, the $\mathtt{GOOSE_{fault}}$ message is published into the network from $\mathtt{IED_X}$, a malicious user that tries to compromise the information exchanges between the other IEDs. However, $\mathtt{IED_X}$ is not part of the blockchain network since unknown nodes can not join the private network without approval by the other nodes. $\mathtt{IED_C}$ and $\mathtt{IED_D}$ being subscribers to this message will immediately execute actions of closing their normally open circuit breakers (i.e., $\mathtt{CB-C_{Close}}$, $\mathtt{CB-D_{Close}}$) and also generate blockchain transactions based on the actions taken. $\mathtt{Txid(CB-C_{Close})}$ and $\mathtt{Txid(CB-D_{Close})}$ are broadcasted to be received by all other IEDs. These transactions will go through the validation process. $\mathtt{IED_C}$ and $\mathtt{IED_D}$ will invalidate these transactions, as will all other legitimate IEDs in the network. At this time, the previous actions executed by $\mathtt{IED_C}$ and $\mathtt{IED_D}$, $\mathtt{CB-C_{Close}}$ and $\mathtt{CB-D_{Close}}$, will be reversed based on the invalidated transactions.

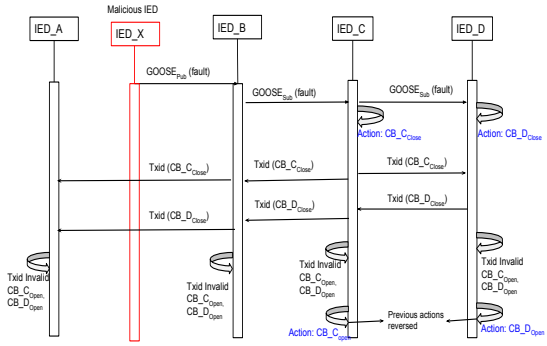


Fig. 4. Blockchain activity sequence for self-healing under malicious attack

III. Blockchain Information Organization in the Proposed Architecture

This section presents how blockchain collects transaction information from the GOOSE messages, organizes and stores this information, distributes the new updates to the neighbor devices, and generates a report for further investigation.

A. Acquire transaction information

GOOSE is usually sent as Layer 2 multicast and hence used within the substation (i.e., intra-substation). GOOSE can be routed into the wide area network using layer 2 tunneling or transport over layer-3 routers with UDP/IP headers [12]. Figure 5 shows a typical GOOSE packet frame. The variable portions is contained in the Application layer. The GOOSE Application Protocol Data Unit (APDU) has 12 unique fields that is used to organize its message.

When a change of event occurs and an IED publishes a GOOSE message, some fields (`gocbRef`, `goId`, `t`, `stNum`, `confRev`, `allDaTA`) in the GOOSE APDU can be changed into transaction inputs. A generated transaction by the IED will then contain these inputs together with either the Media Access Control (MAC) or Internet Protocol (IP) source and destination addresses for record keeping.

GOOSE messages are published spontaneously into the network when an event change occurs or periodically to repeat

```
Frame 1: 165 bytes on wire (1320 bits), 165 bytes captured (1320
bits)
Ethernet II, Src: SuperMic_3d:2e:9f (00:25:90:3d:2e:9f), Dst: Iec-
Tc57_01:28:50 (01:0c:cd:01:28:50)
GOOSE
   APPID: 0x0001 (1)
   Length: 151
   Reserved 1: 0x0000 (0)
   Reserved 2: 0x0000 (0)
goosePdu
    gocbRef: SERVER-GOOSELDevice1/LLN0$GO$CB_Goose_TRIP1
    timeAllowedtoLive (msec): 1000
    datSet: SERVER-GOOSELDevice1/LLN0$Goose_TRIP1
    goID: Goose_TRIP1
    t: May 13, 2016 13:30:28.228710949 UTC
    stNum: 2
    sqNum: 0
    test: False
    ConfRev: 1
    ndsCom: FALSE
    NumDataSetEntries: 2
allData
    bitString: BITS 0000-0015: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    boolean:  TRUE
```

Fig. 5. GOOSE packet after event

the same event state. However, to reduce the number of transactions generated by an IED, only transaction generation from specific events leading to self-healing action need to be added to the blockchain. The corresponding blockchain transaction input, which we call `Txid(GOOSE)`, that can be generated is shown in Figure 6.

```
Goose_to_Transaction
{
    "Txid": "2fef4b992c1f88e33b43647b98fccda8f5cc670exxxxx",
    "Src":  "SuperMic_3d:2e:9f (00:25:90:3d:2e:9f)",
    "Dst":   "Iec-Tc57_01:28:50 (01:0c:cd:01:28:50)",
    "size": 165,
    "gocbRef": SERVER-GOOSELDevice1/LLN0$GO$CB_Goose_TRIP1,
    "goID": Goose_TRIP1,
    "t": May 13, 2016 13:30:28.228710949 UTC,
    "stNum": 2,
    "ConfRev": 1,
    "allData": 0000-0015: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0,TRUE
}
```

Fig. 6. `Txid(GOOSE)`: a GOOSE packet translated into blockchain transaction

B. Organize, store and distribute transaction information

Each IED acts as an independent blockchain node that participates in adding and validating a block. A GOOSE message with an event change published into the network by an IED (i.e., `GOOSE(Pub)`), will instantly generate a new `Txid(GOOSE)` transaction into the network. Similarly, an IED that receives a subscribed GOOSE message (i.e., `GOOSE(Sub)`), and executes an action will also generate a new transaction, `Txid(Action)` into the network.

When new transactions arrive at an IED, the IED validates and then stores the new arrivals at a backlog until block creation occurs. At the same time, the IED also maintains an internal reference or mapping between the GOOSE message it has either published or subscribed to, and the transactions generated. `GOOSE(Pub)` $\leftrightarrow$ `Txid(GOOSE)` and `GOOSE(Pub)` $\leftrightarrow$ `Txid(Action)`). Hence, it is possible to reverse actions when transactions have been invalidated. Figure 7 shows the internal mapping in an IED between GOOSE frames (published/subscribed) and transactions generated which are stored in the backlog as valid or invalid.

The blocks are connected through cryptographic hash and stored according to a timestamp creation and confirmation order. This makes it easier to extract relevant information at any point in the network. Both GOOSE and blockchain rely on broadcast communications to publish or propagate the latest event updates, hence the subscribing or participating entities receive the new updates autonomously.

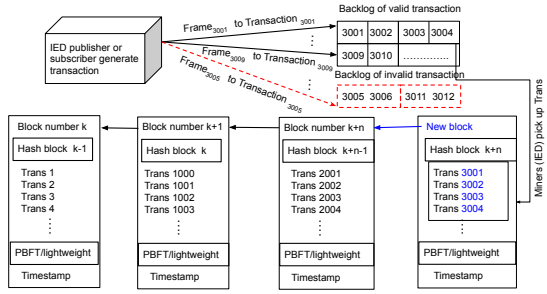


Fig. 7. Mapping in IED for GOOSE frames and blockchain transactions

IV. Simulation Study

In this section, we carry out a simulation study of the proposed architecture. The architecture is modeled with Stochastic activity network (SAN) models using the Möbius tool [13]. SAN is a general and modular stochastic modelling formalism, which is built from atomic block models. We use the simplified self-healing application in Figure 1 in our simulation study.

A. Model

We develop three atomic models consisting of the power system network, 5G communication system, and the blockchain transactions process. Firstly, a model is developed for the power system network made up of $\mathtt{R-DSO}$, $\mathtt{DSO}_1$, IPP, two feeder lines and the four circuit breakers. Secondly, a model is also developed for the 5G based communication of the four IEDs. Finally, a model is developed for the blockchain transactions in the network. The overall system is modelled by connecting the atomic sub-models using the Join formalism in Möbius. The reward model functionality in Möbius is used to collect statistics of interest. We describe below a summary of the atomic models used in our simulation study:

1) Power system: Figure 8 shows the atomic model for the power system network of the self-healing application. It has 6 places. `Power_line_Ok` represents the initial state of feeder 1 while feeder 2 has initial `No_Power` state. Bother feeder lines can be in `Power_line_Failed` state. The Breaker place represents the state of the 4 circuit breakers. The `Customer_OK` and `Customer_No_power` states represent the power supply states for $\mathtt{DSO}_1$.

2) Communication model: Figure 9 shows the atomic model of the communication between the four IEDs in the network. The communication is based on IEC61850 publisher-subscriber multicast mechanism whereby IEDs publish a change of state of their breaker state (i.e., failed or OK) to the other IEDs (e.g., `Goose_A_broadcast`). An IED receiving GOOSE messages will initiate a self healing activity to execute

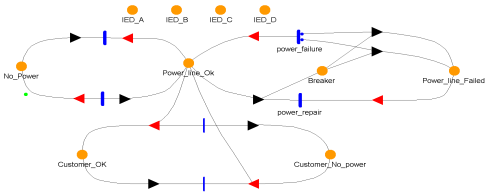


Fig. 8. Power system atomic model

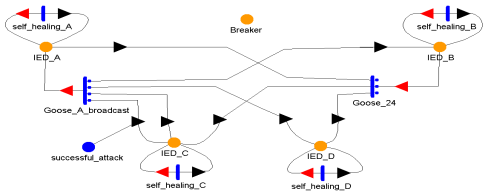


Fig. 9. 5G communication atomic model

a predefined action on its breaker (e.g., `self_healing_C`). In addition, the atomic model shows a scenario whereby there is a man-in-the-middle cyber attack on the communication between `IED_A` and `IED_C` with a probability of success. In a successful attack, the communication of a failed breaker state (e.g., breaker status = open) from `IED_A` is altered to an OK state (breaker status = closed).

3) Blockchain model: The atomic model of the blockchain transactions process is shown in Figure 10. The IEDs with blockchain nodes form an overlay network topology, in which we assume nodes forming a ring topology. If a feeder line fails and the breaker state changes to open, an IED publishes GOOSE messages to the other IEDs in the network and at the same time generates a blockchain transaction corresponding to the GOOSE message. This transaction propagates to the other IEDs in the network with each receiving IED validating the new arrival. The validation requires arrival of the blockchain transactions from all IEDs connected to the IED. Once the transactions are validated, a corrective actions (i.e., `self_healing_A`) will be executed on the IED if the validated state is different from the initial state received from the GOOSE multicast message. Block generation process is an independent process that was not considered in this atomic model. This is because block generation events do not affect the overall performance of our study model. It is a process that collects valid transactions into a block and pushes the new block to the neighbor nodes for bookkeeping.

TABLE I
DELAY / SERVICE TIME

Component	Circuit breaker	IED	Communication (5G)	Blockchain transactions
Delay / Service time [msec]	1000	10	10	1000

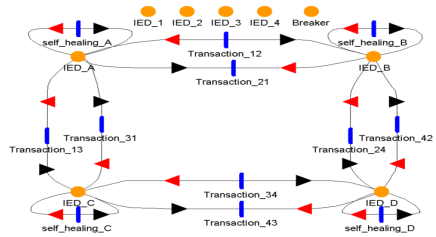


Fig. 10. Blockchain transactions interaction process atomic model

B. Case Study

In the case study, we evaluate the downtime and unavailability of power supply to $\texttt{DSO}_1$ in a normal self-healing architecture and the proposed blockchain self-healing architecture. We study the impact of cyber attacks on the 5G-based communication between the IEDs by modeling a man-in-the-middle attack between `IED_A` and `IED_C`. We conduct a sensitivity measure of cyber attacks with varying attack probability of success and its impact on the $\texttt{DSO}_1$ power unavailability.

The delay and service times assumed for the circuit breaker, IEDs, 5G communication and blockchain transactions are shown in Table I while the failure rates and repair time for the feeder lines are shown in Table II.

TABLE II
FAILURE AND REPAIR RATES

Component	Failure rate [/year]	Repair rate [hours]
Feeder lines	0.01	4

C. Results

1) The impact of attack success probability: Figure 11 shows the downtime and unavailability experienced by $\texttt{DSO}_1$ with increasing probability of successful attack for the normal self healing architecture and the proposed self-healing with blochchain support. The x-axis represents the probability of successful attack, p, and the y-axis indicates the down time in seconds per year, t_U. As can be observed from the figure, self healing with blockchain support has a significant reduction in the downtime on $\texttt{DSO}_1$ compared to the normal self healing operation. Furthermore it is observed for both architectures, the downtimes increase with increasing attack probability. For the normal self healing architecture, With probability $p = 0.1$, the downtime of $\texttt{DSO}_1$ is $t_D = 2676.39$ [seconds/year] (44.6 minutes/year) for normal architecture while downtime $t_D = 4.8$ [seconds/year] with blockchain support architecture. With probability of successful attack increased to $p = 0.8$, the downtime for $\texttt{DSO}_1$ is $t_D = 20659.2$ [seconds/year] (344.3 minutes/year) for normal architecture while downtime 8.6 seconds/year with blockchain support architecture.

For the normal self healing, when there is a successful attack on the communication between `IED_A` and `IED_C`, IPP can not

supply power to $\texttt{DSO}_1$. Hence, $\texttt{DSO}_1$ remains in a no power state until the feeder 1 ($\texttt{R} - \texttt{DSO} \leftrightarrow \texttt{DSO}_1$) is repaired. On the other hand, with our proposed architecture, the blockchain transactions act as second-tier security mechanism which enable the actions of a successful attack between IED_A and IED_C to be reversed. Hence, $\texttt{DSO}_1$ will remain in no power state for sometime until the blockchain transactions invalidate the successful attack actions. Power is restored to $\texttt{DSO}_1$ with feeder 2 ($\texttt{DSO}_1 \leftrightarrow$ IPP). We assume here that the blockchain transaction processing time per IED is 1 second.

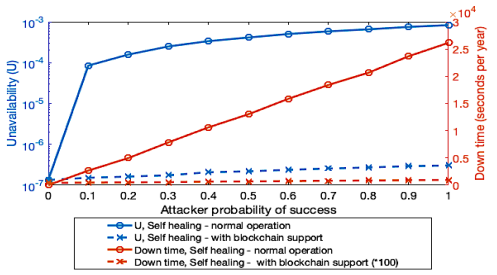


Fig. 11. Downtime and unavailability of $\texttt{DSO}_1$ for normal self healing architecture and proposed self healing architecture with blockchain support

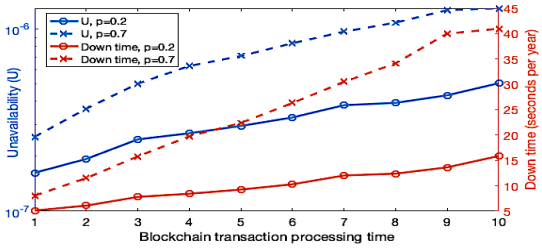


Fig. 12. Downtime and unavailability of $\texttt{DSO}_1$ for proposed self healing architecture with blockchain support considering varying blockchain transaction processing time per IED

2) Effect of blockchain transactions processing time: Figure 12 shows the downtime and unavailability observed at $\texttt{DSO}_1$ when we consider different blockchain transaction processing times per IED for our proposed architecture. We evaluate based on two attack probabilities. It was observed that for all values of blockchain transaction processing time considered, the downtime was higher for probability of successful attack, $p = 0.7$ compared to $p = 0.2$. The downtime increases linearly with the transaction processing time, t_p, while the unavailability (U, increases exponentially (note the log-scale on the axis).

In the cases evaluated, even though the downtime increases due to increasing blockchain processing times, the blockchain support architecture still achieves a much larger savings in the downtime observed as compared to the normal operation with a 4 hour feeder repair time.

V. Conclusion and Future Work

Self healing applications among distributed entities such as DSOs, microgrids and IPP having their own administrative domains require building trust between the participating units. However, due to the time-critical nature required in self-healing operations, there is the challenge of security mechanisms to deploy in order not to affect speed of operations. This paper has proposed an architecture based on blockchain as a second-tier security layer to validate the time critical messages in a self-healing application. The architecture provides security for the real-time application in that actions may be reversed after invalid transactions are detected, while the ledger maintains the bookkeeping. A simulation study was conducted and it was shown that our architecture results in less downtime (no power state) for the DSO considered when compared to a normal self healing.

The proposed architecture can address the impact of cyber-physical security for real-time self-healing in the SDG thereby increasing the grid immunity towards cyber-physical attacks. In the future work, we plan to further study the impact of blockchain in providing support for self-healing operations.

References

[1] Jinju Zhou et al. "What's the difference between traditional power grid and smart grid? — From dispatching perspective". In: *2013 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*. 2013, pp. 1–6.

[2] Pratik Kalkal and Vijay Kumar Garg. "Transition from conventional to modern grids: Modern grid include microgrid and smartgrid". In: *2017 4th International Conference on Signal Processing, Computing and Control (ISPCC)*. 2017, pp. 223–228.

[3] IEC. *IEC standard for communication networks and systems for power utility automation - Part 90-1: Use of IEC 61850 for the communication between substations*. Tech. rep. IEC, 2010.

[4] Petar Popovski et al. "5G Wireless Network Slicing for eMBB, URLLC, and mMTC: A Communication-Theoretic View". In: *IEEE Access* 6 (2018), pp. 55765–55779.

[5] Haiyu Ding et al. "Use Cases and Practical System Design for URLLC from Operation Perspective". In: *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*. 2019, pp. 1–6.

[6] Tianyou Li and Bingyin Xu. "The self-healing technologies of smart distribution grid". In: *CICED 2010 Proceedings*. 2010, pp. 1–6.

[7] IEC 62351. "Power systems management and associated information exchange - Data and communications security - Part 6: Security for IEC 61850". In: (2020).

[8] Juan Hoyos, Mark Dehus, and Timthy X Brown. "Exploiting the GOOSE protocol: A practical attack on cyber-infrastructure". In: *2012 IEEE Globecom Workshops*. 2012, pp. 1508–1513.

[9] Liang Zhu et al. "Priority-Based uRLLC Uplink Resource Scheduling for Smart Grid Neighborhood Area Network". In: *2019 IEEE International Conference on Energy Internet (ICEI)*. 2019, pp. 510–515.

[10] Dimitrios Sikeridis et al. "A blockchain-based mechanism for secure data exchange in smart grid protection systems". In: *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*. 2020, pp. 1–6.

[11] Yaqin Wu and Fuxin Song Wang. "Hybrid Consensus Algorithm Optimization: A Mathematical Method Based on POS and PBFT and Its Application in Blockchain". In: Apr. 2020, pp. 18–23.

[12] IEC. "IEC standard for communication networks and systems for power utility automation - Part 90-5: Use of IEC 61850 to transmit synchrophasors information according to IEEE C37.118 IEC 61850-90-5 TR Ed 1.0. Technical report". In: (2012).

[13] Shravan Gaonkar et al. "Performance and Dependability Modeling with Möbius". In: *ACM SIGMETRICS Performance Evaluation Review* 36 (Mar. 2009), pp. 16–21. DOI: 10.1145/1530873.1530878.

ISBN 978-82-326-6206-7 (printed ver.)
ISBN 978-82-326-5570-0 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU
Norwegian University of
Science and Technology