Blockchain Governance: A Model for Optimal Decision Making using Game Theory

By

Rabia Akram

00000276476

Supervisor

Dr. Tauseef Ahmed Rana

Department of Computer Science

A thesis submitted in conformity with the requirements for the degree of

Masters of Science in Software Engineering (MS SE)

In

Military College of Signals

National University of Science and Technology (NUST)

Islamabad, Pakistan

September, 2022

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis written by ~~Mr~~/MS **Rabia Akram,** Registration No. **00000276476**, of **Military College of Signals** has been vetted by undersigned, found complete in all respect as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial, fulfillment for award of MS/MPhil degree. It is further certified that necessary amendments as pointed out by GEC members of the student have been also incorporated in the said thesis.

Signature: ___________________________

Name of Supervisor Dr. Tauseef Ahmed Rana

Date: _______________________________

Signature (HoD): ______________________

Date: _______________________________

Signature (Dean/Principal): _____________

Date: _______________________________

Copyright Notice

ABSTRACT

Blockchain is an "immutable ledger" a cutting-edge emerging technology that stores and tracks all kinds of data. Whereas Blockchain governance is a decentralized process of proposing, accepting, and rejecting change. Being decentralized in nature has its consequences such as decision-making and achieving mutual consensus is costly and challenging. Existing governance models are not so effective due to a lack of decision-making in the Blockchain environment such that users cannot connect and communicate to make instant decisions for the proposed change. Each proposal takes a long time to get accepted or rejected as a result, the community ends up initiating a fork. To solve these issues, this thesis proposes a unique governance model by the implementation of the game theory approach for instant decision-making. It uses agent-based simulation to implement ON-chain governance in conjunction with percolation theory and voting mechanism to solve connectivity and communication issues. The uniqueness of our framework is shaping opinion by modeling three types of user interaction and their influence on each other after interaction i.e. Confident, unsure and fixed users. Once a majority of the community shares the same opinion at this point with the help of game theory each user will decide to fork or not to fork such that it maximizes their utility and also produce a less resulting impact on the latency of the original Blockchain. The results from the Blockchain governance simulation confirm the validity of our framework and its effect on the latency of the Blockchain network. The research concludes that after interaction when a decision is made with a minority of network agreeing with the proposal and the majority rejecting it, the resultant effect on latency will be less as compared to when a majority of network agrees and forks apart leaving the original Blockchain with higher latency and lower transaction rate.

Acknowledgements

On the road to success, there is always we, not me. There is always another who is standing behind you, maybe not coming on the forefront but always at your back. Praying for you and supporting you through it all. I am thankful to my Lord Allah S.W.T to have instilled the strength and courage in me to work. For He is the one who has guided me throughout this work at every step and for every new thought which came to me. Indeed, I could never have done nothing without the will and help of the Almighty. I would also like to express special thanks to my supervisor Dr. Tauseef Ahmed Rana and co-supervisor Dr. Ayesha Maqbool for their help and support throughout my thesis. Without their help I wouldn't have been able to complete my thesis. I appreciate their patience and guidance throughout the whole thesis.I would also like to thank my GEC Members Asscoc Prof Dr. Ihtesham ul islam and Asst Prof Dr. Ikraum Syed for always being available for help and support throughout my degree in coursework as well as in thesis. Finally, I would like to express my gratitude to the people who have been cheering me on and being my firm support since day one; my beloved parents, siblings and my daughter, for their patience, love and support through all the tough times and for never giving up on me. It is solely because of them that I was able to dedicate myself to my work and give my best while they handled everything else.

DEDICATION

Dedicated to my parents, Mr. Muhammad Akram & Shameem Akhtar and my Daughter Zainab whose unwavering support and cooperation drove me to this accomplishment.

TABLE OF CONTENTS

TABLE OF CONTENTS vii

1 INTRODUCTION 1

  1.1 Blockchain Governance 1

  1.2 Game Theory 2

  1.3 Research Questions 2

  1.4 Research Objectives 2

  1.5 Research Methodology 3

  1.6 Thesis Outline 4

2 PRELIMINARY STUDY 5

  2.1 Blockchain Technology: Brief History 5

  2.2 Blockchain Governance 5

    2.2.1 OFF-chain Governance 7

    2.2.2 ON-chain Governance 8

  2.3 Game Theory 13

  2.4 Solution to existing problems 14

  2.5 Conclusion of chapter 15

3 BLOCKCHAIN BASED SIMULATION MODEL 16

  3.1 Introduction 16

  3.2 Blockchain Architecture 16

  3.3 Blockchain simulation agents 19

  3.4 Network Setup 20

  3.5 Blockchain Simulation 21

    3.5.1 Simulation Results 23

  3.6 Conclusion 26

4 BLOCKCHAIN FORK SIMULATION 27
4.1 Introduction 27
4.2 Blockchain fork simulation 27
4.1.1 Create Fork 28
4.3 Results 30
4.4 Conclusion 32
5 PROPOSED DECISION-MAKING MODEL 33
5.1 Introduction 33
5.2 Percolation Theory 33
5.2.1 Percolation Simulation 34
5.2.2 Results 36
5.3 Voter Model 36
5.3.1 Voter Model implementation strategies 37
5.4 Voter Model Simulation Results 41
5.4.1 Neutral Interaction 41
5.4.2 Extremal Interaction 44
5.4.3 Fixed point Interaction 46
5.5 Conclusion 47
6 A Model for Optimal Decision-making using Game Theory 49
6.1 Introduction 49
6.2 Simulation and results 50
6.3 Conclusion 54
7 CONCLUSION AND FUTURE WORK 56
7.1 Conclusion 56
7.2 Future Work 57

REFERENCES 58

LIST OF FIGURES

Figure 3.1 Blockchain Layered Architecture 17
Figure 3.2 Network setup 20
Figure 3.3 Blockchain Setup blue nodes 150, Red %validator nodes 50 21
Figure 3.4 Blockchain simulation 22
Figure 3.5 Average delay for T.R 1 24
Figure 4.1 Fork simulation interface 28
Figure 4.2 Create Fork 29
Figure 4.3 Latency plot for varying %Fork_Split_Nodes (a 20%, b50%, c75%, d 100%) 30
Figure 5.1 Node connectivity 34
Figure 5.2 For Pc = 0.1 and Pc = 1 36
Figure 5.3 Neutral opinion startagy 38
Figure 5.4 Extremal opinion. 39
Figure 5.5 Fixed opinion startagy. 40
Figure 5.6 Neutral interaction for equal +ve, -ve influencers 41
Figure 5.7 Neutral interaction opinion distribution. 42
Figure 5.8 Neutral interaction opinion distribution for –ve agent majority 43
Figure 5.9 Neutral interaction opinion distribution for high percolation 43
Figure 5.10 Neutral interaction opinion distribution for one +ve influncer more with p 0.3 44
Figure 5.11 Extremal Interaction with equal influencers 44
Figure 5.12 Extremal Interaction with equal influencers, extremal rate 0.3 45
Figure 5.13 Extremal Interaction with more –ve number of influencers 45
Figure 5.14 Fixed point Interaction with equal influencers. 46
Figure 5.15 Fixed Interaction with more -ve influencers 47
Figure 6.1 Blockchain decision-making Setup 50
Figure 6.2 Majority of negative influencer after interaction 51
Figure 6.3 Latency plot after fork 52
Figure 6.4 Effect on latency of original chain and forked –ve influencer’s chain 52
Figure 6.5 Majority of positive influencer after interaction 53
Figure 6.6 Effect on latency of original chain and forked –ve influencer’s chain. 53
Figure 6.7 Majority of -ve influencer after fixed interaction 54
Figure 6.8 Effect on latency of original chain and forked fixed –ve influencer’s chain. 54

LIST OF TABLES

Table 2.1 Governance Methods 7
Table 2.2 Blockchain Governance in different platforms 9
Table 2.3 Types of Game theory 13
Table 3.1 Blockchain Stakeholders in proposed model 19
Table 3.2 Affect of change in %Validators on average dalay 25
Table 3.3 Average delay for varying number of Nodes in network 25
Table 4.1 Latency effect before and after fork 31
Table 5.1 Neutral Interaction 38
Table 5.2 Extremal Interaction 39
Table 5.3 Fixed point interaction 40

Chapter 1

1 INTRODUCTION

This chapter introduces Blockchain Governance and Game theory. It addresses the existing issues in Blockchain governance. Moreover, highlights the research questions and organization of thesis.

1.1 Blockchain Governance

Blockchain is a new cutting-edge technology. To simply define Blockchain we say it's "an immutable record or ledger" use to store all kinds of data and can also manage and track asset ownership. It combines many concepts, methodologies, and technologies such as cryptography, ledgers, immutability, group consensus, trustless-ness, etc. [1]. Satoshi Nakamoto released a white paper on Bitcoin in the year 2009, which used the concept of Blockchain to implement transactions based on cryptocurrency, which gave hype to the Blockchain as an immutable network that connects everyone to perform transactions [2]. Concurrently, every technology needs some basic principles, rules, and regulations to work smoothly. For this purpose Blockchain introduced the idea of governance [3] "a process of creating, implementing and monitoring policies by members of an organization". Blockchain governance is a decentralized process such that the way people propose changes and the mechanism of acceptance or rejection of those changes is distributed. Change is important according to a Blog post [5] by Fred Ehrasam CEO of Ethereum, Success of Blockchain lies in its continuous change according to needs of the environment. Consequently, Blockchain has continuous growth in such a way that in the first generation there was no concept of governance. The second generation introduced the OFF-chain governance which was implemented, tested, and worked efficiently but still required the implantation of decentralized decision-making as it was more aligned toward centralized governance. To involve everyone in decision-making, the third generation introduced ON-chain governance suitable for the distributed environment but it still has a long way to go. History shows the time-consuming, vague, and chaotic decision-making [6] in Blockchain due to a lack of connectivity and communication among users of the network. Hence, Blockchain needs a way to achieve instant decision-making in well-connected network and extensive user participation to accept or reject the update for a change proposal. To solve these issues we are proposing a framework to make an instant decision based on game theory.

1.2 Game Theory

Game theory is a study of mathematical models of strategic interaction between rational decision-makers. Ziyao et al. [7] state in their survey that game theory is a new solution for the Blockchain community. N.Alzahrani and N.Bulusu [4] used game theory in consensus protocol, which models honest validator's behavior. Another study [53] uses stochastic game theory to enhance the security of Blockchain network. Whereas none in existing literature used game theory for Blockchain governance. Hence, for effective governance, a structured framework is required where the community can easily communicate, discuss, propagate and monitor the decisions made. This leads to a focus on evolutionary stochastic games such that users in the network can predict the behavior of each other and respond accordingly to achieve equilibrium and maximize utility. This can also discourage forks, DOS attacks, community split, etc. Fork is defined as two or more routes to follow to satisfy the need. A portion of the community that questions the legitimacy of a governance decision can make a copy of the Blockchain network and maintain a "fork" of the Blockchain outside of the governance of the original chain as a result affecting its latency. Hence, to solve connectivity and communication problems our model proposes an influencing voting mechanism where users influence each other by holding an opinion about proposal. Moreover, Percolation theory provides 3 dimensional connectivity for each users' communication in the way that high percolation generates widespread of influence.

1.3 Research Questions

For the purpose of finding a solution which can assure effective governance, we have defined the following set of research questions for our study:

- **RQ1:** What is the effect on latency of network when fork occur?
- **RQ2:** How to achieve network connectivity?
- **RQ3:** How a Blockchain will perform a decision-making?
- **RQ4:** when to fork?
- RQ5: How can we model a system for optimal decision-making?

1.4 Research Objectives

By stating research objective we can answer RQ5: How can we model a system for optimal decision-making? This research aims to fulfill three major goals:

- To solve RQ2 we need application of percolation theory for network connectivity.
- Use of influencing voter model strategy to shape opinion of network which solves RQ3 by achieving pure decision.
- Applying stochastic game theory model to predict RQ4: when to fork? Based on voter model results. So majority network stays on the chain without effecting the working of existing Blockchain solving RQ1. Moreover, it will also predict response of community to the protocol update.

1.5 Research Methodology

This research introduces the concept of Blockchain governance and game theory. We conducted a literature review to identify weaknesses in existing governance strategies, closely related work, game theory strategies, and historical forks to model the behavior of our framework. Analyses of existing issues lead us to propose an Agent-Based Modeling (ABM) to simulate the Blockchain governance model for optimal decision-making using game theory. It is difficult to solve decision-making analytically so we choose to develop and simulate the framework in NetLogo by changing various parameters during simulation. To verify and validate the results we will use stochastic modeling, which models the probability of various outcomes under different conditions, using random variables. The research concludes with the final simulation results and future work.

1.6 Thesis Outline

This thesis consists of the following chapters:

- **Chapter 1:** This chapter introduces Blockchain Governance and Game theory. It addresses the existing issues in Blockchain governance. Moreover, highlights the research questions and research methodology of a thesis.
- **Chapter 2:** It presents the detailed study on Blockchain Governance, game theory, and historical fork decision-making.
- **Chapter 3:** This chapter proposes Architecture, simulation, and results for Blockchain.
- **Chapter 4:** This chapter proposes simulation and results for the Blockchain fork. Moreover, it provides stochastic modeling of the effect on latency of blockchain after a fork.
- **Chapter 5:** This chapter proposes the implementation of the percolation theory for network connectivity and the influencer voter model for decision-making. Various interaction strategies are listed in Tables and simulated to verify the results.
- **Chapter 6:** This chapter amalgamates Blockchain fork, percolation theory, voter model, and game theory. Simulation and results are discussed in detail.
- **Chapter 7:** It Concludes the thesis and also provides future work to further explore this research

Chapter 2

2 PRELIMINARY STUDY

This chapter provides a brief introduction to Blockchain and Blockchain governance. A literature review of existing governance models to identify weaknesses to propose a solution. Furthermore, listed some game theory strategies to select the best possible strategy for research.

2.1 Blockchain Technology: Brief History

Blockchain technology is distributed ledger-based technology which offers a way to distribute digital information among various computers in a network. A Blockchain is a sequence of immutable records of information that is time stamped and managed by a bunch of PCs in a network. Each PC stores a copy of the information, all the PCs in the network make sure that the same information is stored across the network and whenever anything new is added, all the PCs on the network validate the information against some rules and after mutual agreement, the new information is added to the chain. This is how this technology, by nature, offers a guarantee that the digital information distributed across the network is incorruptible, immutable, and cannot be tampered with. A white paper [2] was published by Satoshi Nakamoto in 2008 proposing a system for electronic transactions which will not rely on trust and uses a peer-to-peer network that will record the public history of those electronic transactions. In 2009, the Bitcoin network was created using the idea which was given in the whitepaper. A new age of cryptocurrencies began [9] and many digital currencies were introduced one after the other in the market using the same basic concept that was used to make Bitcoin. In the second version of Blockchain, Ethereum [8] Platform was introduced with the concept of a scripting technology, Smart Contract, which could be used to build decentralized applications. In the third version of Blockchain, various decentralized applications for various domains including supply chain [10], digital identity [11] and financial tech, etc. were developed.

2.2 Blockchain Governance

Every technology needs some basic principles, rules, and regulations to work smoothly. For this purpose Blockchain introduced the idea of governance [3] defines the establishment of policies, and continuous monitoring of their proper implementation, by the members of the governing body of an organization. For a governance process to work effectively, rules should be aligned with the overall participants' goals, and the rulers should enforce positive and negative actions within this governance structure. Blockchain governance is a decentralized process such that the way people propose changes and the mechanism of acceptance or rejection of those changes is distributed. Most organizations make decisions on their future projects and proposals under a central authority. Whereas Blockchain is a decentralized system that implements the concept of peer-to-peer communication if a central authority makes a decision it will lead to a single point of failure and may expose the Blockchain to many other threats. Thus the application of structured decision-making is important to incorporate the changes proposed by users so Blockchain can keep on evolving. As a Blog by Fred Ehrasam CEO of Ethereum states [5] successful Blockchains evolve to fulfill the needs of stakeholders. Initial design can attract users but change is also equally important to grow and keep users enticed to Blockchain. To understand Blockchain governance we first need to know key stakeholders of Blockchain e.g. Core developers: maintain and develop the Blockchain, Node operators: maintain the copy of Blockchain ledger, miners: secure Blockchain by hash power to validate transactions, and users: people using the network. The degree of involvement of each stakeholder can be judged by applied governance strategy. Table 2.1 lists different types of strategies.

Governance	**Strategy**
Decentralized Autonomous Organization (DAO) [13]	Allows for group of governance through a combination of smart contracts and issuing tokens.
Liquid Democracy	System where everyone can vote for themselves or delegate their votes.
ON-chain [14]	Direct voting on platform over a proposed protocol change. Voting is done by the stakeholders of the platform. Once voted in by the community, upgrade is hard-coded into the protocol.
OFF-chain [14]	A decision-making that first takes place on a social level. Census is done by leaders of community and is afterwards being actively encoded into the protocol by the developers.

Table 2.1 Governance Methods

This Table2.1 lists all the governance strategies but we are interested in strength and weakness of ON-chain and OFF-chain governance to model our framework.

2.2.1 OFF-chain Governance

Blockchain has continued to evolve. In the first generation, there was no concept of governance. The second generation introduced the OFF-chain governance defined as the decision function view of governance by Vitalik Buterin co-founder of Ethereum [15]. As Table 2.1 describes OFF-chain, inputs are various wishes by multiple stakeholders (developers, node operators, miners, shareholders, etc.) and output is a final decision that can't be changed after the census. Proposed changes are implemented [14]. Fred Ehrasam co-founder of Coinbase [16] wrote an article on Blockchain governance that was widely debated within the community. OFF-chain governance can be seen as decision-making that first takes place on a social level. Census is done by leaders of the community and is afterward actively encoded into the protocol by the developers. Ehrasam argues that Bitcoin and Ethereum mostly rely on OFF-chain governance. For instance, Bitcoin developers share their improvement proposals (BIPs) through a mailing list and Ethereum collects improvement protocols (EIPs) on Github. Bitcoin and Ethereum both fall under the category of OFF-chain governance because for major decisions to be implemented community relies on the leaders such as Ethereum is reliant on the guidance of Ethereum's creator Vitalik Buterin. If someone disagrees with the proposed change then the party can initiate a hard fork [17] as Bitcoin has a Bitcoin cash hard fork and Ethereum has Ethereum classic created as a result of a famous DAO attack. If we want to see the advantages of OFF-chain, then it can be seen in form of Bitcoin and Ethereum's success. Everything has associated drawbacks, OFF-chain governance is relatively centralized because decisions are made by the Blockchain team, miners, or leaders which in turn excludes casual user who lacks the knowledge to implement decisions [14]. Secondly, decisions can take a long time to implement due to time-consuming consensus just like representative democracy.

2.2.2 ON-chain Governance

To make Blockchain governance decentralized the third generation introduced ON-chain governance "the coordination view of governance" [15]. ON-chain governance declares code is law means that all software changes and upgrades, once voted in by the community, are hard-coded into the protocol [12]. The motivation of ON-chain governance is to transfer incentives from miners, node operators, and developers to end users. This concept still raises so many questions on whether users have adequate knowledge to vote for critical decisions such as platform development direction. Consensus achieved in ON-chain governance is direct voting. Researchers comment that no use case shows the validity of this consensus mechanism, so it is still in the testing and growing phase. One common example of an ON-chain governance project is TEZO [18] which advertises itself as a truly leaderless smart-centered platform. Here users can vote on anything including chain rewrites and a decision is made by involving the majority of the community present at the time of proposal. Moreover, any developer can submit a code update to the project together with an invoice which is paid if the update is implemented. Dfinity [19] is also ON-chain governance that deals with problems associated with hacking like the DAO hack [31]. It votes on whether to roll back or not to recover from the loss caused in events of such an attack. It implements a "Blockchain Nervous System" (BLS) algorithm for protecting users from attacks and dynamically improving the ON-chain security and governance. Another example is Decred, which implements a more complex ON-chain governance model based on distributing the power between stakeholders and miners. Decred has a hybrid proof of work/proof of stake consensus mechanism [20]. We have listed some platforms using different strategies of governance in Table 2.2.

PLATEFORM	Governance	DESCRIPTION
Bitcoin [2]	OFF-chain	Consensus done by leaders of community (developers, miners, team leader etc.)
Ethereum [8]	OFF-chain	Benevolent dictator for life, Ethereum is reliant on the guidance of Ethereum's creator Vitalik Buterin. While the input from team is welcome.
Hyperledger Frameworks [21]	Open Governance Linux Foundation IBM	Unique approach to consensus. The Hyper ledger Technical Steering Committee (TSC) is selected by active maintainers and contributors community for making technical decisions.
CORDA [22]	Open Governance	The Corda Network Governing Body will be selected to represent the interests of all users.
Tezos [18]	ON-chain	Census is achieved by voting.
Dfinity [19]	ON-chain	Uses voting mechanism.
Decred [23]	Hybrid of OFF-chain and ON-chain	Direct voting ON-chain. Voting on higher-level issues occurs OFF-chain in Politeia, a web platform where the community can submit, discuss and vote on proposals.

Table 2.2 Blockchain Governance in different platforms

ON-chain governance provides many benefits such as a Decentralized decision-making process, quicker consensus, transparency, and fewer malicious hard forks. The problem with ON-chain is its dependency on votes from the community which can be challenging if the network grows large causing scalability issues. Another issue could be that after consensus minority may not feel satisfied with the output, so they could leave the community and end up initiating a hard fork into a new Blockchain.

The literature review shows there are similar issues in both strategies i.e. connectivity, communication, and instant decision-making. Moreover, after a decision or voting if users are not satisfied they will end up initiating a fork. The fork is defined as two or more routes to follow to satisfy the need. Bitcoin fork can be categorized into two types of events occurring in Blockchain [24]. Blockchain communities generally would prefer to stay together than to fork apart. However, because Blockchains are software running on the internet, a portion of the community that questions the legitimacy of a governance decision can make a copy of the Blockchain network and maintain a "fork" of the Blockchain outside of the governance of the original chain as a result affecting its latency. For example, Bitcoin Cash and Ethereum Classic are examples of Blockchain forks [25]. There are two types of forks, a soft fork, and a hard fork. Soft forks change the rules in such a manner that users following the old rules will still get on the new chain. An example of a soft fork is a restriction imposed on a block size limit from 1MB to 500kB. Even though a 1MB block was previously considered valid, full nodes that update to support this soft fork reject any blocks larger than 500kB after the soft fork activates[24, 26]. Whereas a hard fork is a change to the protocol that loosens the rule set enforced by an original chain. A block that is considered invalid before the hard fork activation, will be considered valid for the users following the fork rules after fork activation. An example is a hard fork that increases the block size limit from 1MB to 2MB. Even though a 2MB block was previously considered invalid, full nodes that update to support this hard fork will accept any blocks up to 2MB in size after the hard fork has been activated. [24, 26]. If the community forks, it affects the throughput of the original Blockchain. Hence, the fork is an unfavorable outcome for the Blockchain network happens due to inefficient decision-making.

We conducted a study to analyze some historical forks and how their decisions were made. We will be listing historical forks of Ethereum, Bitcoin, Decred, and Tezos along with their decision-making. There are 105 Bitcoin fork projects in total. Of those, 74 are considered active projects relevant to holders of Bitcoin (BTC). The remaining 31 are considered historic and are no longer relevant [25,29] Changing the block size of Bitcoin was proposed in the 2017 proposal Bitcoin improvement proposal (BIP) 101 stating "Increase block size from 1MB to 8MB" published by one of bitcoin's core developer Gavin Andresen [29, 30]. Community discussions are held on social platforms such as Github, Bitcointalk, Twitter, etc. In 2017 a meeting with miners and Core Developers took place referred to as the "New York Agreement", where an agreement was reached to support SegWit and a 2MB hard fork Known as SegWit2x. As a result, Bitcoin forked into Bitcoin cash. Everyone who held a Bitcoin received an equal amount of Bitcoin Cash such that a 1:1 ratio of BTC: BCH.

Whereas Ethereum [8] declared its road map long before in the form of future hard forks. Which will implement Ethereum Improvement Proposals (EIPs) for networks' slow development but on 17th June 2017 DAO got hacked due to a coding error and 1/3 or $50 million were stolen from the Dao wallet. As a result, the value of ETH quickly went from $20 to $13. Hacker however could not get access to stolen money until after 28 days due to a smart contract. The community had two ways one to do nothing, the other to create a hard fork. As too much money was involved so the community decided to not sit idle but create a hard fork and rewrite a ledger to recover the money. Which lead to the first hard fork at a block height of 1,920,001, Ethereum and Ethereum classic on 20th July 2016. Hence, Fork caused a split in the community. People who didn't agree with the decision stayed on the original chain. Who owned ETH received an equal amount of ETC.

Decred [23] has hybrid governance. Decred change proposal (DCP) Decentralize Treasury Spending [32] was submitted and viewed by a community of platforms for the censorship, and documentation style. Reddit-style voting was conducted on Politeia. This proposal was published ON-chain, where stakeholders will vote to approve it several days after being published. 97% of the community voted YES and 2.5% voted NO. Hence, the proposal got accepted by the community to enter the implementation phase.

Tezos [18] is an ON-chain governance platform. A proposal is proposed ON-chain then it goes through 4 phases of voting to get acceptance and implementation. It has bakers who vote for change and delegators who delegate their voting right to a baker if they don't want to vote on a proposal. Tezos specifies a particular quorum and participation threshold to be met to proceed to the next phase and acceptance of the proposal. Its criteria for proposing and acceptance of change are very rigid that's why it has o forks until now. It is the best ON-chain platform so far. To analyze the decision-making process in Tezos we studied the decision-making process for Athens's (Pt24m4xiP) first amendment proposal of Tezos. Nomadic Lab proposed two proposals Athens A and Athens B in February 2019. [32] Such that Athens' A: reduction in roll size from 10,000 ꜩ to 8,000 ꜩ and increment the gas limit and Athens B: increased the gas limit. 216 bakers voted with

47,049 rolls, total Participation was 84.35% of which 99.89% Supermajority said yes to Athens A. Hence, Athens A activates on May 30th after 12:40 am.

After reviewing the historical forks we can say that decision-making plays a very important role in Blockchain governance Blockchain being decentralized, is not under the control of one person or team, it needs a mechanism to make decisions. Such that stakeholders should interact with each other to achieve similar goals and maximize the networks' utility. Structured decision-making can solve connectivity and communication issues whereas, Time-consuming process of decision-making can be solved by game theory.

2.3 Game Theory

Game theory is a study of mathematical models of strategic interaction between rational decision-makers. Ziyao et al. [7] propose in their survey that game theory is now a new solution for the Blockchain community. Through this mechanism nodes in the network can predict the behavior of each other and respond accordingly to achieve equilibrium and maximize utility. This can also discourage forks, DOS attacks, community split, etc. There are several types of game theories listed in Table 2.3

Game Approach	Strategy	Uses
Non-cooperative Game	The strategy that the player takes must be spontaneous, and each player is rational in cooperation.	Fork chain selection, pool selection, security issues and interaction between Blockchain users and miners.
Extensive Game	The interaction among the players using a game tree illustrates decisions made at different points with their payoffs at the end of each branch.	Selection of transactions to be included in the block, optimization of pool's mining rewards, security issues, selection of fork chain.
Stackelberg Game	The followers decide their strategies after observing the strategies of the leaders. Both aim to maximize their own utilities.	Setting transaction fees, selection of miners for verification
Stochastic Game	This are Several static non cooperative games that are repeated over time. Each static non cooperative game is called state of the game. Player can change strategy based on past actions of other player.	Mining management, the selection of chain to mine, honest mining or selfish mining. Security issues, the decision of the proper time to release the mined block and adding a block to the chain.

Table 2.3 Types of Game theory

N.Alzahrani and N.Bulusu [4] used game theory in consensus protocol, which models honest validator's behavior. Another study [53] uses stochastic game theory to enhance the security of Blockchain networks. Whereas no one in existing literature used game theory for Blockchain governance. This research focuses on evolutionary stochastic games. The data from historical fork decisions making will be used for the evolutionary stochastic game theory model to predict the future forks and community response to updates. The Blockchain governance applied with a stochastic game is an innovative idea to protect the decentralized network from a fork that causes great latency issues in the original Blockchain. A stochastic game theory can be used for making the decision such that the proper time for a fork in a manner that an original Blockchain does not suffer. Hence, it makes a spontaneous decision whether to fork or not by analyzing the current state of Blockchain. Solving the time-consuming decision-making process.

2.4 Solution to existing problems

After the detailed analysis of the Blockchain's OFF-chain and ON-chain governance, the method of proposing, implementing, and forking phenomena. We conclude that Blockchain governance has still a long way to go to thrive. Literature review Identifies the following weakness in existing Blockchain governance.

- Network Connectivity
- Communication between stakeholders
- Uninformed users
- Time-consuming decision-making

We noticed during research that no one talked about network connectivity and communication of the network during the proposed change. Studies are present to improve Blockchain network security, delays, and throughput but little to no light has been shed on network connectivity such as protocol for nodes communication, and flow of information in a distributed environment in such a way that it reaches every stakeholder of the system to have a pure decision in the well-connected network. Studies are silent in this particular aspect so we have explored this issue in chapter 5 and suggested a framework that implements the concept of network connectivity through percolation meanwhile communication is done through the influencer voter model considering a different types of interactions for different scenarios. To shape the opinion of a network with an adequate level of connectivity, users with a specific opinion will interact with other users, subsequently changing the opinion of interacting users to align the community on similar opinions. Based on these interactions and connectivity we can analyze how people will behave when the new proposal comes into the Blockchain community. The behavior spread faster and beyond across well-connected large cluster networks than across random, poorly clustered ones. To have instant decision-making, we have applied game theory, which decides whether to fork or not by analyzing the current state of Blockchain.

2.5 Conclusion of chapter

In this chapter, the literature review of Blockchain governance, fork, and game theory is presented. This research provides a detailed study of the history of Blockchain governance, and decision-making while forking and a brief study of existing game theory strategies. This chapter is concluded with a solution for listed problems to achieve pure and well-informed decisions in the Blockchain network.

Chapter 3

3 BLOCKCHAIN BASED SIMULATION MODEL

This chapter provides the base architecture of Blockchain by using agent-based modeling. It simulates the addition of blocks in the Blockchain ledger and also defines the role of stakeholders in the network. Moreover, it plots the latency of Blockchain.

3.1 Introduction

Abstract representation of a design of an existing system is called a model. It describes the dynamics of the system by combining structural, mathematical expression, and logical relationships. Whereas, Simulation [34] is a quantitative method that mimics the behavior of the system by executing the model. It is used to predict the “what-if” behavior of a system such that how it will respond under certain conditions, to new policies and designs without disturbing the system's functioning. Our Blockchains’ base model is built using Agent-Based Modeling (ABM) [35] simulation, ABM mimics real-life systems having to build formal models. In Blockchain simulation we use agents such as nodes, blocks, and transactions to interact with each other to model and predict the behavior of the system. ABM offers accurate prediction by modeling individual units of the real world and their actions are explicitly represented in the model. Existing Blockchain simulations are developed with programming languages (e.g. java, c++, or python) or special purpose simulation languages whereas we have opted for NetLogo to develop ABM-based Blockchain simulation. NetLogo is a modeling environment designed for coding and running agent-based simulations [35]. It can hold large and complex simulations with reasonable speed and run simulations with thousand or more agents while facilitating interaction with agents at runtime.

3.2 Blockchain Architecture

Blockchain is layered architecture, it can be divided into three basic layers network layer, consensus layer, and incentive layer [37]. We have built our simulation model on these layers such that the Network layer creates nodes, and controls network-level connectivity, and communications between nodes. The consensus layer defines the rules and algorithms used to reach the consensus about the state of the Blockchain. While the incentive layer manages reward distribution after achieving Conesus by participating nodes. For decision-making, we will model only two layers of Blockchain (i.e. network and consensus layer) incentive layer is out of scope for this thesis. Refer to Figure 3.1 for the Blockchain layer architecture model.

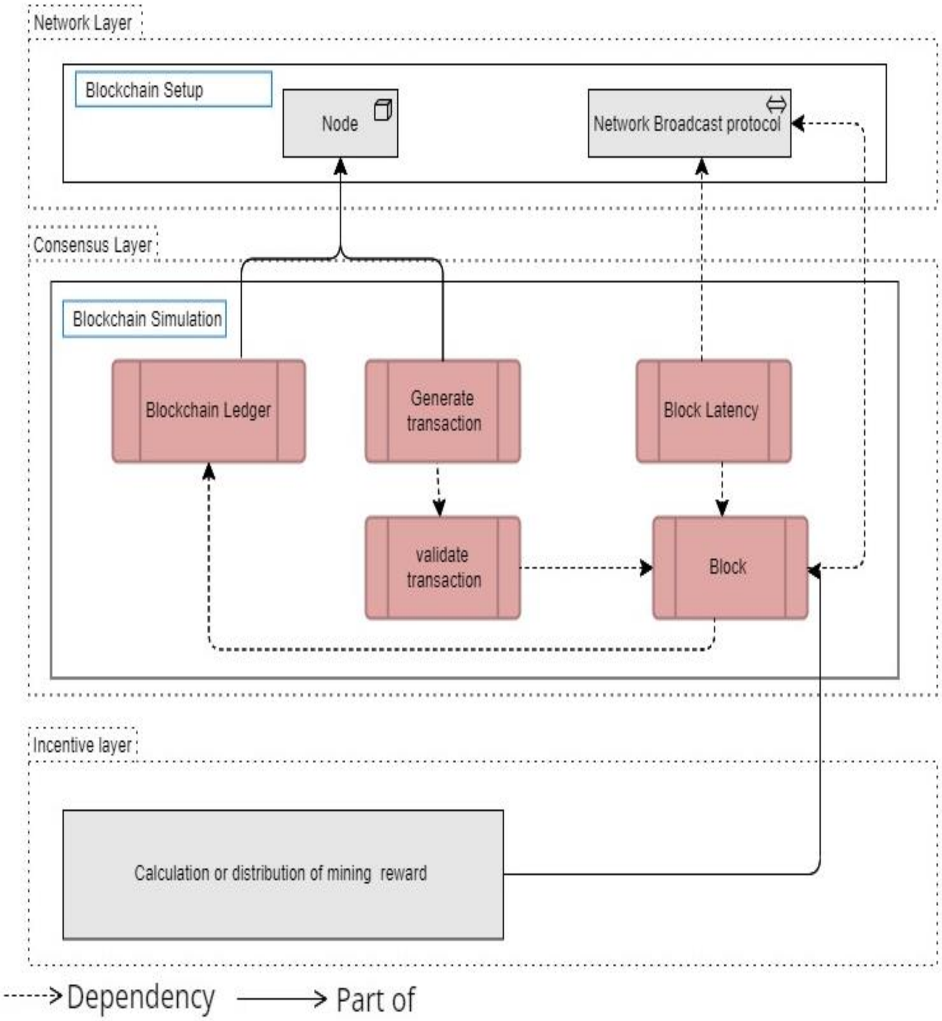


Figure 3.1 Blockchain Layered Architecture

Connected computers in the Blockchain network are called nodes. These nodes make up a Blockchain network. All participant nodes in this architecture are connected via a Blockchain network, which is responsible for broadcasting a transaction to all nodes in the network and calculating latency for each block. Blockchain Network has no central authority. Transactions generated by nodes are forwarded to the Blockchain network. In the Blockchain network, consensus takes place and the nodes in the network validate each of the transactions. After validation, these transactions are packed in new blocks which are broadcasted to every peer node in the network, and these transactions are recorded on their copy of the ledger. In the last step, it will calculate the latency of the block, latency defined as the delay encountered between input and output. While in Blockchain, we refer to the time required to add the next block of transaction in the Blockchain ledger. It is the time user encounters when he initiates the transaction to its acceptance and addition to the chain [38]. It is to note that the primary focus of our research is governance and network latency such that how opinions are built and communicated throughout the network to make pure decisions whereas consensus mechanism is secondary and incentive layer are tertiary factors. So we have not added any particular consensus mechanism. To analyze governance issues, a consensus is a fixed parameter it hardly takes 1 unit of time in the proposed model, whereas the focus is kept on how policies are made, implemented, and communicated. Although consensus can be implemented with governance this limits the scope of our research. Our main goal is to model decision-making in Blockchain. Subsequently, to model this architecture we have following stakeholders and their responsibilities listed in Table 3.1:

Agent	**Responsibilities**
Validator Node	Validator node is responsible for validating transaction, adding it to new block and broadcasting this block to network. It can also generate transitions. Maintains Blockchain leger.
Non-validator node	Non-validator nodes are responsible for generating transactions like users, then waiting for transaction to be accepted. They also maintain Blockchain ledger.
Positive influencer Node	Nodes having positive opinion. Can communicate with network and influence the network positively. These nodes can also be validator or non-validator node. Detail explanation in voter model.
Negative Influencer Node	Nodes having negative opinion. Can communicate with network and influence the network negatively. These nodes can also be validator or non-validator node. Detail explanation in voter model.
Unsure Node	Nodes unsure of their opinion. Can interact with influencers and change its opinion accordingly. These nodes can also be validator or non-validator node. Detail explanation in voter model.

Table 3.1 Blockchain Stakeholders in proposed model

3.3 Blockchain simulation agents

We have three types of agents in model node, block, and transaction. Node manages the setup of nodes, creating genesis block, generating transactions, and setting apart the percentage of nodes to be validators from total nodes. Block owns block size, Hash created from the date-time-and block who number, validated transaction list, the id of Blockchain, pre block, and next block's hash. If it is not genesis then assign it the hash of the genesis block. While transaction owns transaction data, transaction owner, waiting time of the transaction to be validated, time at which transaction generated and time it got validated to calculate the latency of the network, id of transactions Blockchain, and some parameters to check if the transaction is valid, waiting or added to block. We have some variables which can be varied during simulation to check the effect on results such as the number of nodes, transaction rate, and percentage of validators in the network. By setting these variables to different values every time we can get different simulation results.

3.4 Network Setup

Firstly we create Nodes and genesis blocks by setting the percentage of required validators in the network divide the nodes in the network as validator and non- validator nodes. A number of validators in the network can be calculated by. $Validators = num\ of\ nodes * (\%validator * 0.01)$. Refer to Figure 3.2 flow chart for setup of Blockchain network. According to equations if we have a number of nodes 150, 50% validator nodes network has 75 validators.

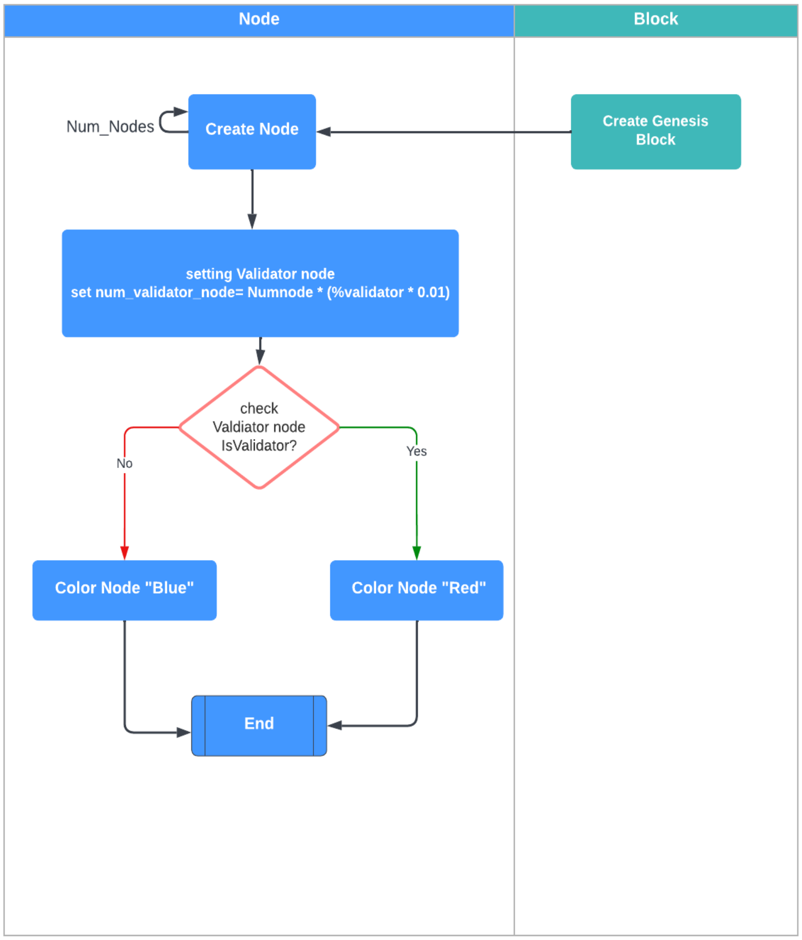
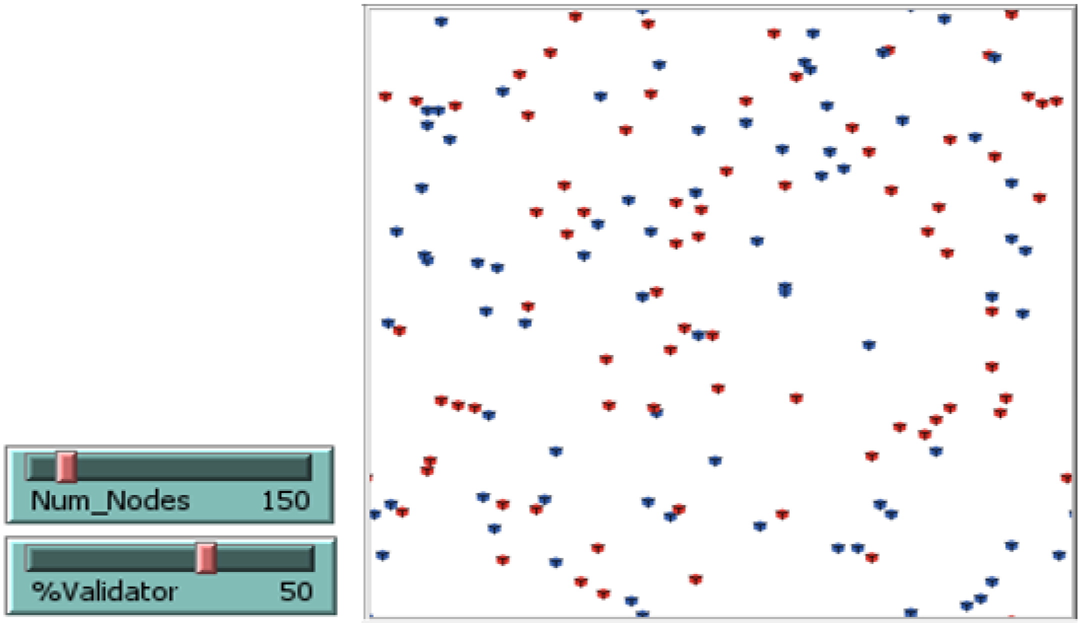


Figure 3.2 Network setup

Suppose we have Number of nodes 150, 50% validator nodes in network then distribution will be as follow Figure 4.



Num_Nodes 150

%Validator 50

Figure 3.3 Blockchain Setup blue nodes 150, Red %validator nodes 50

Num_nodes can also be altered with changing percentages of the validator. This is why we used ABM, it let us create a network to our own choice and simulated the environment to see different results every time.

3.5 Blockchain Simulation

After creating Nodes and genesis block now we will see how transactions are generated, processed and packed in a block. As a result we can measure the average delay faced by each transaction before getting blocked and appending to chain. See Figure 4 flow chart for Blockchain simulation.

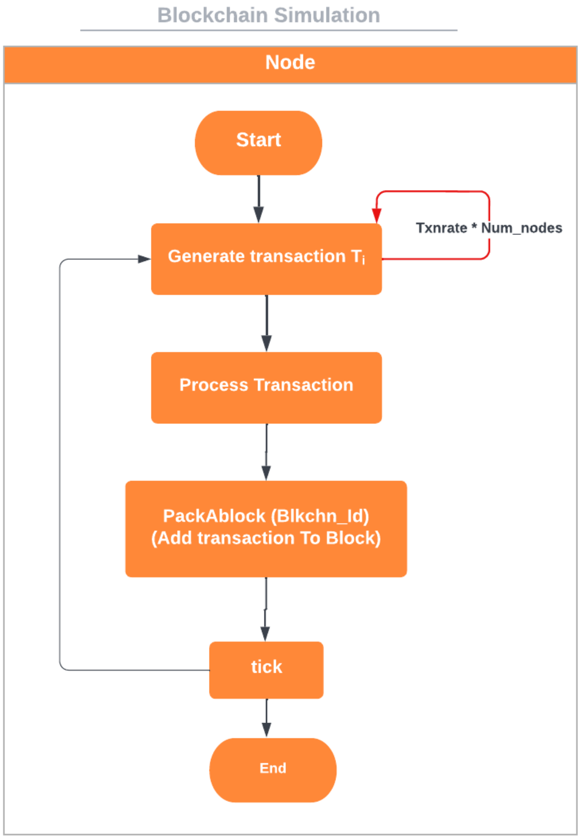


Figure 3.4 Blockchain simulation

For N number of nodes, NetLogo calls randomly each node to run operations iteratively. Tick is a built-in procedure that enhances each time procedure runs and ends when the user wants to end the function call otherwise can run every time the tick increases until stopped. As we can see there are 3 procedures called by a node to run the Blockchain, below is a stepwise explanation of each procedure.

a) Generate Transaction

It will generate random transactions based on the number of nodes and the transaction rate set in the network. Such as if we have 150 nodes and the transaction rate is 0.4 then according to $Num\_nodes * Trnxrate$ will be 60. So it will run operation random 60 to generate a transaction.   As we know NetLogo selects any random node from number of nodes to be a transaction generator and will create one transection and set its data.

b) Process Transaction

Check for transections awaiting to be validated. Setting the random delay for each transaction and the calculating sum of the average delay of each transaction that is packed in the block.

c) **PackAblock**

If the number of validated transactions is ready to be packed then make an ordered list of a transaction according to the pre-defined number of transactions that can be added. Keep the count of added transactions. Create block hash by concatenating date and time with id. Once the transactions are packed validator call all nodes to copy the packed transactions and the hash. Update the ordered list to all nodes, and add a new block to the existing Blockchain ledger of all nodes.

d) Validator nodes start working on the next block by using the previous block's hash to add to chain. Which shows that they have accepted the block.

After addition of a block to the chain we have measured its delay. Which is stochastically plotted in a section 3.5.1 along will detail discussion on changing transaction rate, number of nodes, and percentage of validator's effects on the throughput of the network.

3.5.1 Simulation Results

We will start with setting up the Blockchain interface with 3 sliders namely number of nodes, %validator (number of validators in network), Transaction rate can be varied from 0.1 to 1. (Generating transaction rate per sec). We will run the simulation for 100 ticks, with setting num_nodes 150, 50% validators and transaction rate 0.1. It will generate random transactions based on number of nodes and transaction rate set in network such that $Num\_nodes * Trnxrate$. For average delay measures there is auto scale plot having tick on x-axis and average delay on y-axis. Calculation of average delay is as follow

1) Sum of average delay = delay time of previous transaction + (end time of current transaction – start time of current transaction)

2) Count transactions = counter of packed transaction.

3) Average delay = (sum of average delay / count transaction)

These equations runs 5 times at each tick if number of transaction to be packed in block are 5.

<u>Changing transaction rate</u>

To analyze the effect of change in transaction rate on network latency, we will run the simulation for constant 100 ticks, number of nodes 150, 50% validators and varying transaction rate 0.1 and then for 1.

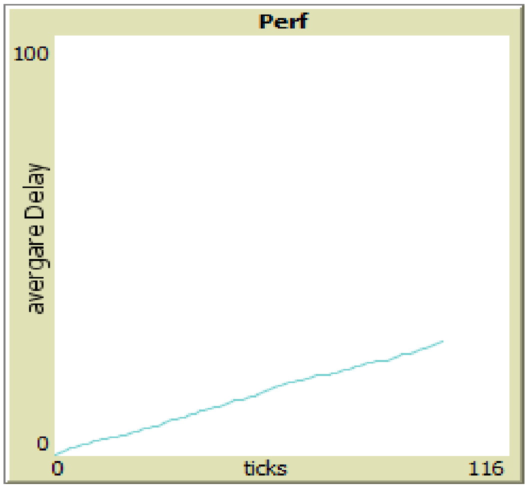


Figure 3.5 Average delay for T.R 1

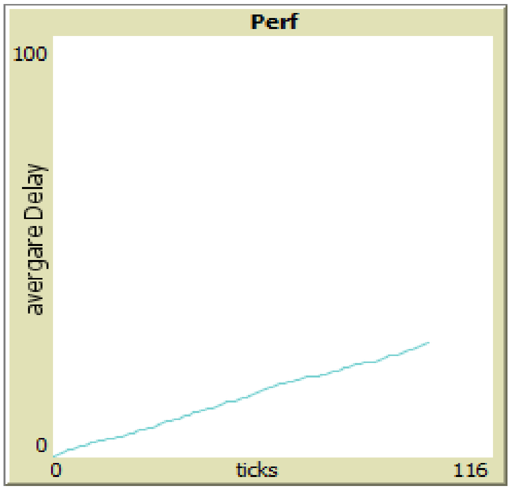


Figure 3.6 Average Delay for T.R 0.1

Transaction Rate	Average Delay
0.1	11.103
0.2	22.07
0.6	24.31
1	26.92

Table 3.2 Nodes 150, %validator 50%, transaction Rate variable 0.1-1.0

Keeping other parameters constant, change transaction rate 0.1 to 1 we can see from Figure 3.5, 3.6 and Table 3.2 that average delay is increased such that 0.1 = 11.103 and 1.0 = 26.92. Hence, we can say $Transaction\ rate \propto Average\ Delay$ such that transaction rate is directly proportional to average delay more transaction rate more average delay to add the block, less transaction rate less time to add a block also more transactions waiting to get validated because now validator are busy in validating the transactions.

Changing %validator

To analyze, how changing percentage of validators in network effects the latency? We will keep constant 150 nodes in network, transaction rate 0.2 and calculate average delay for 500 ticks. The Plots and Table have been used to display the results. Such that Table 3.4 list the results of average delay for changing %validator.

%Validator	**Average Delay**
30%	10.18
50%	9.44
80%	9.18

Table 3.2 Affect of change in %Validators on average dalay .

According to Table 3.4 increasing the percentage of validators can decrease the average delay to add a block to the chain. Hence, we can say $\%validator \propto \frac{1}{Average\ Delay}$ more number of validators, less average delay whereas fewer validators more average delay and more transactions waiting to be validated which adds to the latency of the network.

Changing Number of nodes

To analyze, how changing percentage number of nodes in network effects the latency? We will keep 40% validator nodes in network, transaction rate 0.3 constant and calculate average delay for 100 ticks.

Nodes	**Average Delay**
150	21.76
300	24.35
600	26.57

Table 3.3 Average delay for varying number of Nodes in network

According to Table 3.4, for a small number of nodes latency is less whereas for an increasing number of nodes latency of Blockchain increases. Also, the time to generate and add a block of transactions also increase which in turn gives a boost to network latency. Hence, we can say if we increase the transaction rate then $Number\ of\ nodes\ \propto Average\ Delay$ such that more number of nodes more average delay vice versa. We have to set the transaction rate low in such a huge network e.g. network having nodes 1000 and a transaction rate of 0.1 it will still result in an increased average delay with more transactions still waiting to be validated, delay in acceptance of transactions, and block addition to the chain. More the network gets bigger more latency problems are introduced for larger network transaction rate should be as lower as possible to milliseconds. So the size of the network plays a vital role in the latency of the network.

3.6 Conclusion

This chapter includes the introduction to Agent-Based Modeling, Blockchain architecture, simulation, stakeholders in our model, and results of the simulation. It concludes that the latency of Blockchain can be described as delay encountered by the network during the processing of the transaction to the addition of block to Blockchain. By Blockchain simulation AB modeling we have experimented with varying different parameters in the network such as the number of nodes, transaction rate, and percentage of validators to see the effect on latency of the network. Results have shown the pattern such that we can say if we increase the transaction rate then Number of nodes ∝Average Delay such that more number of nodes more average delay vice versa. Transaction rate and network size have direct proportionality with network latency whereas the percentage of the validator is inversely proportional to latency because if we have more validators then latency will be low and vice versa. Hence, if we want low latency then we need to have a balanced number of validators not less it will affect latency but not too many they will gain power in the network and could hack the network, and low transaction rate for big or small Blockchain. Lower latency can shape the Blockchain to move forward.

Chapter 4

4 BLOCKCHAIN FORK SIMULATION

The purpose of this chapter is to answer the research questions RQ 1: What is the effect on latency of the network when a fork takes place? And RQ4: when to fork? By simulating the fork model. It also discusses the effects of the fork on the latency of the original blockchain.

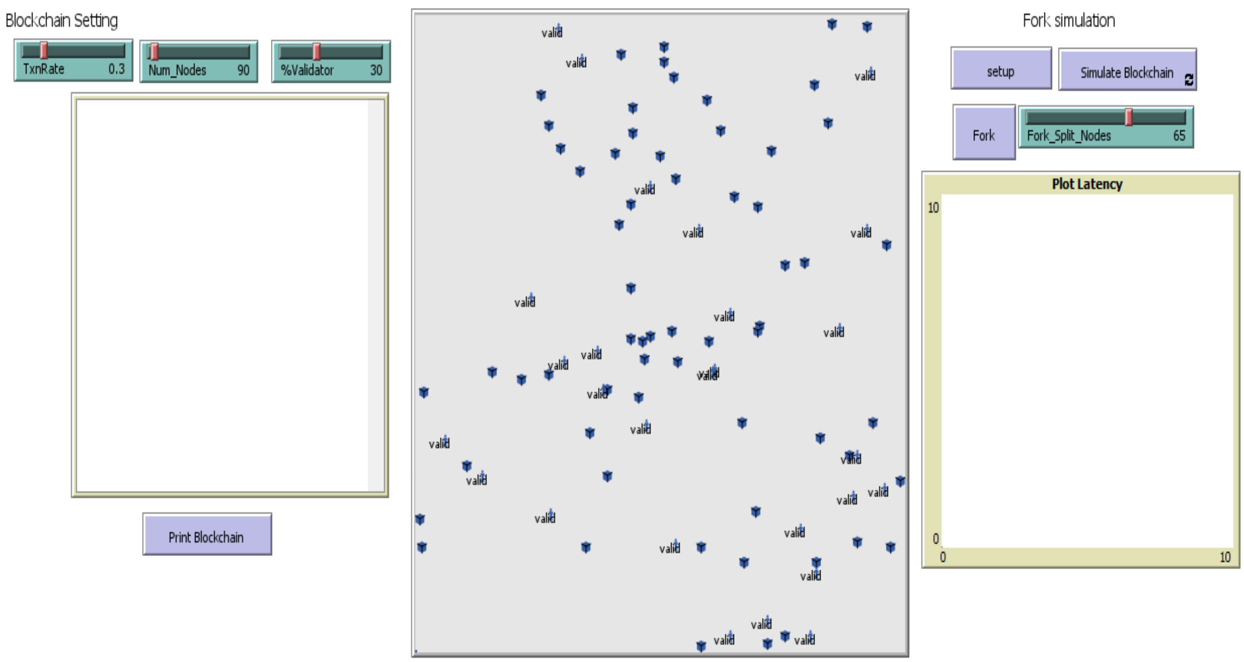
4.1 Introduction

In the previous chapter, we have seen how the block is broadcasted, and its related latency. To accept or reject a decision, stakeholders in the network can fork a chain at any instant causing a chain split known as a fork. A fork is defined as two or more routes to follow to satisfy the needs of stakeholders. After the fork, Blockchain has two ledgers with similar previous history but after the split, both parent and child chain maintains their ledger onwards. This causes latency issues due to the split of the network.

4.2 Blockchain fork simulation

We have already described the architecture of Blockchain in chapter 3. Now we will implement and simulate the fork chain by adding a variable node split threshold to observe the effect on the latency of the Blockchain. Firstly create Nodes and genesis block than by setting the percentage of required validators in the network we divide the nodes in the network as validator and non-validator nodes. Refer to Chapter 3, Figure 3.2 the flow diagram of the setup.

Figure 4.1 shows the interface for fork simulation such that a number of nodes are 90, %validator 30, transaction rate 0.3, %fork split nodes 65%. A number of validators in the network can be calculated by $Validators = num\ of\ nodes * (\ \%validator * 0.01)$ such that we have 45 validators in the current setting. Valid label agents represent the validator nodes while blue blocks are non-validator nodes.


Blockchain Setting

TxnRate 0.3

Num_Nodes 90

%Validator 30

Fork simulation

Fork_Split_Nodes 65

Figure 4.1 Fork simulation interface

While adding a block as defined in Chapter 3 often a delay occurs during the broadcast of the new block, at the same time other nodes may generate a new block before hearing about the recently broadcasted block this conflict leads to fork such that nodes in the network have multiple views of Blockchain ledger. Our research implements a fork to observe what happens fork occurs? Percentage of network joining forked chain in contrast to how much of percentage stays on the network? And the effect on latency when the fork happens.

4.1.1 Create Fork

Transactions are generated, processed, and packed in a block. As a result, we can measure the average delay faced by each transaction before getting blocked and appended to the chain. During simulation, we can create a fork at any instant by pressing the fork button. It calls the function to create a fork and split the network into chains according to %fork_split_nodes such that it will split the network into two. Figure 4.2 explains the fork phenomena. Each network has validator and non-validating nodes divided from the original Blockchain. Equations 4 and 5 split nodes between fork chains according to the percentage fork split threshold.

4 Validator for forked chain=(%Fork_split_node* 0.01* num_validNodes)

*5 non-validator nodes in forked chain=(Fork_Split_Nodes * 0.01 * (Num_Nodes - num_validNodes))*

After split transactions will be packed in a block to be broadcasted to its corresponding Blockchain. Each chain will have its own measure of latency.

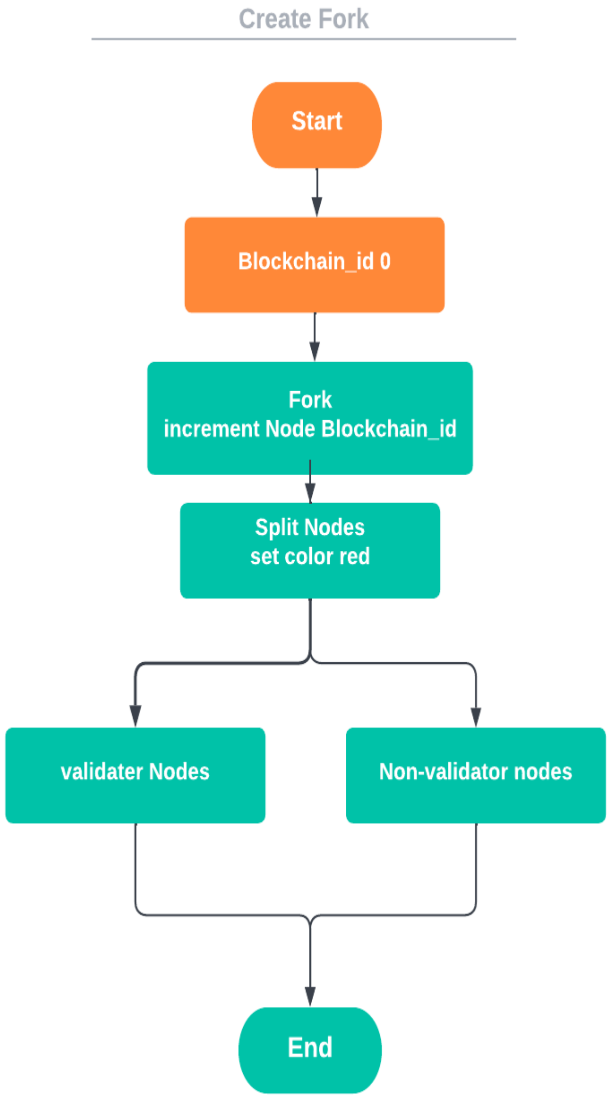
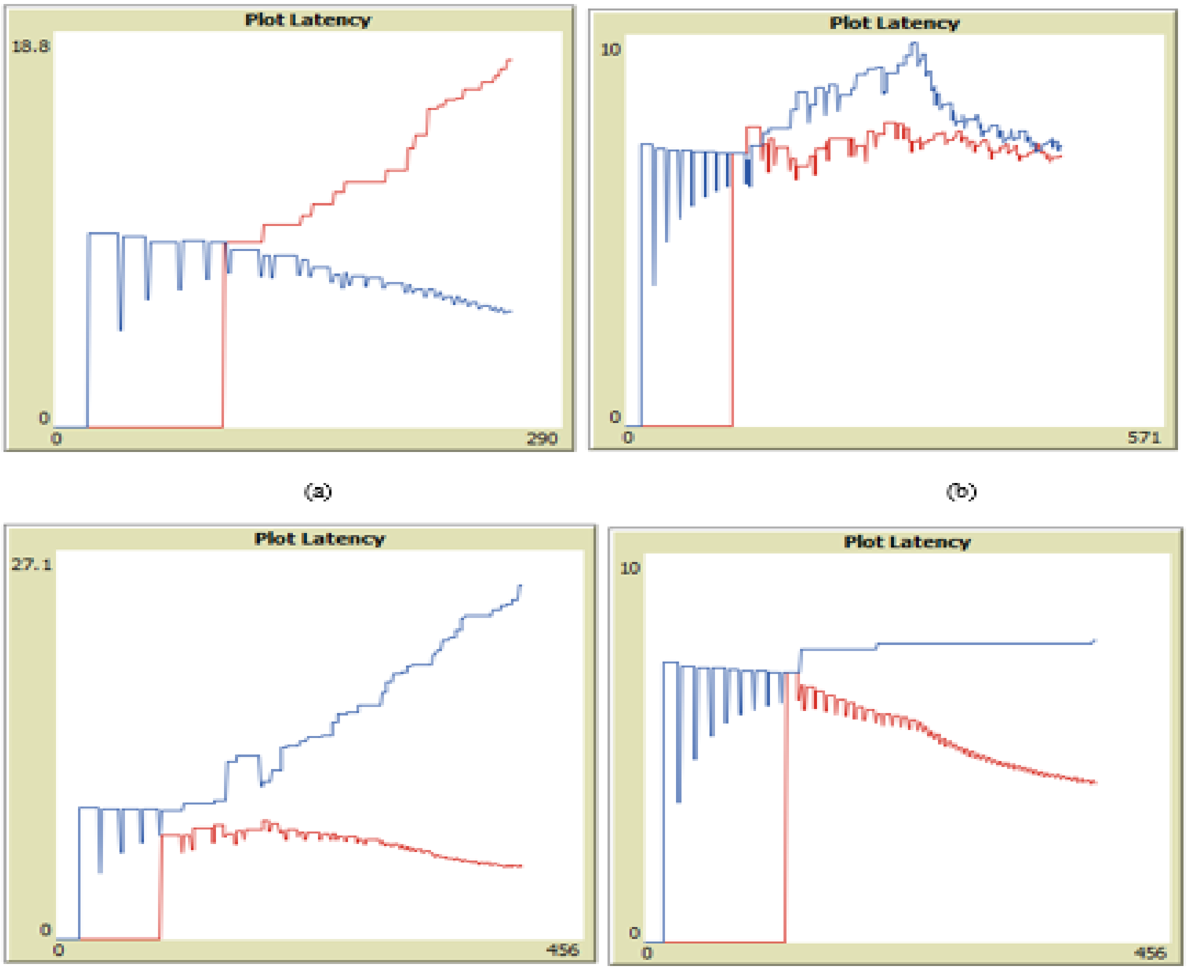


Figure 4.2 Create Fork

4.3 Results

We will experiment with Blockchain network having constant number of nodes 350, %validator 45, Transaction rate 0.3, and varying percentage of fork splitting nodes. Figure 4.3 shows the latency of two chains after fork where in plot blue line represents the latency of original chain, and red line shows latency of forked chain.



(a)

(b)

(c)

(d)

Figure 4.3 Latency plot for varying %Fork_Split_Nodes (a 20%, b50%, c75%, d 100%)

Now plotting these values in Table 4.1, before and after fork to see the relationship of latency for Blockchain.

%Fork_Split_Nodes	**Blockchain 0**			**Blockchain 1**		
	Non-validators	Validators	Latency	Non-validators	Validators	Latency
20%	280	140	4.44	70	35	15.4
50%	176	79	7.37	174	78	7.22
70%	88	39	24.6	262	118	4.99
100%	1	0	7.7 Abandoned after fork	349	157	4.08

Table 4.1 Latency effect before and after fork.

As you can see from Figure 4.3 and Table 4.1, when we increase the threshold of splitting nodes in Blockchain, latency increases in original and decreases in forked chain. By setting splitting nodes threshold as low as 20% then two chain splits happens, original chain have more non-validator and validator nodes while forked chain have less number of non-validator and validator nodes as a result latency of original Blockchain becomes high at start of fork then begins to decrees because it has more nodes to perform consensus, whereas latency of forked chain being small network at start is more but becomes more as time increases due to less number of nodes to perform transactions. Similarly, when we keep split threshold 100% whole community migrate to new forked chain abandoning original chain as a result no transactions occur on original chain. So its latency becomes to steady state whereas latency of forked chain continuous according to number of nodes and transaction rate it has. Hence, we can say $\%Fork\_split\_nodes \propto latency\_of\_orignal\_network$ and $\%Fork\_split\_nodes \propto 1/latency\_of\_forked\_network$ such that fork_split_nodes more latency more for original chain vice versa and fork_split_nodes more less network latency for forked chain.

4.4 Conclusion

In this chapter, we have done detailed analyses of how a fork happens, it's after effects, and the ratio of the split of the network. It also answered RQ 1: What is the effect on latency of the network when a fork takes place? By plotting, graphs and tables with variable node split threshold such that if fork_split_nodes is increased then resultant latency will also increase for the original chain and vice versa. While for the forked chain, an increase in fork_split_nodes leads to a decrease in network latency.

Chapter 5

5 PROPOSED DECISION-MAKING MODEL

The essence of Blockchain Governance is decision-making. So to answer RQ2: How to achieve network connectivity? RQ3: How a Blockchain will perform decision-making? This chapter proposes the voter model, its simulation, and results to perform pure decision-making.

5.1 Introduction

Blockchain does not have a pure decision-making process because it does not have network connectivity and communication among nodes. Studies are silent in this particular aspect so we have explored this issue and proposed the concept of network connectivity implemented through percolation meanwhile communication is done through a voter model having a different types of interactions for different scenarios. When we have an adequate level of connectivity, nodes having opinions will easily interact to influence each other as a result they will have a decision aligning the whole community at one opinion [40]. Such as fork or not to fork, accept or reject the proposal, etc.

5.2 Percolation Theory

Percolation theory was derived from solving a problem of how water flows over a porous stone? In what conditions water can flow from top to bottom of stone? What is the probability that water will reach the bottom of the stone? So, this is a phase transitioning process, a network of disconnected, small clusters becomes a large spanning cluster at a critical threshold [41]. Broadbent & Hammersley (1957), [42] introduced and described Bond percolation as a network of 3d connectivity such as in Figure 1, node N1 in the network is connected with a maximum of 3 neighbor nodes (N2, N5, N6) subsequently node2 is connected with 2 more nodes and so on to achieve percolation. As a result, information percolates across the network from many routes and no node will be left uninformed. By applying the same idea of percolation we set the Blockchain network in periodic lattice with Three degrees 3d (i.e. 3 neighbors) percolation having n x n x n vertices, and edges. Vertices are known as “site” and edges are “bond”, dimensions d together with the value of p satisfying $0 \leq p \leq 1$ such that the bond between two nodes might be open with probability p, or it could be closed with probability 1-p, Hence, for given p, what is the probability that there exists an open path from one node to another node? Probability of connecting with a neighboring node or not depends on the percolation threshold defined as [4] “creation of large clusters or long-range connectivity. If the critical threshold pc (i.e. pc ≥ 0) is low small network clusters will be formed having low communication; while if pc (i.e. pc≤1) is above the threshold, it will form a large cluster that connects all or most of the network. Now there are two ways to implement percolation a) site percolation and b) bond percolation. We are using bond percolation see Figure 5.1 by declaring the edges of the node to be open or close with ½ probability for each state.

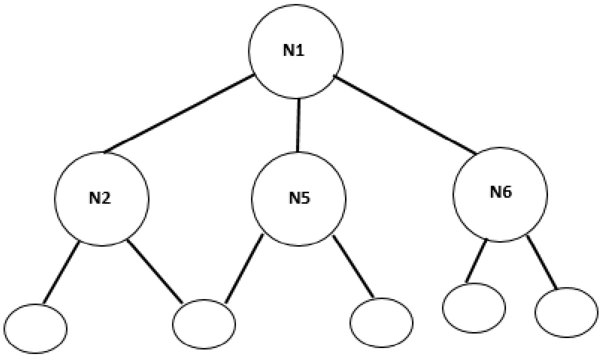


Figure 5.1 Node connectivity

5.2.1 Percolation Simulation

To implement network connectivity we are using agent-based simulation. In NetLogo firstly, we defined a turtle breed of nodes. Each turtle node agent owns my_neighbors, num_neighbors, connected_neighbor, and isConnected? Variables. My_neighbors stores the neighboring agent for each node, num_neighbors stores the number of neighbors (i.e. 0, 1, 2, or 3), connected_neighbor Returns connected neighbors based on percolation threshold (i.e. pc = 0.5), isConnected? Returns true if all 3 edges are connected and false if one of the 3 edges is close or has no neighbor. NetLogo executes the code iteratively for randomly selected node agents until every node runs the command. The algorithm for creating neighbors is as follows.

Algorithm 1: An outline of the forming 3d neighbor

```
Create n_Nodes
```

```
create maximum of 3 neighbors for each node
set agentset  my_neighbor = list of neighboring node agents
count num_neighbor from agentset my_neighbor excluding myself
end
```

Algorithm 2: An outline of the Bond Percolation Connectivity

```
node with num_neighbor > 0
set   temp_NeighList n-of (round (random ( Percolation * num-neighbors + 1))) my-neighbors //depending on percolation threshold with how many nodes connection is established and saved in new agentset
if (connected_nighbour != nobody ) and (temp_NeighList != nobody)
        Assign temporary neighbor list to connected neighbor excluding node itself.
Else
        connected_nighbour =  nobody
end
```

The number of neighbors can be 0 to 3. This is 3d bond percolation with pc varying from 0 to 1 such that the threshold decreases with an increasing number of neighbors. Each bond between nodes can be true (open) with probability p and False (close) with probability 1-p. Starting from any node you can reach to the edge has 2 conditions a) if all nodes are connected and open then we will have a fully dense graph i.e. $pc \leq 1$. b) If nodes are not fully connected to reach the edge then small clusters are formed i.e. $pc \geq 0$.

5.2.2 Results

In this section we will see how dense the network is connected by changing the percolation. Let's keep number of nodes 130. It can also be varied but we will analyze percolation in network of 130 nodes.

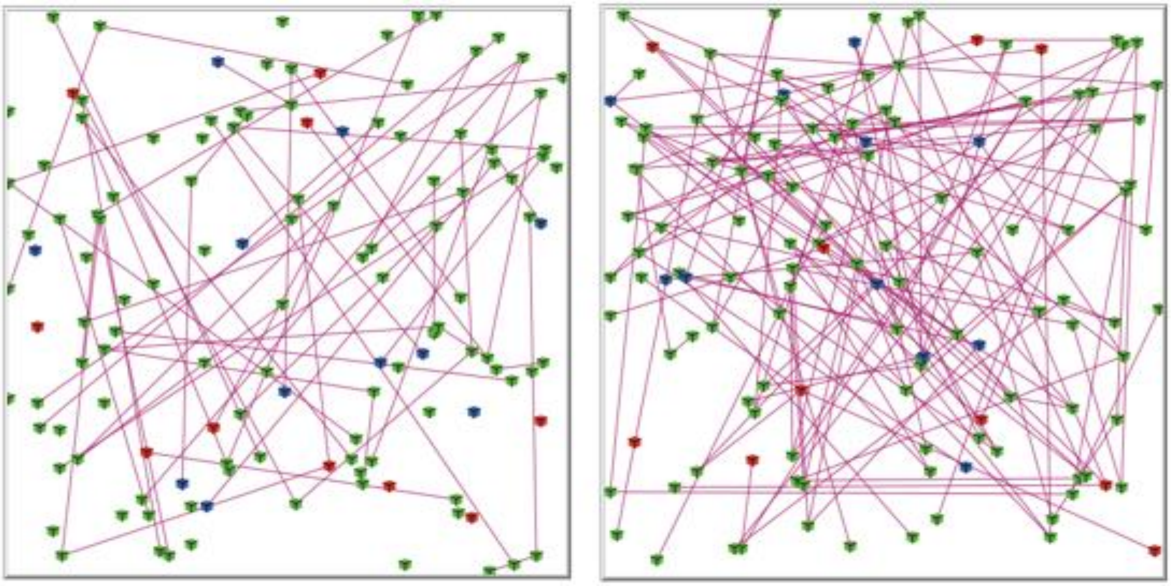


Figure 5.2 For Pc = 0.1 and Pc = 1

Result in Figure 5.5 shows how with increasing percolation we get densely connected graph. There is a visible difference in Percolation 0.1 and 1.0 connectivity. Answering RQ2: How to achieve network connectivity? Hence, $percolation \propto connectivity$ we can say less percolation threshold less connectivity, more percolation threshold more densely connected graph.

5.3 Voter Model

Social network links people together which affects human behavior by getting influenced by the opinion of the individual. Richard A. Holley and Thomas M. Liggett in 1975 [45] introduced the voter model as a system of an interacting particle with probably of p. voters coexist on a voter model graph with two or more clusters depending on the number of opinions in the network. In a connected graph at each point, there is a "voter" connected with another voter establishing the connection to interact with each other. The opinion changes at the random time given by voters on some issue under the influence of interacting with neighbor voters. The opinion is one of two states 0 or 1, a voter can have at a given time. The opinion of the voter is changed according to stochastic rules when a voter is randomly selected at a random time. For a randomly selected voter, one of its neighbors is chosen according to percolation to adopt the opinion of the randomly selected neighbor. Various models explore the social behavior of agents in a network having an influencing opinion such as the Name Game (NG) [46] Model works on 2 opinions also known as the binary agreement model. The q-vote [47] and AB model [48] where several nodes having the same opinion bring out the change. In the noise reduction model [49] voter keeps track of total nodes aligned toward changing opinion, but a change of opinion can only happen when this counter of nodes holding change opinion reaches a certain threshold. We are going to use an influencer mechanism similar to the models listed above, such that an individual agent adopts a particular opinion state after getting influenced by a socially connected neighbor.

5.3.1 Voter Model implementation strategies

In this research, we have studied how the positive or negative influence of one node can affect the dynamics of the system. As it is difficult to solve the voter model analytically so we are using agent-based modeling to simulate the interactions among nodes of the network. This voter model has N node agents, each of which can hold 1 of 3 possible states such that positive influencer, negative influencer, and unsure agent. In an update event, a random node is selected which adopts the state of one of its randomly selected neighbors. After a certain number of iterations, a cluster is formed showing a majority of agents having a similar opinion at a time t. we aim to align the community on one decision. Now upon interaction we set following the rules:

- A positive influencer changes an opinion of a negative agent upon interaction to unsure.
- A Negative influencer changes the opinion of a positive agent to unsure.
- A positive influencer changes the opinion of an unsure agent to a positive and vice versa.

5.3.1.1 Neutral Interaction

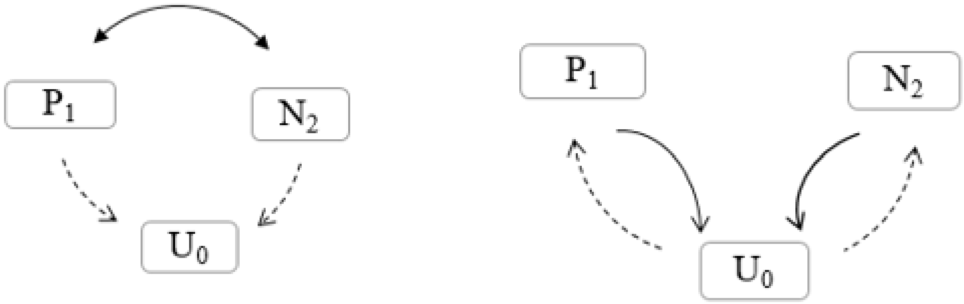


Figure 5.3 Neutral opinion startagy illustrates the transition of state after interaction. Solid line represent influencer agent's interaction with neighboring agent having specific opinion. Dashed line represent resultant opinion change of neighboring agent after

Table 5.1 below lists possible outcomes when agents interact with each other such that for random selection of influencer- neighbor pair updates the state of the neighbor. In the Left column before interaction, we have listed influencer first and neighbor second. The second column shows the updated state of influencer-neighbor after the interaction.

Before Interaction	After Interaction
P_1N_2	P_1U_0
P_1U_0	P_1P_1
P_1P_1	P_1P_1
N_2P_1	N_2U_0
N_2U_0	N_2N_2
N_2N_2	N_2N_2

Table 5.1 Neutral Interaction

So we can observe from Table 5.1 that when opposite opinions interact it changes the state of the neighbor to unsure. Whereas when positive or negative opinion interacts with unsure it changes the state of the neighbor to interacting influencer's state. In such interactions unsure agent that changes state can be seen as a "flip-flopper or confused" this type of agent remains uncertain of its new state and can switch its state back and forth.

5.3.1.2 Extremal Interaction

The second type of interaction Figure 5.4 we modeled is an extremal version in which when a positive agent interacts with a negative agent it changes its state to positive and vice versa. In this type of state, the change agent becomes fully committed to its new state. Whereas interaction with an unsure agent remains the same as defined in neutral interaction.

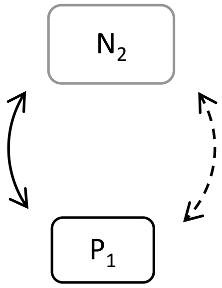


Figure 5.4 Extremal opinion startagy illustrates the transition of state after interaction. Solid line represent influencer agent's interaction with neighboring agent. While dashed line represent resultant opinion change of neighboring agent after interaction.

Before Interaction	After Interaction
P_1N_2	P_1P_1
P_1U_0	P_1P_1
P_1P_1	P_1P_1
N_2P_1	N_2N_2
N_2U_0	N_2N_2
N_2N_2	N_2N_2

Table 5.2 Extremal Interaction

so we can observe from first and third row of Table 5.2 that when opposite opinions interact it changes state of neighbor to strongly negative or positive based on influencing agent's state. We can see such agent as "confident" about its state. Whereas when positive or negative opinion interact with unsure it change state of neighbor to interacting influencer's state.

5.3.1.3 Fixed point Interaction

The third type of interaction Figure 5.5 caters the people who are stubborn and do not changes their opinion upon interaction with positive and negative agents.

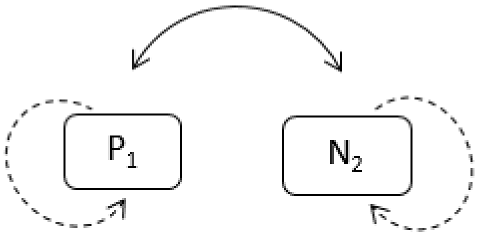


Figure 5.5 Fixed opinion startagy illustrates the transition of state after interaction. Solid line represent influencer agent's interaction with neighboring agent. While dashed line represent resultant opinion change of neighboring agent after interaction.

Before Interaction	After Interaction
P_1N_2	P_1N_2
P_1U_0	P_1P_1
N_2P_1	N_2P_1
N_2U_0	N_2N_2

Table 5.3 Fixed point interaction

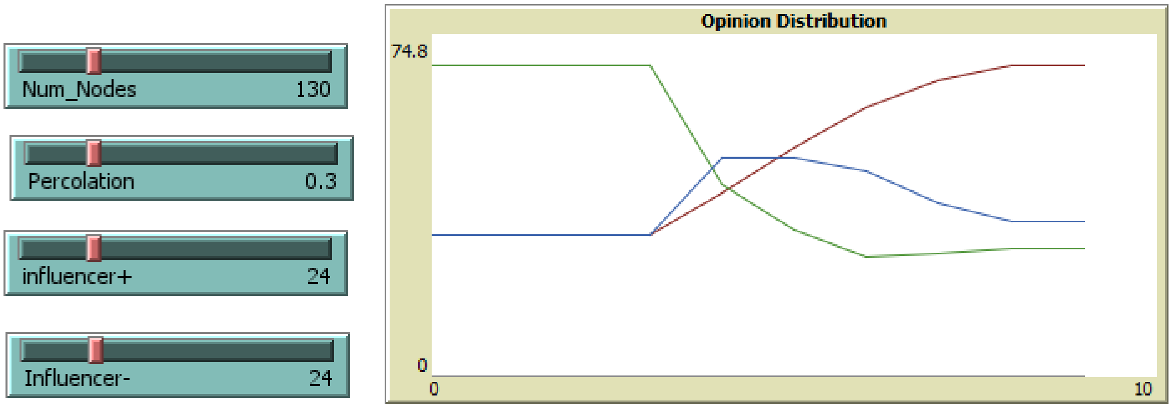
The first and third row of Table 5.3 shows when opposite opinions interact in a fixed strategy state of the neighbor remains the same representing the stubbornness phenomena. Can also be called inflexible agent interaction. Whereas when positive or negative opinion interacts with unsure it changes the state of the neighbor to interacting influencer's state.

5.4 Voter Model Simulation Results

As we know it is difficult to solve the voter model analytically so by using agent-based modeling we have simulated the model. In this section, we will show the results of listed interaction strategies by changing parameters and will analyze the variations in the results.

5.4.1 Neutral Interaction

To study neutral interaction, suppose we have 130 nodes in the network having percolation P = 0.3. Setting the network with 50% unsure agents and 50% influencers. Influencers are further divided into 50% positive influencers and 50% negative influencers.



Num_Nodes 130
Percolation 0.3
influencer+ 24
Influencer- 24

Figure 5.6 Neutral interaction for equal +ve, -ve influencers

We have used a stochastic plotting scheme to see the distribution of the agent's opinion according to time. The x-axis gives time while the y-axis shows the group of agents having a specific opinion. At the start, we can see both positive and negative agents are equal in number i.e. positive (Red line), negative (Blue line) agents, and remaining unsure (Green line) agents. When the simulation starts we can see the sudden change of states due to a full 3d dense connection with percolation 0.3. After the interaction, we can see a reduction in unsure agents and –ve influencers while +ve influencers increases.

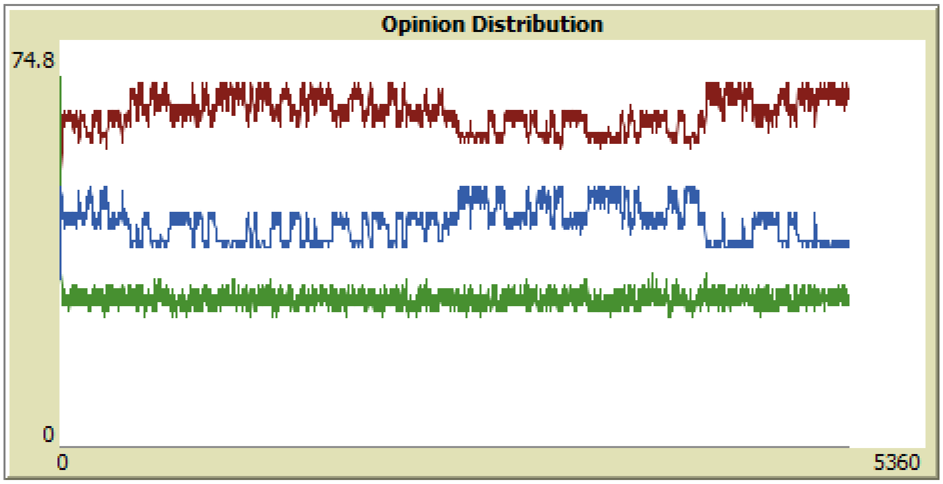


Figure 5.7 Neutral interaction opinion distribution

After some time it will achieve this steady state such that network now have more +ve influencer than -ve and unsure agents it will remain in this state until we make connection more dense for nodes to communicate more. To have a complete steady state where +ve influence overcomes the network by increasing the connectivity of network we have following results.

As number of agents are equal there is also an equal probability of –ve agents to overcome the network by –ve influence. Figure5.8 shows results of –ve agent majority.

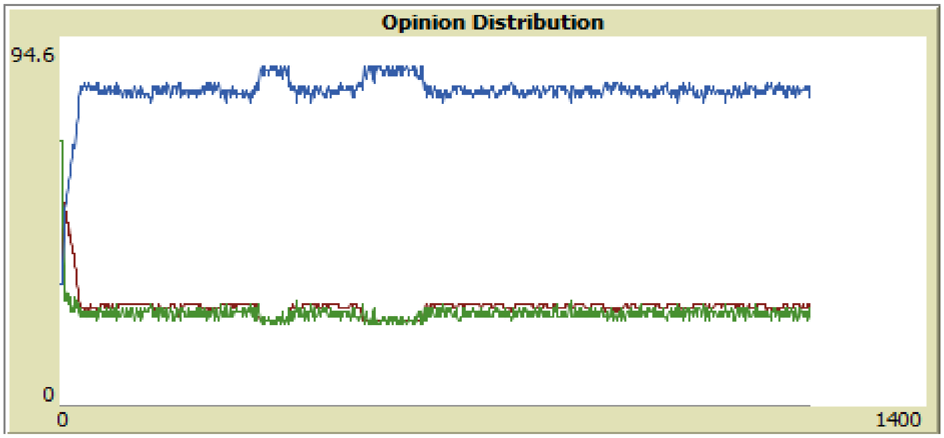


Figure 5.8 Neutral interaction opinion distribution for –ve agent majority

Now I will show final results by varying percolation from 0.8 in equal number of +ve and –ve influencers. See Figure network achieved steady state more quickly due to strong communication.

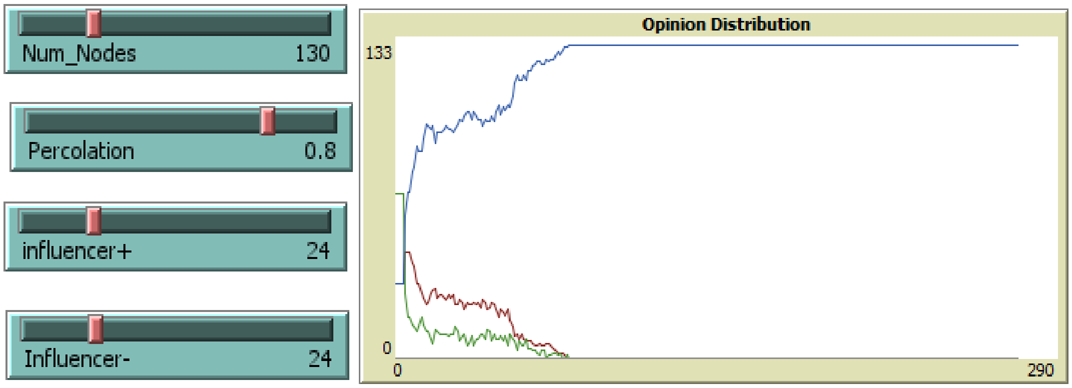

Num_Nodes 130
Percolation 0.8
influencer+ 24
Influencer- 24

Figure 5.9 Neutral interaction opinion distribution for high percolation

Now if we can observe that setting only one influencer more in +ve state then -ve state with 0.3 percolation effects the result such that +ve and –ve influencer are in flip floper state changing back and forth. If we establish connection one more time can notice the positive majority network from a Figure5.10 below.

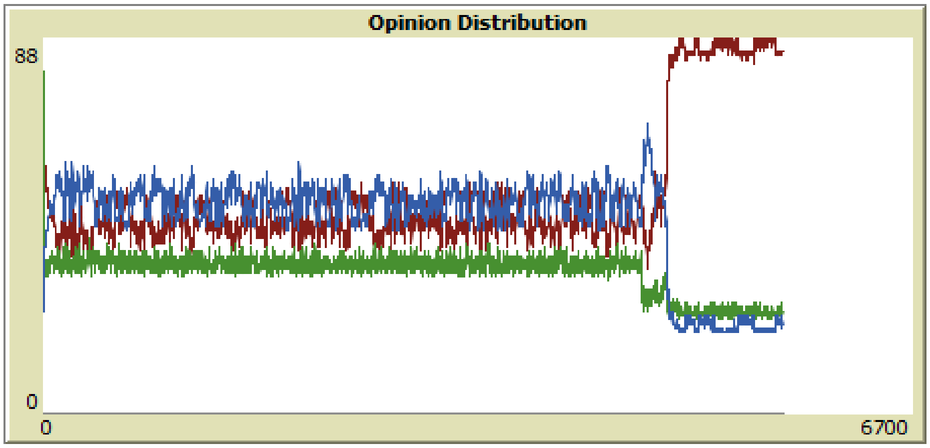


Figure 5.10 Neutral interaction opinion distribution for one +ve influncer more with p 0.3, increased connectivity

Hence, from all the simulation results we can see how neutral interactions take place such that by only increasing 1 +ve agent it can grasp the majority of the network. We can vary different variables such as nodes, percolation, and influencers to explore more.

5.4.2 Extremal Interaction

To analyze Extremal interaction, suppose we have 130 nodes in the network connected with 3 degree percolation P = 0.3. Setting the network into 50% unsure agents and 50% influencers among those influencers we have further divided them into 50% positive influencer and 50% negative influencers.

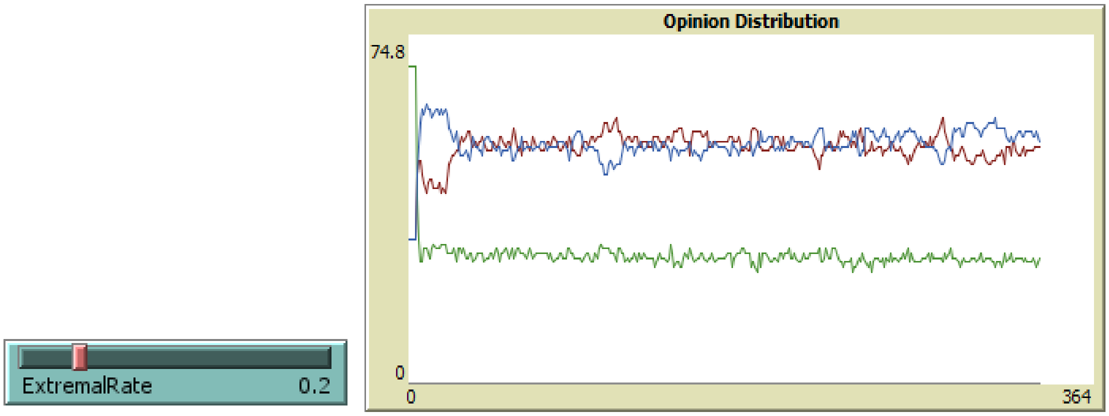

ExtremalRate 0.2

Figure 5.11 Extremal Interaction with equal influencers

As the number of influencers are equal we can see in Figure 5.11 the plot of extremal interaction setting it to 0.2. According to the rule when a positive influencer interacts with of negative influencer its state changes to positive and vice versa that's why we can see how blue and green lines are changing states. Hence, we have a less unsure agent and nearly equal +ve and –ve agents in a steady state. In another case, if we change the extremal rate to 0.5 results will change. Figure 5.12 below shows the change in the influence of the opposite node is stronger in the network now.

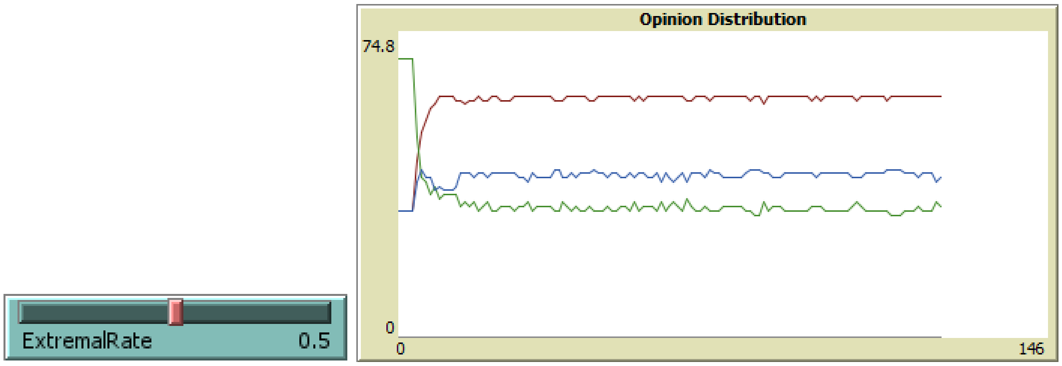

ExtremalRate 0.5

Figure 5.12 Extremal Interaction with equal influencers, extremal rate 0.3

The above-described cases were for an equal number of agents. Now we take another case where we increase one –ve influencer with an extremal rate of 0.6. Such that –ve influ set to 25 and +ve to 24. After simulation, we either get a positive majority or in the second case negative majority. Refer to the Figure 5.13 below:

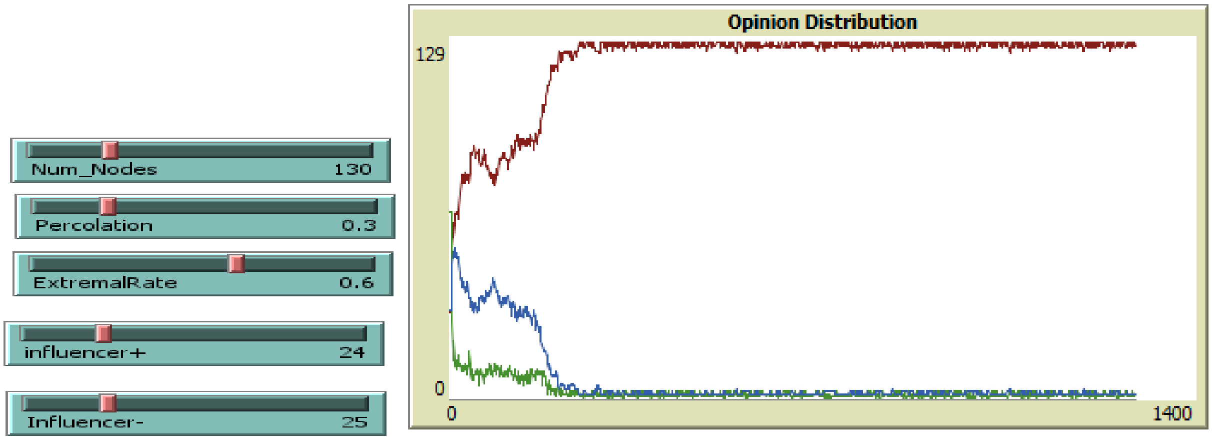

Num_Nodes 130
Percolation 0.3
ExtremalRate 0.6
influencer+ 24
Influencer- 25

Figure 5.13 Extremal Interaction with more –ve number of influencers

By increasing, percolation networks' opinions will reach to ideal state more quickly. Hence, we have seen how increasing and decreasing extremal rate and influencers affect the network division. If the network has equal influencers and the extremal rate is low then a division of network opinion is equal. While by increasing the extremal rate makes one of two opinions gain a majority of the network. Whereas by keeping the extremal rate constant and varying influencers ratio we noticed minority gaining majority due to extremal rule and nodes switch from positive to negative or form negative to positive very strongly.

5.4.3 Fixed point Interaction

For observing fixed agents, who do not change their state and stay stubborn. Suppose we have 130 nodes in the network connected with 3d percolation P = 0.3. Setting the network into 50% unsure agents and 50% influencers. We have introduced a fixed point switch when ON it allows the network to have fixed +ve and –ve agents who do not change opinion upon interaction. Figure 5.14 shows at the start of connectivity agents changed state by interacting with unsure agents after that agents who become +ve or –ve stayed persistent with their state.

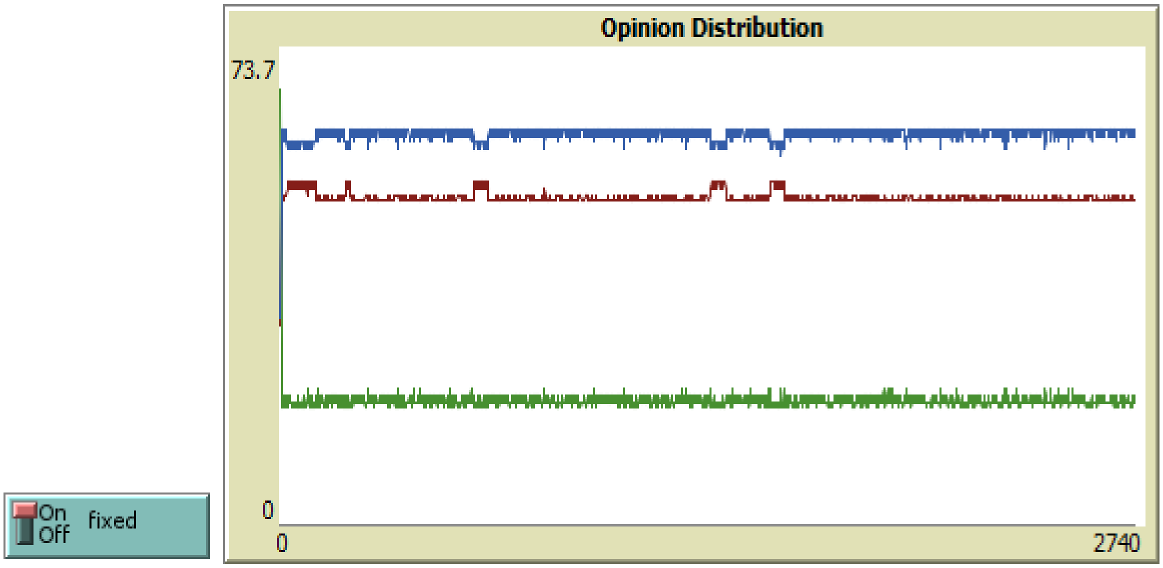


Figure 5.14 Fixed point Interaction with equal influencers.

Figure 5.15 shows plot for increasing –ve influencer in fixed interaction same is true for increasing +ve influence.

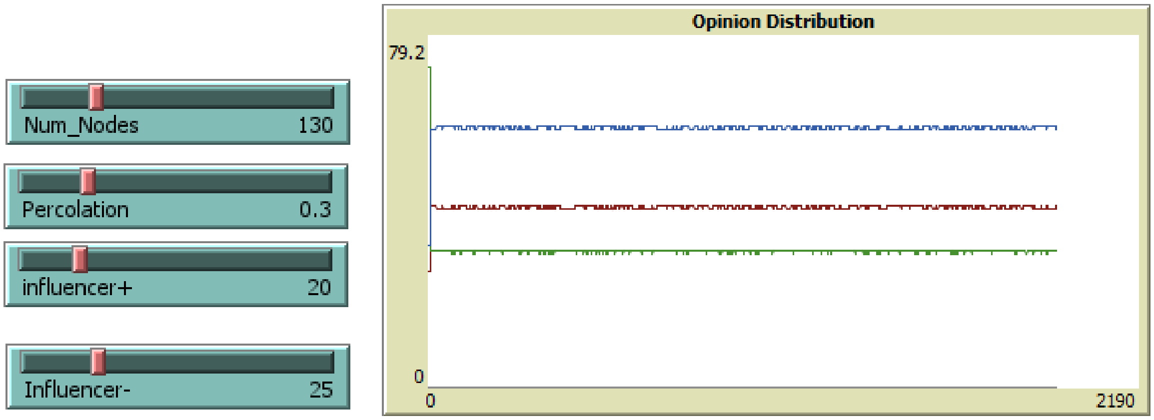


Figure 5.15 Fixed Interaction with more -ve influencers

We can see that in fixed agent interaction model nodes don't change their opinion upon interaction with opposite opinion node as a result opinion having higher number of influencer will take the lead.

We have simulated three types of interactions that define the behavior of agents confused, confident, and stubborn. The confused agent is modeled in neutral interaction where the agent keeps on switching its state from +ve or –ve to unsure until one of the opinions reaches gaining majority nodes. Whereas in the confident models the extreme state change and stays in that state. Such that minority can gain a majority of network with random confident agents. The third stubborn agent simulates the interaction of agents having fixed opinions that on interaction do not change as a result initial majority wins. These models are solved by varying different parameters for observing the results.

5.5 Conclusion

In this chapter, we have introduced our proposed model, implementation, and results. Blockchain does not have a pure decision-making process because it does not have network connectivity and communication among nodes. To solve these issues we provided a solution to RQ2: How to achieve network connectivity? Done by percolation such that $percolation \propto connectivity$ we can say less percolation threshold less connectivity, more percolation threshold more densely connected graph. And RQ3: How a Blockchain will perform decision-making? Solved through influencer voter model with three types of interactions i.e. neutral, extremal, and fixed interaction. The voter model has agents having positive, negative, and unsure agents. Upon simulation, these agents interact with each other and adopt the opinion of the influencer. The model reaches a steady state when the entire network ends up holding the same opinion or it does not further change opinion with time. Based on these interactions and connectivity we have analyzed how people will behave when a new proposal is introduced in the Blockchain community. The behavior spread faster and beyond across well-connected large cluster networks than across random, poorly clustered ones.

Chapter 6

6 A Model for Optimal Decision-making using Game Theory

In this chapter, we aim to achieve an answer for RQ4: when to fork? By evaluating how the decision is made in our system. Blockchain amalgamates four concepts: fork, percolation, voter model, and game theory. We have proposed an agent-based simulation to run a Blockchain environment, perform decision-making and observe the effect of a fork on the network.

6.1 Introduction

We noticed during research that no one talked about network connectivity and communication of the network during the proposed change. Studies are present to improve Blockchain network security, delays, and throughput but little to no light has been shed on network connectivity such as protocol for nodes communication, and flow of information in a distributed environment in such a way that it reaches every stakeholder of the system to have a pure decision in the well-connected network. Studies are silent in this particular aspect so we are proposing ON-chain decision-making ABM simulation to model the behavior of Blockchain. A framework that implements the concept of network connectivity through percolation meanwhile communication is done through the influencer voter model considering different types of interactions for different scenarios. To shape the opinion of a network with an adequate level of connectivity, users with a specific opinion will interact with other users, subsequently changing the opinion of interacting users to align the community on similar opinions. Based on these interactions and connectivity we can analyze how people will behave when the new proposal comes into the Blockchain community. The behavior spread faster and beyond across well-connected large cluster networks than across random, poorly clustered ones. To have instant decision-making, we have applied game theory, which decides whether to fork or not by analyzing the current state of Blockchain. Hence, when the network decides to fork everyone in the network will take part and give their vote to maximize their pay-off as well as the network's goal. Opinion having a majority will take over the network and it will carry on adding blocks on the same Blockchain with lower latency. While minority who forked away will have less number of nodes and high network latency. Our proposal is unique in the sense that no one has used percolation, voter model, and game theory together for Blockchain governance.

6.2 Simulation and results

In this module, we have connected all three above simulation models to view the full process of decision-making and the effect on latency of the original Blockchain. Firstly we will set up Blockchain, then set the desired level of connectivity. Afterward, set the opinions distribution as required among nodes based on the ratio of influencers to unsure agents in the network and percolation. The interface of the model after establishing the network with the following setup: 90 nodes, percentage of validators 51%, transaction rate 0.5, percolation 0.3, positive influencer 8%, and negative influencer 14%. As shown in Figure 6.1. The opinion distribution plot shows the number of +ve influencers with a red line, -ve influencers blue line, and unsure agents with a green line. Whereas, plots are for the latency of Blockchain network.

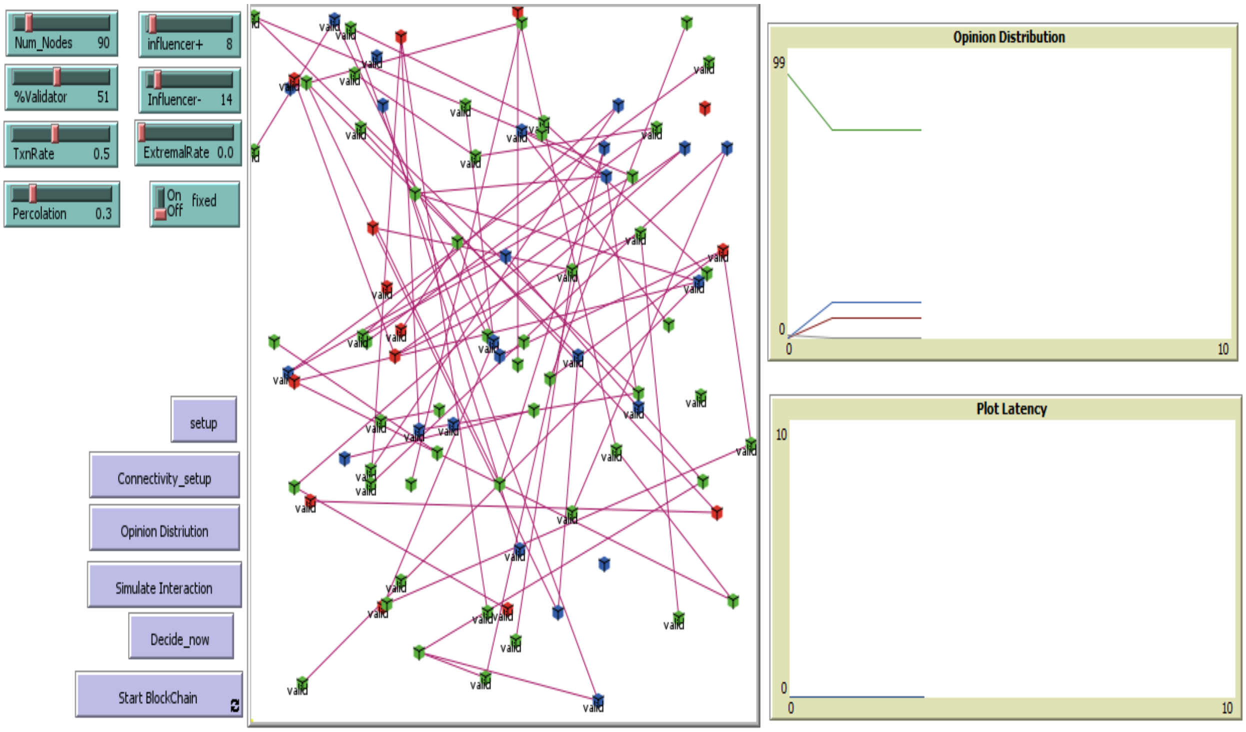

Num_Nodes 90
%Validator 51
TxnRate 0.5
Percolation 0.3
influencer+ 8
Influencer- 14
ExtremalRate 0.0
On Off fixed

Figure 6.1 Blockchain decision-making Setup

When we will start Blockchain concurrently simulating the interaction between nodes of the network has more negative influence than positive. Keep on simulating interaction until it becomes to steady state and doesn’t change its state on more interaction as shown in Figure 6.2 plot. Negative influence more after the interaction.

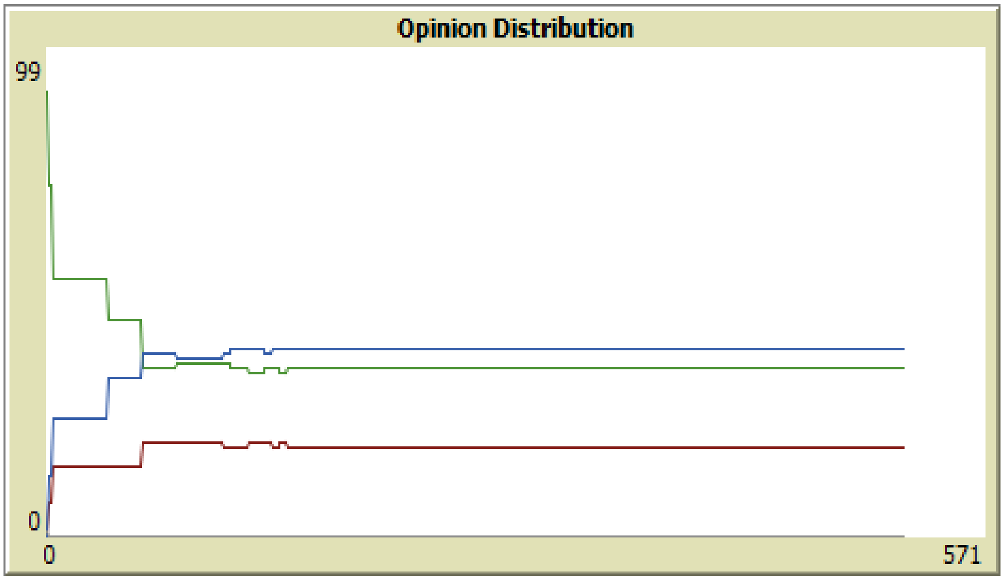


Figure 6.2 Majority of negative influencer after interaction

At a steady state, we decide to fork the original Blockchain as a result negative influencers will make a new Blockchain. Now if we observe the latency of the original Blockchain (blue line) in Figure 6.3 before the fork latency was 8.33 as soon as the fork happens the latency of the original Blockchain increases to 12.25 due to the split of the network to the forked chain. Whereas the forked chain (red line) has a high frequency at the start and then starts to decrease but the latency of the forked chain is much higher than the original chain due to less number of nodes and validators to process transactions.

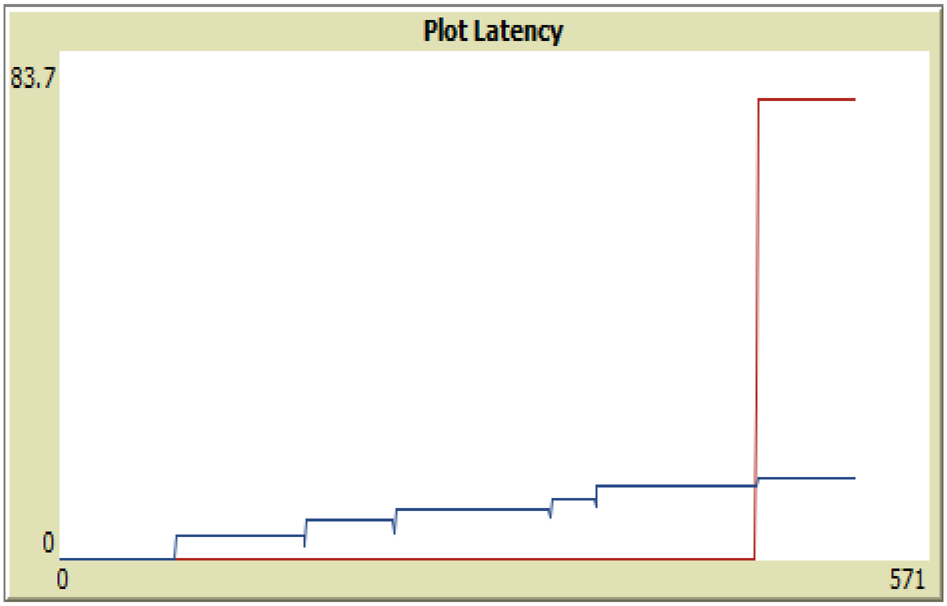


Figure 6.3 Latency plot after fork

With time we can observe the changes in latency of original chain which keeps on increases at a steady rate. Whereas forked chain shows rapid change in latency due to low transaction rate, more time to validate and add block to chain. Figure 6.3

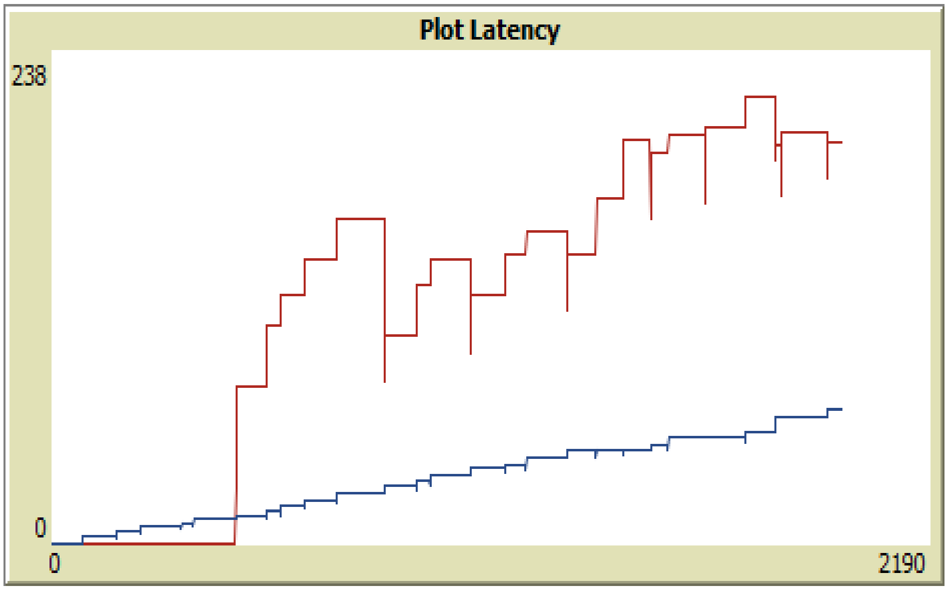


Figure 6.4 Effect on latency of original chain and forked –ve influencer's chain

In the second case if we change the ratio of influencers by increasing +ve influencers and decreasing –ve influencers in the network. We can see in Figures 6.4 and 6.5 a sharp rise in the latency of negatively influenced forked chain. Because it has a lesser number of nodes having –ve opinion hence more latency. Whereas the original chain is also affected by node split with increased latency but comparatively less than the forked chain due to more nodes in the network.

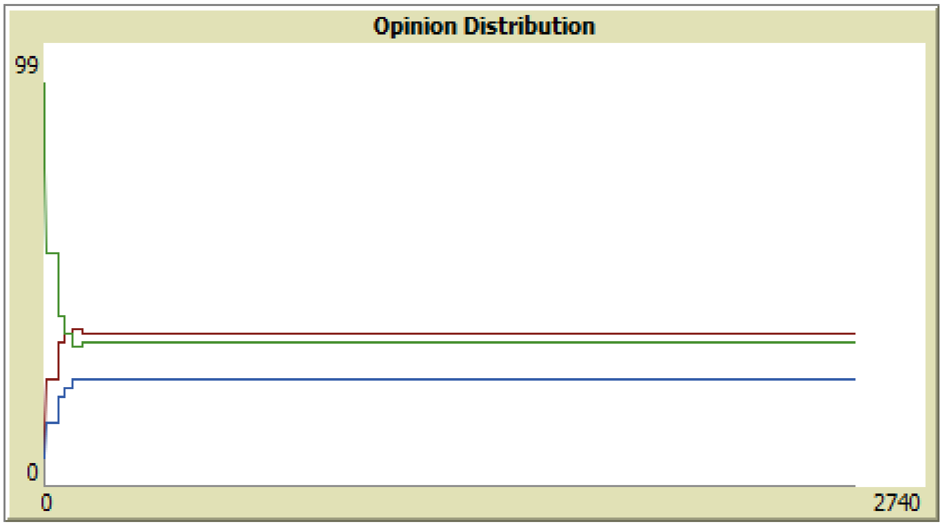


Figure 6.5 Majority of positive influencer after interaction

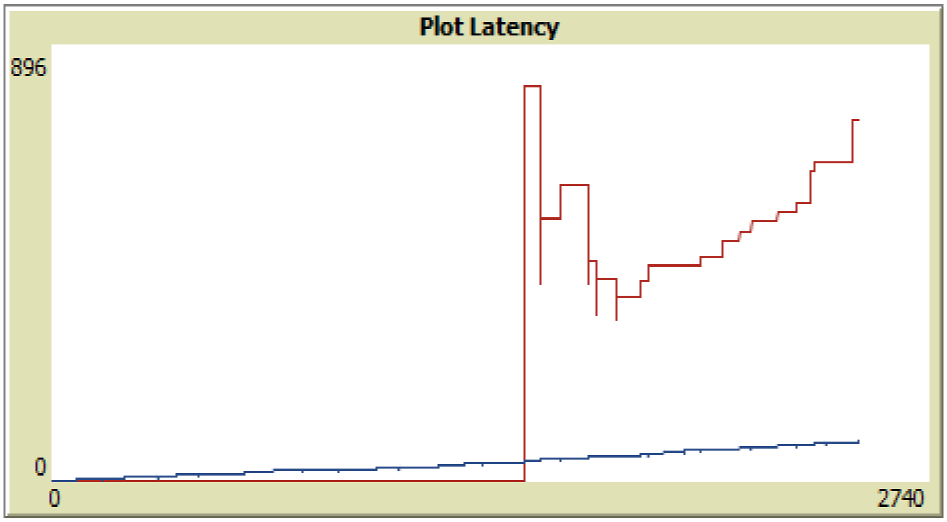


Figure 6.6 Effect on latency of original chain and forked –ve influencer's chain.

Let's take a case of fixed interaction in network having 150 nodes, percentage of validators 40%, transaction rate 0.3, percolation 0.3, negative influence 59% and +ve influence is 16% for certain proposal. Figure 6.8 shows majority of negative influencer who remain negative upon fixed interaction.

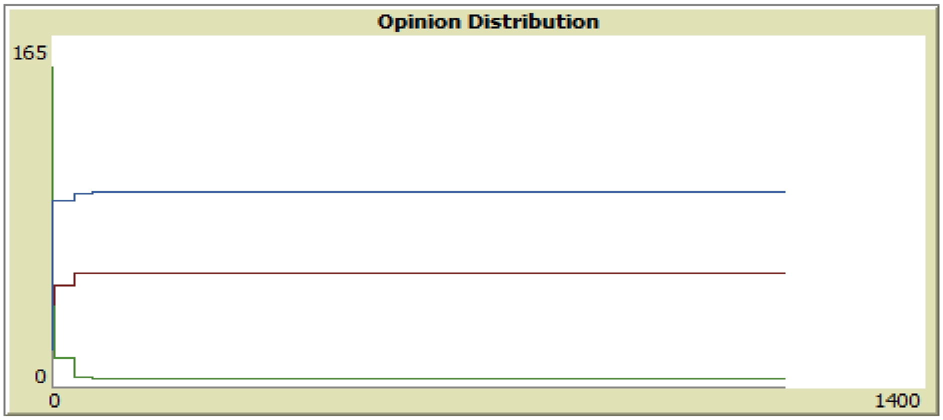


Figure 6.7 Majority of -ve influencer after fixed interaction

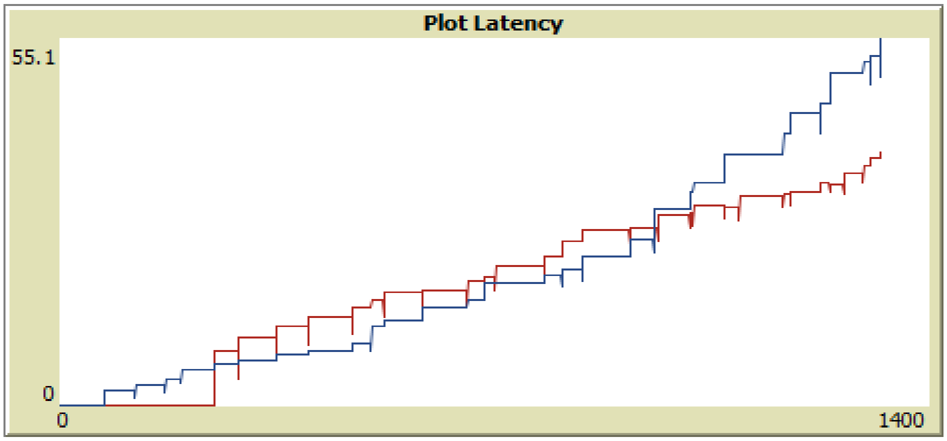


Figure 6.8 Effect on latency of original chain and forked fixed –ve influencer's chain.

By keeping –ve influencer very high we can see from Figure 6.9 that now forked chain's latency is less due to more number of nodes and original chain latency increases due to less number of nodes in the network. As a result, people who have rejected the proposal are on more numbers and forked away.

6.3 Conclusion

This chapter concludes by answering RQ4: when to fork? And the effects on the original Blockchain's latency when users holding negative opinion fork away? After simulation, we have analyzed the following factors that affect the latency i.e.$number\ of\ nodes,\ transaction\ rate \propto latency$, $percolation \propto connectivity$, $\%validator \propto 1/latency$, and finally at the time of decision for proposed change nodes with negative opinion fork away $nodes\ with\ negtive\ opinion \propto latency\ of\ orignal\ chain$ such that, fewer negative influences less effect on original Blockchain latency, more negative influencers more change on original Blockchains latency. Hence, users who don’t agree with the proposal will fork away affecting the latency of the original Blockchain greatly because of two reasons firstly a decrease in several nodes. Secondly, transactions don’t get processed on time.

Chapter 7

7 CONCLUSION AND FUTURE WORK

In this chapter, the thesis concludes with a summary of all the work done in this research. It provides the conclusion of the proposed model. Also, list the possible future work to explore.

7.1 Conclusion

As Blockchain is a relatively new domain, it is quite unstable, and all sorts of work has been going on it. As more researchers dive in to this domain, new possibilities and unexplored horizons start opening which need to be explored, pondered upon and found. For this research, the main challenge we faced was to make decision-making more structured and well regulated. Blockchain governance is a decentralized process of proposing, accepting, and rejecting change. Being decentralized in nature has its consequences such as decision-making and achieving mutual consensus is costly and challenging. Existing governance models are not so effective due to a lack of decision-making in the Blockchain environment such that users cannot connect and communicate to make instant decisions for the proposed change. Each proposal takes a long time to get accepted or rejected as a result, the community ends up initiating a fork. To solve these issues, this thesis proposes a unique governance model by the implementation of the game theory approach for instant decision-making. It uses agent-based simulation to implement ON-chain governance in conjunction with percolation theory and voting mechanism to solve connectivity and communication issues. The uniqueness of our framework is shaping opinion by modeling three types of user interaction and their influence on each other after interaction i.e. Confident, unsure and fixed users. Once a majority of the community shares the same opinion at this point with the help of game theory each user will decide to fork or not to fork such that it maximizes their utility and also produce a less resulting impact on the latency of the original Blockchain. The results from the Blockchain governance simulation confirm the validity of our framework and its effect on the latency of the Blockchain network. The research concludes that factors that affect the latency of blockchain are $number\ of\ nodes,\ transaction\, rate\ \propto latency$, $percolation\ \propto connectivity$, $\%validator\ \propto 1/latency$, and finally at the time of decision for proposed change nodes with negative opinion fork away $nodes\ with\ negtive\ opinion\ \propto latency\ of\ orignal\ chain$ such that, fewer negative influences less effect on original Blockchain latency, more negative influencers more change on original Blockchains latency. Hence, when a decision is made with a minority of network agreeing with the proposal and the majority rejecting it, the resultant effect on latency will be less as compared to when a majority of network agrees and forks apart leaving the original Blockchain with higher latency and lower transaction rate.

Some challenges which have restricted us in this thesis need to be catered in the future work.

7.2 Future Work

As discussed earlier, Blockchain governance remains to be discovered as new and innovative technology and there is a lot of need for research in the area of Blockchain governance. There are many consensus algorithms such as proof of stake POS, proof of work POW, byzantine fault tolerance BFT, etc implemented in different Blockchains. Our framework lacks the implementation of consensus mechanisms as our primary focus is Blockchain governance such that how opinions are built and communicated throughout the network. Whereas consensus mechanism is a secondary factor. So, this aspect is yet to be explored in future research. Similarly, the incentive layer needs to be analyzed for reward of influencers and how it is distributed among users. Another dimension to be explored is the implementation of the proposed model for Blockchain platforms such as private, public, and hybrid infrastructure to check the effectiveness of the proposed Blockchain governance. Also, one can work to make Blockchain voting more secure by eliminating malicious nodes from the network. Hence, this research has paved way for hundreds of possibilities to explore and work on in the future.

REFERENCES

1. B.Kris and C.Decker, Certified Blockchain Business Foundations (CBBF) Official Exam Study Guide v1.1. Blockchain Training Alliance, Inc., 2019.
2. S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", 2008. Bitcoincash.org.
3. *Governance*. Business Dictionary. (n.d.). Retrieved July 13, 2022, from https://businessdictionary.info/definition/governance/
4. Naif, A., & Nirupama, B. (2018). "Towards True Decentralization: A Blockchain Consensus Protocol Based on Game Theory and Randomness."
5. F. Ehrsam, "Blockchain Governance: Programming Our Future," Medium, 29-Nov-2017. [Online]. Available: https://medium.com/@FEhrsam/Blockchain-governance-programming-our-future-c3bfe30f2d74.
6. Governance in the Blockchain Economy: A Framework and Research Agenda Roman Beck IT University of Copenhagen 2300 Copenhagen, Denmark romb@itu.dk
7. "A Survey on Applications of Game Theory in Blockchain" Ziyao Liu, Nguyen Cong Luong, Wenbo Wang, Dusit Niyato, Ping Wang, Ying-Chang Liang, Dong In Kim https://arxiv.org/abs/1902.10865
8. V. Buterin, (2014). "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform." https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_Whitepaper_-_Buterin_2014.pdf
9. J. Pereira, M. Tavalaei and H. Ozalp, "Blockchain-based platforms: Decentralized infrastructures and its boundary conditions", Technological Forecasting and Social Change, vol. 146, pp. 94-102, 2019. Available: 10.1016/j.techfore.2019.04.030.
10. D. Yafimava, "Blockchain In the Supply Chain: 10 Real-Life Use Cases and Examples | OpenLedger Insights", Openledger.info, 2019. [Online]. Available: https://openledger.info/insights/Blockchain-in-the-supply-chain-use-cases-examples/.
11. "Identity Management on the Blockchain - ShoCard | Identity for a Mobile World", ShoCard | Identity for a Mobile World, 2019. [Online]. Available: https://shocard.com/cpt_news/identity-management-on-the-Blockchain/.
12. Vlad Zamfir "Blockchain Governance 101," Medium, Sep 30, 2018n [Online]. Available: https://blog.goodaudience.com/Blockchain-governance-101-eea5201d7992
13. Shermin, V. (2020, December 10). What is dao - decentralized autonomous organizations. BlockchainHub. https://Blockchainhub.net/dao-decentralized-autonomous-organization/
14. Wessel Reijers, Iris Wuisman, Morshed Mannan, Primavera de Filippi, Christopher Wray, et al.. Now the Code runs itself : ON-chain and OFF-chain governance of Blockchain technology. Topoi, Springer Verlag, 2018, 37, ff10.1007/s11245-018-9626-5ff. ffhal-02046663f
15. V. Buterin, "Notes on Blockchain Governance," Vitalik Buterin's website, 17-Dec-2017. [Online]. Available: https://vitalik.ca/general/2017/12/17/voting.html.
16. "LinuxFoundationX: LFS170x Blockchain: Understanding Its Uses and Implications," Course. [Online]. Available: https://courses.edx.org/courses/course-v1.

17. "What Are Forks and How Do They Impact the Price of Cryptocurrency?," Commodity.com, 30-Nov-2018. [Online]. Available: https://commodity.com/cryptocurrency/what-are-forks/
18. L.M. Goodman. (2014, September 2). Tezos — a self-amending crypto-ledger white paper. Retrieved July 12, 2022, from https://tezos.com/whitepaper.pdf
19. Dominic, W. (2016, October). DFINITY Foundation: Internet computer. DFINITY Foundation | Internet Computer.from https://dfinity.org/
20. "What is Blockchain Governance? Complete Beginner's Guide," Blockonomi, 21-Sep-2018. [Online]. Available: https://blockonomi.com/Blockchain-governance/.
21. Sean, B., Tamas, B., & Mic, B. et al. (2018, August). An introduction to Hyperledger. White paper introduction to Hyperledger. from https://www.hyperledger.org/wpcontent/uploads/2018/08/HL_Whitepaper_IntroductiontoHyperledger.pdf
22. Richard Gendal, B. (2018, May). The Corda Platform: An introduction. corda white paper . Retrieved July 12, 2022, from https://www.corda.net/content/corda-platform-whitepaper.pdf
23. Jake Yocom-, P. (2016, February). Introduction to decred governance. Overview - Decred Documentation. from https://docs.decred.org/governance/overview/
24. Acheson N (2018) Hard Fork vs. Soft Fork. CoinDesk, 16 March 2018. https://www.coindesk.com/information/hard-fork-vs-soft-fork/. Cited on 30 July 2018
25. Robles G, González-Barahona J (2012) A Comprehensive Study of Software Forks: Dates, Reasons and Outcomes. In: Hammouda I., Lundell B., Mikkonen T., Scacchi W. (eds) Open Source Systems: Long-Term Sustainability. OSS 2012. IFIP Advances in Information and Communication Technology, vol 378. Springer, Berlin, Heidelberg
26. John, L. (2019, March 11). The differences between a hard fork, a soft fork, and a chain split, and what they mean for the... Medium. From https://medium.com/@lightcoin/the-differences-between-a-hard-fork-a-soft-fork-and-a-chain-split-and-what-they-mean-for-the-769273f358c9
27. Nitta, K. (2014, January 24). decision-making. Encyclopedia Britannica. https://www.britannica.com/topic/decision-making
28. Ba, S., Stallaert, J., & Whinston, A. B. (2001). Research Commentary: Introducing a Third Dimension in Information Systems Design – the Case for Incentive Alignment. Information Systems Research, 12(3), 225-239
29. How many Bitcoin Forks are there? forkdrop.io. Retrieved July 13, 2022, from https://forkdrop.io/how-many-bitcoin-forks-are-there
30. Peters, K. (2021, July 13). A history of Bitcoin hard forks. Investopedia. from https://www.investopedia.com/tech/history-bitcoin-hard-forks /
31. Santos F (2018) The DAO: A Million Dollar Lesson in Blockchain Governance. MA Dissertation, Talinn University of Technology, Talinn
32. "Decred update proposals" https://proposals.decred.org/proposals/c96290a
33. "Tezos update proposals" https://tzkt.io/proposals
34. Alharby M and van Moorsel A (2020) "BlockSim: An Extensible Simulation Tool for Blockchain Systems." Front. Blockchain 3:28.doi: 10.3389/fbloc.2020.00028

35. Luis R, Segismundo S, William H. “Agent-Based Evolutionary Game Dynamics” Copyright Year: 2019, Publisher: University of Wisconsin-Madison
36. Alharby, M., & van Moorsel, A. (2020, April 28). BlockSim: An Extensible Simulation Tool for Blockchain systems. Frontiers. Retrieved July 13, 2022, from https://www.frontiersin.org/articles/10.3389/fbloc.2020.00028/full
37. Alharby, M., & van Moorsel, A. (2020, April 28). BlockSim: An Extensible Simulation Tool for Blockchain systems. Frontiers. Retrieved July 13, 2022, from https://www.frontiersin.org/articles/10.3389/fbloc.2020.00028/full
38. Catt, M. (2019, March 31). Blockchain Fundamentals: Latency & capacity - featuring the ark ecosystem. Medium. Retrieved July 13, 2022, from https://medium.com/ku-Blockchain-institute/Blockchain-fundamentals-featuring-the-ark-ecosystem-part-1-af1f9052e579
39. Mingxiao, D., Xiaofeng, M., Zhe, Z., Xiangwei, W., & Qijun, C. (2017, October 1). A review on consensus algorithm of Blockchain: Semantic scholar. . Retrieved July 13, 2022, from https://www.semanticscholar.org/paper/A-review-on-consensus-algorithm-of-Blockchain-Mingxiao-Xiaofeng/0bf2cb4ae68275f4fd71a30f191dc95793a0d49e
40. D Volovik and S Redner “ Dynamics of confident voting”, Center for Polymer Studies and Department of Physics, Boston University, Boston, MA 02215, USA, Published 2 April 2012, Online at stacks.iop.org/JSTAT/2012/P04003 doi:10.1088/1742-5468/2012/04/P04003
41. Percolation theory. Wikipedia. from https://en.wikipedia.org/wiki/Percolation_theory
42. Broadbent, Simon; *Hammersley, John* (1957). "Percolation processes I. Crystals andmazes". Mathematical Proceedings of the Cambridge Philosophical Society. **53** (3):629641
43. F. Harary, in Studies in Social Power (University of Michigan Press, Ann Arbor, 1959).
44. N. E. Friedkin and E. C. Johnson, Journal of Mathematical Sociology 15, 193 (1990).
45. Clifford, Peter; Aidan W Sudbury (1973). "A Model for Spatial Conflict". Biometrika. 60 (3): 581–588.
46. “Role of feedback and broadcasting in the nameing game” A. Baronchelli, Phys. Rev. E 83, 046103 (2011).
47. Castellano C, Mu˜noz M A and Pastor-Satorras R, ‘Nonlinear q-voter model , 2009 Phys. Rev. E 80 041129
48. Castello X, Baronchelli A and Loreto V, Consensus and ordering in language dynamics, 2009 Eur. Phys. J.B 71 557
49. Dall’Asta L and Castellano C, Effective surface-tension in the noise-reduced voter model , 2007 Europhys.Lett. 77 60005
50. Social consensus through the influence of committed minorities J. Xie,1 S. Sreenivasan !,1, 2 G. Korniss,2 W. Zhang,3 C. Lim,3 and B. K. Szymanski1
51. BlockSim: An Extensible Simulation Tool for Blockchain Systems Maher Alharby 1,2* and Aad van Moorsel
52. The role of inflexible minorities in the breaking of democratic opinion dynamics Serge Galam and Frans Jacobs