

A crypto-economy based distributed & asynchronous Hashgraphy algorithm for a Tracking system

By:

SUSHMITHA KOMMUSHETTY

Supervisor:

MR. BAHMAN SASSANI (SARRAFPOUR)

**A thesis submitted in partial fulfilment of the requirements for the
degree of Master of Computing**

Unitec Institute of Technology, 2018

ABSTRACT

Cryptography and economy concepts have solidified in a particularly remarkable and convoluted approach to make crypto-economy. The advancement that it has experienced over the span of the last couple of years is tremendous and it is significantly improving and being used more comprehensively.

The improvement of crypto-economy has been dependent clearly on the development of Distributed Ledger Technology (DLT), including the first and related programming and computational progressions, starting from public key cryptography. Distributed Ledger (DL) received enormous consideration in most recent years & is steadily moving on with a very promising future ahead.

This thesis gives an orderly review of significant DL algorithms - Blockchain, Tangle, and Hash graph applicable to comprehend and structure the Distributed Ledger Technology (DLT) field. Furthermore, in view of this it portrays Hashgraph ideas and explains the Hashgraph DLT functioning in detail. The primary motive in picking Hashgraph is the freshness of technology and it's amazing attributes- more faster, more fair, more efficient secure. An implementation of Hashgraph has been done by developing a prototype and compared against the popular Blockchain technology based prototype to analyse which is the more faster and more secure technology of the two. Hashgraph professes to be exponentially quick with a speed of >250,000 transactions per second (tps), more secure and more proficient contrasted with other DLT calculations Blockchain (3-4 tps) and Tangle (500-800 tps). A real-time tracking system has been developed and hashgraph concept has been applied to it to improve the its speed and efficiency and also making it all the more secure.

The outcome would then be able to be utilized for future research and also prospective successful collaborations of these algorithms in the area of DL.

Keyword/Key points:

Crypto-economy, Cryptography, Distributed Ledger Technology (DLT), Blockchain, Tangle, Hashgraph

ACKNOWLEDGEMENTS

Firstly I would like to acknowledge my principal supervisor, Mr Bahman Sassani (Sarrafpour), and my associate supervisor, Dr Iman Ardekani, for giving me the opportunity to carry out this research project. Thank you for giving me your valuable time, encouraging and guiding me in every step from the beginning throughout to the completion of this research. I would also like to acknowledge my postgraduate program director, Dr Hamid Sharifzadeh. This thesis would not have been successfully completed without their guidance. I would also express my deepest gratitude and acknowledge Mr. Leemon Baird's invention of the Swirld's Hashgraph without which this research wouldn't be possible.

Secondly, I would like to give special thanks to my previous lecturers in Department of Computing at Unitec who have provided me with valuable knowledge and skills during the course. Their knowledge and experience have empowered me to reach the level required to complete this study.

Lastly, I would like to express my appreciation and thanks to my family members who have facilitated and provided me with this opportunity and supported me during this thesis as well as the other requirements to achieve my master's degree.

TABLE OF CONTENTS

Contents

ABSTRACT	i
ACKNOWLEDGEMENTS	ii
TABLE OF CONTENTS	iii
List of Tables	viii
List of Figures	ix
LIST OF ABBREVIATIONS	xii
CHAPTER 1	1
Introduction	1
1.1 Cryptography	1
1.2 Principles of Cryptography	2
1.2.1 Encode	2
1.2.2 Validation	2
1.2.3 Virtue	2
1.2.4 Non cancellation	2
1.3 Different forms of Cryptography	2
1.3.1 Secret/undisclosed Key Cryptography	2
1.3.2 Public Key Cryptography	3
1.3.3 Hash Functions	3
1.4 Types and Techniques of Cryptography	3
1.4.1 Cryptology	4
1.4.2 Crypto-analysis	5
1.4.3 Tools for Cryptanalysis	6
1.4.4 Prerequisites and duties regarding Cryptanalysts	7
1.5 Motivation	7
1.6 Research Contribution	8
1.7 Structure of the Thesis	8

We have nine chapters which are covered briefly in this thesis.	8
1.8 Chapter Summary	9
CHAPTER 2	10
CRYPTO-ECONOMY	10
2.1 Definition of Crypto-economy	10
2.2 Structure of Crypto-economy	10
2.2.1 Consensus protocol	10
2.2.3 State Channels	11
2.3 Crypto-economy Benefits	12
2.4 Crypto-economy Risks	12
2.5 Crypto-economy based Distributed Ledger Technologies (DLTs)	12
2.6 Overview of the different types of attacks on Crypto-economy	13
2.6.1 Crypto-hacking	13
2.6.2 51 % attack	13
2.6.3 Identity theft	13
2.6.4 Sybil Attack	13
2.6.5 DDoS Attack	14
2.7 Existing prevention and defence mechanisms	14
2.8 Chapter Summary	15
CHAPTER 3	16
Distributed Ledger Technology	16
3.1 Definition of a Distributed Ledger Technology	16
3.2 SWOT Analysis	17
3.3 Important aspects of DLT	17
3.3.1 Nature of the Distributed Ledger	17
3.3.2 Consensus Mechanism	18
3.3.3 Cryptographic systems	19
3.4 Key Advantages of DLT	19
3.5 Categories of DLT	21
3.5.1 Open or Permission-less DLT	22

3.5.2 Permissioned DLT	22
3.6 Distributed Ledger Technologies (DLT) Types & Methodology of each Algorithm in detail.	23
Blockchain	23
Tangle	29
Hashgraph	31
3.7 Complete outline of the potential security dangers alongside their effects on different elements in a Distributed Ledger Technology (DLT)	38
3.8 Chapter Summary	39
CHAPTER 4	50
METHODOLOGIES	50
4.1 Research Hypothesis	50
4.2 Method Used for Study	50
4.2.1 Quantitative Research	50
4.2.2 Agile Methodology	51
4.3 Data Collection Process	54
4.3.1 Literature Review Process	54
4.3.2 Data Gathering Process for Implementation of Design and Real-time tracking application	55
4.3.3 Process of Generating Legitimate Traffic	55
4.3.4 Process of Generating Attack Traffic	56
4.3.5 Process of Evaluating Defences	56
4.4 Chapter Summary	56
CHAPTER 5	57
Implementation of Design	57
5.1 Implementation Set-up of Blockchain.	57
5.1.1 Tools & Software used	57
5.1.2 Core concepts of this proof-of-stake blockchain	58
5.1.3 Architecture	58
5.1.4 Process of creating prototype	58
5.2 Implementation Set-up of Hashgraphy.	64
5.2.1 Tools & Software used	64
5.2.2 Process of creating prototype	64
5.3 Derivation from implementation	79

5.4 Chapter Summary	79
CHAPTER 6	80
Security of Hashgraph	80
6.1 Security of Hashgraph	80
6.2 Byzantine Fault Tolerance Theorem	80
6.3 Asynchronous Byzantine Fault Tolerance v/s Partially Asynchronous Byzantine Fault Tolerance	81
6.4 Mathematical Proof of Hashgraph being fully Asynchronous Byzantine Fault Tolerant and resilient to DDoS and Sybil attacks	81
6.4.1 Hashgraph is Byzantine	81
6.4.2 Blockchain is non- Byzantine	82
6.4.3 Hashgraph is resilient to DoS/DDoS Attack	82
6.4.4 Hashgraph is resilient to Sybil Attack	83
6.5 Chapter Summary	84
CHAPTER 7	85
Evaluation of DDoS, Ping of Death & Sybil Attacks on Hashgraph	85
7.1 Overview on DDoS Attack	85
7.2 Hardware and Software Specification	86
7.2.1 Hardware	86
7.2.2 Software	87
7.3 Defence Mechanisms for DDoS Attacks	87
7.4 Implementation of Ping of Death Attack on Hashgraphy system	88
7.5 Implementation of DDoS Attack on Hashgraphy System	92
7.6 Theoretical Evaluation of Hashgraph resilient to Sybil Attack	93
7.6 Chapter Summary	93
CHAPTER 8	94
Application of Hashgraph to Real time tracking application	94
8.1 Flow-chart and architecture of the tracking application	94
8.2 Progress Reports during tracking application build	95
a. The Cumulative Flow Diagram	95
b. Epic Report	95

c. Sprint Report	96
8.3 Backend Database structure of the tracking application	97
8.4 Working of Tracking Application	98
8.5 Security of the application	103
8.6 RESEARCH QUESTIONS ANSWERED	103
8.7 EXPERIMENTAL METRICS USED	103
8.8 Chapter Summary	104
Chapter 9	105
SUMMARY, CONCLUSIONS, AND FUTURE WORKS	105
9.1 Summary & Conclusions	105
9.2 Future works	107
9.3 Future user cases for Consensus Mechanism	108
9.4 Work with Industry	108
APPENDIX	109
Appendix A: Software Specifications	109
REFERENCES	110

List of Tables

Table 3.1: Types of Blockchain networks, consortium type, public type.....	26
Table 3.2: Correlation amongst 3 types of blockchain.....	27
Table 3.3: Typical consensus algorithm comparison.....	28
Table 3.4: Comparison of mainstream DLTs blockchain, tangle, hashgraph.....	40
Table 3.5: Potential security threats on mainstream DLTs and counter measures.....	42
Table 4.1:CredibleResources.....	47
Table 4.2: Hashgraphy evaluation metrics and tools used for collecting data.....	48
Table 7.1:HardwareSpecifications.....	79
Table 7.2:SoftwareSpecifications.....	80
Table 7.3: Types of attack with mitigation techniques.....	80

List of Figures

Figure 1.1 Secret key cryptography functions.....	13
Figure 1.2 Public key cryptography	13
Figure1.3 Hash functions.....	13
Figure 2.1 Popular Distributed Ledger Technology (DLTs) of Crypto-economy.....	24
Figure 2.2 Statics depicting increase in Crypto-economy based Blockchain users from 2015-2018.....	24
Figure 3.1 Strength Weakness Opportunity Threat (SWOT) investigation on DLT	28
Figure 3.2 Structure of Blockchain, Hashgraph, Tangle.....	34
Figure 3.3 A block structure and formation of a chain of blocks	35
Figure 3.4 Blockchain transaction.....	36
Figure 3.5 Block structure	36
Figure 3.6 Gradual increase of Banks interest in Blockchain innovation from 2014 to 2017.....	38
Figure 3.7 A scenario of blockchain branches.....	39
Figure 3.8 Structure of Tangle.....	41
Figure 3.9 Rise in number of Internet of Things users from 2013 to 2019.....	42
Figure 3.10 Event specifications of Hashgraph.....	42
Figure 3.11 Structure of Hashgraph.....	42
Figure 3.12 Step 1 of working of Hashgraph.....	43
Figure 3.13 Step 2 of working of Hashgraph.....	43
Figure 3.14 Step 3 of working of Hashgraph.....	44
Figure 3.15 Step 4 of working of Hashgraph.....	44
Figure 3.16 Step 5 of working of Hashgraph.....	45
Figure 3.17 Step 6 of working of Hashgraph.....	45
Figure 3.18 Step 7 of working of Hashgraph.....	46
Figure 3.19 Step 8 of working of Hashgraph.....	46
Figure 3.20 Step 9 of working of Hashgraph.....	46
Figure 3.21 The Swirlds hashgraph accord calculation.....	47
Figure 3.22 The divide Rounds methodology.....	48
Figure 3.23 To choose Fame system.	48
Figure 3.24 The find Order system.....	49
Figure 3.25 Impressive estimated results for a Hashgraph business application.....	49
Figure 4.1 The Disciplined Agile Delivery (DAD) agile life cycle.....	51
Figure 4.2 Agile Testing -flow chart.....	52
Figure 4.3 Taking a "test first" approach to construction	53
Figure 4.4 Statistics of percentage of companies experiencing DDoS attack.....	56
Figure 5.1 Design of Proof-of-Stake of blockchain.....	58

Figure 5.2 Step 1 Setup & Import.....	59
Figure 5.3 Step 2 Global variables are declared.....	60
Figure 5.4 Different blocks.....	61
Figure 5.5 Step 3 Basic Blockchain functions are coded.....	61
Figure 5.6 Performing hash & prevhash check.....	61
Figure 5.7 Describing step 4 Validator.....	62
Figure 5.8 How the winner is picked.....	63
Figure 5.9 Efficiency of Blockchain.....	63
Figure 5.10 Step 6 of proposing a polluted block.....	64
Figure 5.11 Members of hashgraph.....	65
Figure 5.12 Participants initiate an occasion.....	65
Figure 5.13 Event description.....	65
Figure 5.14 Members Gossip randomly.....	65
Figure 5.15 Live Demonstration of Members gossiping.....	66
Figure 5.16 Hashgraph Formation	66
Figure 5.17 Hashgraph Formation proceeds everlastingly.....	66
Figure 5.18 Live Demonstration of hashgraph formation.....	67
Figure 5.19 Code snippet for statistics and hashgraph formation.....	67
Figure 5.20 Code snippet for x & y coordinates of the Hashgraph created.....	67
Figure 5.21 Creation of rounds logic.....	68
Figure 5.22 Live Demonstration of Creation of rounds.....	68
Figure 5.23 Probability- one Theorem.....	68
Figure 5.24 Live Demonstration describing colour for an event in the consensus algorithm.....	69
Figure 5.25 Working Code Snippet that shows details of Hashgraph	70
Figure 5.26 Consensus information on every members profile in the Hashgraphy simulation application....	70
Figure 5.27 Aek consensus information.....	70
Figure 5.28 Ben's Consensus information.....	71
Figure 5.29 Cate's Consensus information	71
Figure 5.30 Dev's Consensus Information.....	71
Figure 5.31 Voting method.....	72
Figure 5.32 Aek trans/sec maximum.....	72
Figure 5.33 Aek trans/sec minimum.....	73
Figure 5.34 Ben trans/sec maximum.....	73
Figure 5.35 Ben trans/sec minimum.....	74
Figure 5.36 Cate trans/sec maximum.....	74
Figure 5.37 Cate trans/sec minimum.....	75
Figure 5.38 Dev trans/sec maximum.....	75
Figure 5.39 Dev trans/sec minimum.....	76

Figure 5.40 Working code snippet of the graphic context	76
Figure 5.41 Working code snippet of retrieving graphic context for each member.....	76
Figure 5.42 Working code snippet of getStats functionality.....	77
Figure 5.43 Factors of three popular DLT.....	77
Figure 6.1 Structure of swirld identifier.....	79
Figure 6.2 Sybil attack on hashgraph.....	83
Figure 7.1 Size of Largest reported DDoS attack in Gbps 2002 to 2012.....	85
Figure 7.2. DoS/DDoS attack setup.....	86
Figure 7.3 Wi-Fi Properties screen.....	89
Figure 7.4 IPv4 Properties screen.....	89
Figure 7.5 Advanced TCP/IP settings box.....	90
Figure 7.6 Ping of death result window.....	91
Figure 7.7 System Task management.....	91
Figure 7.7 DDoS attack demonstration.....	92
Figure 7.8 DDoS attack effect on Aek's performance.....	93
Figure 8.1 Flowchart of tracking application.....	94
Figure 8.2 Cumulative flow diagram.....	95
Figure 8.3 Epic report.....	96
Figure 8.4 Sprint report.....	96
Figure 8.5 Entity Relationship Diagram.....	97
Figure 8.6 List of all vehicles.....	98
Figure 8.7 Actions List	98
Figure 8.8 Map View.....	99
Figure 8.9 Leader Board.....	99
Figure 8.10 GPS Location Radius.....	100
Figure 8.11 Real-time chatting Feature.....	101
Figure 8.12 Trips information.....	101
Figure 8.13 Event Log.....	102
Figure 8.14 Vehicle Profile.....	102
Figure 8.15 Driver profile.....	103

LIST OF ABBREVIATIONS

BFT	Byzantine Fault Tolerance
BPS	Bytes per Second
BPT	Bytes per Transaction
CPU	Central Processing Unit
CSS	Cascading Style Sheet
DAD	Disciplined Agile Delivery
DAG	Directed Acyclic Graph
DDoS	Distributed Denial of Service
DLA	Distributed Ledger Algorithms
DLT	Distributed Ledger Technology
DNS	Domain Name System
DoS	Denial of Service
ECDL	Elliptic Curve Discrete Logarithm
EPS	Events per Second
HTML	Hyper-text Markup Language
IP	Internet Protocol
IPS	Intrusion Prevention System
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
ISP	Internet Service Provider
JDK	Java Development Kit
KSI	Keyless Signature Infrastructure
LAN	Local Area Network
Mbps	Megabits per Second
MS	Millisecond
NFS	Network File System
NIC	Network Interface Card
NTP	Network Time Protocol
OS	Operating System
PC	Personal Computer
POD	Ping of Death
PoS	Proof-of-Stake
PoW	Proof-of-Work
PPS	Packets per Second
QoS	Quality of Service
RAM	Random Access Memory
SE/EE	Software Edition/Enterprise Edition
TCP	Transmission Control Protocol
TCP/IP	Transport Communication Protocol and Internet Protocol
TDD	Test Driven Development
TPS	Transactions per Second
UDP	User Datagram Protocol

CHAPTER 1

Introduction

Crypto-economics is formed from two words: Cryptography and Economics.

Verification and Validation , Confidentiality & Integrity are main characteristics that reflects in Cryptography. Digital signatures and hashing are mainly the two ideas that are used for cryptography [2].

In simple terms, hashing infers of taking data upto any length & give away output for a fixed length. Private Key being confidential key to client is required for recognizing the data transactions, additionally public key for verification and the link which is public is utilised by the users. Confidentiality can be maintained in public key which itself is a type of a hash & performs the similar functions.

The thought behind digital signature is to achieve cryptographical set-up in the most unique way. Digital signatures seems to be non-forgable, non-revokable, alongside regular validations happening every time, but in real it is not the case. Regardless of how much complex the digital signatures are, there is high odd chances of it being hacked. However these answers can be obtained by cryptography which utilises the concept of both the keys. Here there is one important thing to note : It's inadmissible to choose a public key from another person's private key.

With the impact of hashing techniques on data, verification and digital signatures contrives that rely upon keys stand up to the issues of loss of key or key's denial of access situations. The ascent of quantum PCs has constructed the shot of breaking these regular encryptions. To mark these security issues, Keyless Signature Infrastructure (KSI) has been designed by expert analysts [3-6]. KSI spares the present condition of information loss , structure or system and moreover the hashing techniques. KSI will at that point lead a steady eye, watching out for these hash functions with timestamp, which in a way differentiates whether a catalogue, working system or application is encountering an unapproved access. For censorious information foundation for many important resources & structures, a safety protection organization has been build. By the ahead of time, systems trustworthiness and security of information will be kept up.

Along these lines, Cryptography and money related issues have solidified in an astoundingly flawless and muddled approach to make crypto-economy. The advancement that it has experienced over the span of the latest couple of years is incrementing and it is simply going to hint at enhancement and wider usage.

1.1 Cryptography

Cryptography is a procedure for guaranteeing information and trades utilizing codes with the objective that those for whom the information is normal can examine and process it. Word "Crypt" signifies "covered up" or "concealed" & word "graphy" implies "forming".

In software engineering, cryptography gives us data that we can securely change into other data which cannot be unwinded. These deterministic estimations are used for cryptographic key age and digital verification and validation to anchor data insurance, digital scrutinizing on the web and mystery correspondences, for instance, MasterCard trades and mail.

In recent time, for many researchers & some great mathematicians cryptography has become an important aspect for them to achieve milestones. The ability to securely store and trade unstable information has shown progressive achievement in war and business. [1]

In many nations cryptography is only confined to take away the limitations of its usage. Most likely, the technique is used by scientific population for ideas of developing cryptosystems. Nonetheless, web permits distribution of such innovative projects & strategies related to cryptography, with the goal of being an impressive segment for progressive cryptosystems in general society space.

1.2 Principles of Cryptography

1.2.1 Encode

In the most straightforward approach, encoding refers to remodelling data in structural form. This approach helps to secure and protect when transmitting data to the recipient. Also for the support of experts, information is decoded & then reclaimed back in a unique form. The turnaround in encryption is known as unravelling or decoding. Encoding & decoding needs extra information to encipher & decipher information. Such extra information can be called key. In some instances to encode and decode, we can use the same key, whereas in some special cases different key may be required for encoding & decoding.

1.2.2 Validation

Another fundamental standard of cryptography is Validation. In simple words, a message can be initiated by originator stated in text, by guarantying its confirmation. Directly, one may think how to make it possible? Let's consider, Aek establishes a connection on Ben and now Ben needs confirmation that the message has been in fact sent by Aek. It's possible only when some action has been performed by Aek on the text which Ben think nobody, but only Aek can do. Considering all aspects, this shapes up the fundamental entity of Authentication.

1.2.3 Virtue

Presently, one issue with communication framework is that it confronts loss of uprightness of texts that are transmitted from transmitter to recipient. Which means that Cryptography should make sure about text that's received by recipient and doesn't differ anywhere in the communication channel.

1.2.4 Non cancellation

What will be the outcome if Aek makes an impression on Ben and doesn't accept the fact about sending the text? Situation similar to this case can happen, so cryptography should retain the originator or transmitter should act on these things. Digital signatures is one way to achieve this result.

1.3 Different forms of Cryptography

Cryptography techniques can be classified into the given types below:

- Secret/undisclosed key Cryptography
- Public/accessible key cryptography
- Hash Functions

1.3.1 Secret/undisclosed Key Cryptography

Single key is used in such kind of technique. Transmitter puts a key to encode a text whereas the receiver also puts in alike key to decode its text. This type is also known as balance encoding because only 1 key is utilised.

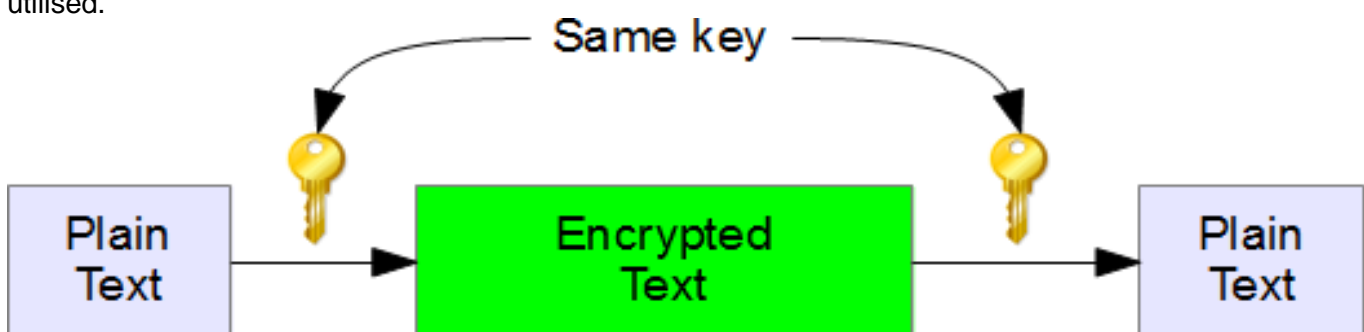


Figure 1.1 Secret key cryptography functions.

One of the biggest problem along such technique is distribution of key because such algorithm uses single key for encoding or decoding.

1.3.2 Public Key Cryptography

To have strong & assured transmission between transmitter & receiver, there are 2 crypto systems which are involved. This helps us to have secured and protected correspondence. Now transmission can take place without worrying about any interference from anyone. Such technique is also known as unsymmetrical encoding because of 2 keys utilised.

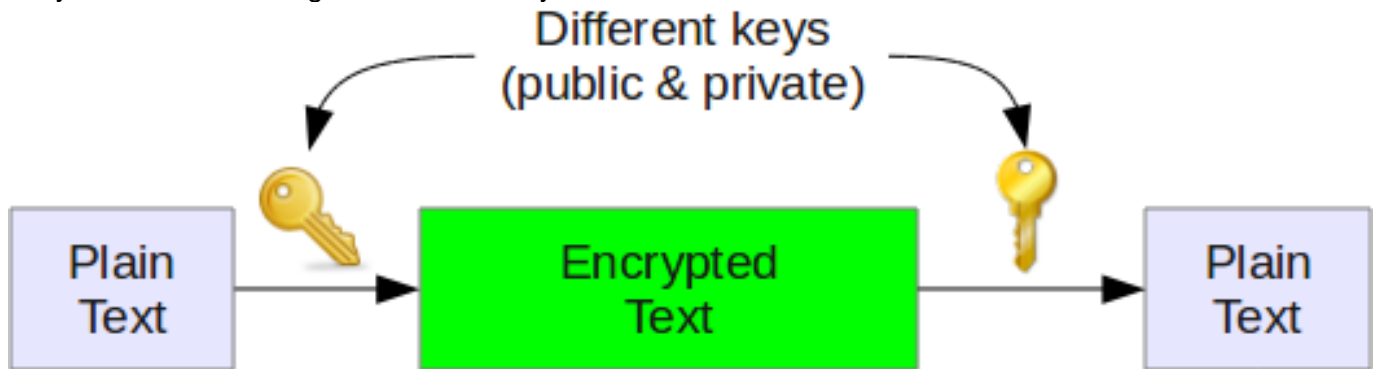


Figure 1.2 Public key cryptography

This type of technique has both types of key means open as well as restricted key. Restricted key is unpublished and confidential as it is not revealed to anyone, whereas open key is distributed amongst everyone who wants to communicate. For example Aek desires to transmit text over to Ben, then Aek should first encode the text using Ben's open key, after that Ben should change text using its own confidential or unpublished key.

So that's how we do the setup for open key validation by open ssh to sign in using 1 server and then moving to another server to avoid infiltration in confidential or unpublished key.

1.3.3 Hash Functions

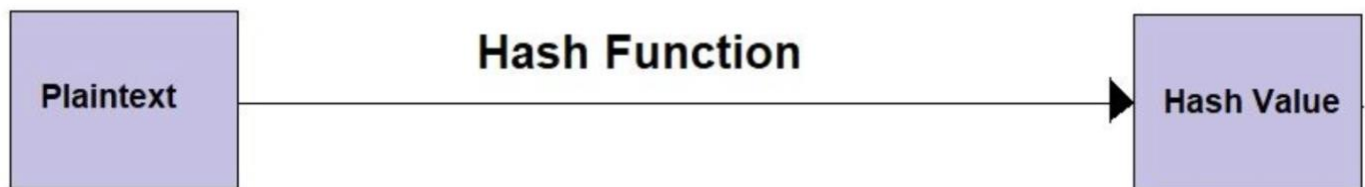


Figure 1.3 Hash functions.

No key is involved in such systems. Rather this function uses values of hash which are fixed and generated by plain text. Integrity of messages are checked using hash functions to make sure it is not altered & compromised.

1.4 Types and Techniques of Cryptography

Cryptology & Cryptanalysis are mainly two assets related to cryptography. They consolidate strategies, just like how microdots combine pictures along with words, help in dealing with information that has been covered. Regardless, in today's tech savvy world, cryptography routinely helps in combining plaintext which are encrypted into the ciphertext by using a technique known as encryption, and then again gets back to plain readable text by decrypting. The ones who get trained in this area of specialization are called cryptographers. [2]

1.4.1 Cryptology

A technique which helps in identifying numbers, conditions, estimations for cryptography is known as cryptology. As we know that cryptanalysis ideology is to be precise & a bit difficult to understand, we however mainly concentrate on the process that helps to run cryptography. With the objective for data to be moored for limit or transmission, it must be changed in such a way, to the point that it would be troublesome for an unapproved individual to have the ability to locate its genuine criticalness. To do this, particular logical conditions are used, which are incredibly difficult to unwind if some of the very stringent benchmarks are achieved. [3]

Some of the issues related to cryptology are mentioned below:

Discrete algorithm issue:

This problem can be best described by displaying the concept of its retrogressive works- declining from a better to a worse state. Suppose there is a number P (which is only divisible by 1 & number itself). Such number P is prime number in excess of 300 units. Assume having 2 integers, h & i. So we now have to calculate "N" estimation, with the objective to identify its worth as shown below.

$$N = h^i \bmod P, \text{ where } 0 \leq N \leq (P - 1)$$

Such an equation is called discrete exponentiation, which is not that difficult for understanding. In any case, the backwards is real when we change it. Now we have been provided with P, h & N, so by making the formation valid, we can find the value of I. But it gets difficult to get the value.

Such issue shapes its purpose behind different open key structure estimations. Many experiments are carried out on similar issues, so the cryptography associated with it, has faced & survived different types of attacks.

Integer Fraternization issue:

Conceptually this issue is quite easy and simple. Imagine if we have 2 large prime numbers p1, p2. To generate the outcome, N we then have to multiply these numbers. It becomes really difficult to find the real value of p1 & p2 when N has already been given. Best example which is based on such issue is "Rivest-Shamir-Adleman" open structure key encryption. So in the end the final outcome what we get is that p1 & p2 are the closed key whereas the N product is open key structure which helps in improvising at an exceptional degree.

This issue had been analysed without any doubt as far back as 20 years, and the agreement is apparently that there is some new law of number manipulations that disallows any substitute ways. Everything considered, the insignificant conviction that it is being reviewed into, drives various additional for pushing it so that going forward it can be found easily.

Elliptic Curve Discrete algorithm issue:

This is another cryptographic tradition subject to a reasonably prominent logical issue. The properties of elliptic twists have been striking for a significant long time; anyway because it came at a later stage, its approach towards cryptography has more firm and better understood.

Assuming there are vertical and horizontal lines on a very huge piece being printed. Each line addresses an entire number with the vertical lines confining x class portions and even lines forming the y class fragments. The intersection purpose of a level and vertical line gives a ton of bearings (x,y). In the exceedingly clear model underneath, we have an elliptic twist that is portrayed by the condition:

$y^2 + y = x^3 \cdot x^2$ (this is confusing for use in a straight-forward application, yet it will speak for the general idea)

As mentioned previously, given a quantifiable manager, we can choose any third point on the curve given any two distinctive main points. This definite manager outlines a "gathering" of restricted length. To incorporate to an elliptic curve, we first need to understand that any straight line that experiences this twist meets it at undoubtedly three points. Eventually, we portray two of these concentrations as states u and v : we would then have the capacity to draw a straight line through two of these concentrations to find another intersection point, at w . We would then have the capacity to draw a vertical line through w to find the last meeting point at x . Directly, we can see that $u + v = x$. This standard works, when we describe another inconsistent point i.e, the Origin, or O , which exists at (speculatively) unknown spotlights on the twist. As odd as this issue may show up, it grants for a convincing encryption structure; anyway it has its depreciators. [4]

1.4.2 Crypto-analysis

By studying Cryptanalysis we can get to know about ciphertext, its figures, also about cryptosystems. Main purpose of this study for cryptosystem is to find & upgrade the techniques to conquer or disable it & also to study its working system. Taking an e.g. ciphertexts are decrypted by cryptanalysts without learning about plaintext origin, the key required for encryption, its algorithm that has been used for encrypting it; also cryptographic algorithms are being targeted by cryptanalysts which involves hashing and usage of digital signatures.

Main aim for cryptanalysis should be finding inadequacies & results that are achieved by cryptanalysts investigation. Also it helps in improving & making the defective algorithms strong. Now looking at both, cryptography's aim revolves around making and upgrading distinct encryption figures, and cryptanalysis for deciphering encoded information.

Many different techniques of attacks can be discovered by Researchers that may help to totally destroy an algorithm which is encrypted, meaning all the ciphertexts which are encrypted along these algorithms have been modified even by not gaining access for the key that's encrypted. Quite often an algorithm's weakness & implementation are showcased by the results achieved by cryptanalytcs. This reduces keys size for achieving its ciphertext.

Considering the figure which contains approximately 128 bits of key that has been encrypted should have 2128 novel solutions; all things considered, bug control attacks along with the figure should evolve to attack less than whole of unique keys. Now in case, cipher cryptanalysis figure out's a attack which reduces the trial numbers required for 240 keys that are different, so we can get an idea that algorithm is destroyed by general sense, up to a level such that attack (Brute force) will be possible in commercially built structures.

Many different affiliates practice Cryptanalysis, that includes government interpreting different nations in ordered correspondences; associations making security things that use cryptanalysts to test their security features; and software engineers, & those individuals who lack education in cryptographic traditions and computations. It is this persistent battle between cryptographers attempting to keep information and cryptanalysts attempting to break cryptosystems that moves the entire array of cryptology data forward. [5]"

Crypto-analysis approach and violation

There are a wide scope of sorts of cryptanalysis attacks and systems, which vary dependent upon how much information the agent has about the ciphertext being analysed. Some cryptanalytic strategies include:

- In ciphertext-only attack, the information about the plaintext data is not known to the attacker. It can only access one or more message which is in encrypted form. In this the attacker also have no information about the algorithm being used for encryption or about the cryptographic key. This is the sort of test that information workplaces normally stand up to when they have received encoded exchanges from an adversary. .
- In a known plaintext attack, access has been granted to the analyst for most of plaintext; with one major duty of analyst being to identify the type of key used for the message to be encrypted or decrypted. At the point when the key is discovered, the attacker can change all messages that had

been encoded using that key. Clearly cryptanalysis is a type of known plaintext attack that uses an immediate estimation to describe how a known plaintext attacks depending upon the attackers ability to discover few or rather most of a encoded message, or even the arrangement of the first plaintext. For example, if the attacker realizes that a particular message is steered to or about a particular person, that person's name may be a sensible known plaintext.

- In a select plaintext attack, encryption algorithm is either known to the analyst or they are provided with the gadget to perform encryption. Information related to key can be derived by analyst who encrypts plaintext along the intended algorithm.
- A differential cryptanalysis attack is a type of selected plaintext attack that examines sets of plaintexts rather than single plaintexts. So when a intended algorithm comes across types of data , the analyst can figure out on how the targeted algorithm will operate.
- Basic cryptanalysis attacks are nothing but resemblance of cryptanalysis attacks that are differential, where plaintext sets are used instead of its pairs. Here, only some part of the pair is kept constant and the remaining plaintext part is revised or reformed. To get the best results of this attack , we have to perform it onto the block cipher which are related upon the substitution-change frameworks.
- A side-channel attack depends upon information accumulated from the physical structure being used to encode or translate. Productive side-channel uses data that is neither the ciphertext coming about on account of the encryption system nor the plaintext to be mixed, but rather may be related to the proportion of time it takes for a structure to respond to express request, the proportion of force eaten up by the encoding system, or electromagnetic radiation created by the scrambling structure.
- A vocabulary strike is a technique usually used against mystery state archives and attempts the human affinity to use passwords reliant on trademark words or viably proposed continuation of letters or numbers. The dictionary attack works by encoding all of the words in a word reference and short time later checking whether the resulting hash organizes a mixed mystery express secured in the SAM record structure or other mystery state report.
- Man-in-the-middle attacks happen when cryptanalysts find ways to deal with attack implants themselves, into the correspondence channel between two public events who wish to exchange their keys for secure correspondence by methods for lost or public key establishment. The attacker by then plays out a key exchange with each public event, with the primary agreement believing they are exchanging keys with each other. The two public events by then end up using keys that are known to the attacker.

Various types of cryptanalytic attacks can join methodology for convincing individuals to reveal their passwords or encryption keys, making Trojan horse programs that take mystry keys from disastrous losses' on PCs and send them back to the cryptanalyst, or deluding a harmed individual into using an incapacitated cryptosystem.

Side-channel attacks have also been known as timing or differential power analysis. These surprise attacks came to wide notice in the late 1990s when cryptographer Paul Kocher was disseminating results of his examination into timing attacks and differential power analysis attacks on Diffie-Hellman, RSA, Digital Signature Standard (DSS) and distinctive cryptosystems, especially against executions on cash cards. [6]

1.4.3 Tools for Cryptanalysis

Since cryptanalysis is essentially a numerical subject, the devices for doing cryptanalysis are described in educational research thesis. Regardless, there are various tools and distinctive resources available for those excited about getting comfortable with doing cryptanalysis. Some of them include:

- CrypTol is an open source adventure that produces e-learning programs and an electronic interface for getting some answers concerning cryptanalysis and cryptographic computations.

- CrypTol is a space unequivocal vernacular at first expected to be used by the National Security Agency deciding cryptographic figures. CrypTol is circulated under an open source permit and available for open use. CrypTol makes it useful for customers to screen how counts function in programming programs written to decide the computations or figures.
- CrypTol can be used to oversee cryptographic timetables instead of with entire cryptographic suites.
- CryptoBench is a program that can be used to do cryptanalysis of ciphertext delivered with various fundamental findings. It can disorganize or disentangle with 29 distinct symmetric encryption computations; encode, change, sign and affirm with six different open key findings; and create 14 different sorts of Cryptographic hashes similarly as two exceptional sorts of checksum.
- Ganzúa (which implies picklock or skeleton key in Spanish) is an open source cryptanalysis instrument used for set up polyalphabetic and monoalphabetic figures. Ganzúa allows customers to portray optional figure and plain letters combinations, considering the right cryptanalysis of cryptograms obtained from non-English substance. Being a Java application, Ganzúa can continue running on Windows, Mac OS X or Linux. [7]
- Cryptanalysts normally use various other data security mechanical assemblies including framework sniffers and mystery state breaking programming; anyway it isn't rare for a cryptanalytic pro to make their own special custom instruments for unequivocal endeavours and challenges.

1.4.4 Prerequisites and duties regarding Cryptanalysts

A cryptanalyst's obligations may incorporate creating calculations, figures and security frameworks to encode delicate data and information and breaking down and decoding distinctive sorts of hidden data, including disorganized information, figure writings and media communications conventions, in cryptographic security frameworks.

To ensure about their security of networks, the information which are sensitive are encrypted in one form will be sent through its own network to many agencies including those associated with government contracts with cryptanalysts.

Cryptanalysts can be in commanding of many different obligations but not limited to:

- Shielding basic data from being blocked replicated, changed or erased.
- Assessing, breaking down and focusing on shortcomings in cryptographic security frameworks and calculations.
- Planning security frameworks to counteract vulnerabilities.
- Creating numerical and factual models to dissect information and take care of security issues.
- Testing computational models for exactness and dependability.
- Exploring, testing, checking & inquiring about cryptology speculations that are new, also for its applications.
- Hunting down shortcomings in correspondence lines.
- Guaranteeing budgetary information is not in order and opens just for approved clients.
- Assuring that transmitted information messages are not hacked & balanced for any movement.
- Release secretive texts used for military coding systems, prerequisite required for law & sources affiliated to government.
- Discovering way on for the information to be encrypted similar to methodologies which are new for encoding texts to cover up information which are tricky.

1.5 Motivation

Based on literature analysis, the reason for conducting the research work on Hashgraphy technique are:

1. The freshness and crisp idea of Hashgraphy.

Hashgraphy is the latest technological advancement of DLT. As it is new very little work has been done on

it yet. By directing the theory work, the idea of Hashgraphy will be connected to monitor and analyse a real-time tracking system.

2. The outstanding test results of Hashgraphy.

Mance Harmon, Co-facilitator of Swirlds and Hedera expresses that Hashgraph can process unlimited trades each second, showed up distinctively in connection to proof of work blockchains like Bitcoin or Ethereum's blockchain that can complete 5-7 trades reliably. This relationship depended upon the Hashgraph's open source test results.

Since time is an exchange off between throughputs, inaction, quantities of PCs, and geographic course, the tests show these exchanges off. For instance, the outcomes show 30 PCs can accomplish 50k trades each second transversely in excess of 8 by and large regions in 3 seconds, or simply 1.5 seconds crosswise over more than 2k miles, or .75 seconds in a single region. [8]

The attacks could have been prevented if appropriate security systems had been in place. All of these examples demonstrate the ineffectiveness of existing security mechanisms to detect and prevent DDoS attacks, and, as a consequence, are the motivation behind this research.

1.6 Research Contribution

In this research, I am applying Hashgraphy algorithm to real time tracking system to monitor and analyze Vehicles and to track their location accurately.

The algorithm is intended to be lightning fast, secure, safe and efficient to improve city transport system of vehicles road routes, or delivery tracking etc.

Real-time data will be delivered from weather to traffic conditions and the best route will be suggested.

The algorithm is supposed to resist to DDoS, Sybil attacks. It implements Keyless Signature Infrastructure (KSI) to address the security threat of identity loss.

1.7 Structure of the Thesis

We have nine chapters which are covered briefly in this thesis.

- *First Chapter* , the current chapter, introduces & gives brief description of cryptography & mentions the history of cryptography, the motivation behind this study, followed by the contribution of this research.
- *Chapter two* provides an overall view of crypto-economy, its benefits and risks involved. It also gives a short description of crypto-economy based Distributed ledger technologies (DLTs). There is also coverage of different types of attacks on crypto-economy. Mitigation techniques are also described along with existing prevention & defence mechanism.
- *Chapter three* gives the brief description about DLTs. SWOT analysis has been done. Key features of DLT such as the nature of distributed ledger, its mechanism showing consensus & cryptographic mechanisms. Categories of DLT, different types of DLT and its methodology are explained in the chapter. Security dangers alongside effects on DLT are outlined.
- *Chapter four* covers the hypotheses and the methodology employed for this research. It also mentions that what type of methodology was chosen and why. This chapter also presents the process of collection of data & the ways in identifying the literature review process & research experiment.
- *Chapter five* includes a detailed explanation and implementation of design. Firstly it covers the implementation of blockchain, the specification of all hardware and software used in the implementation design & concept of proof of stake. The architecture design is also given to understand the process of creating prototype. Then comes implementation of hashgraph. Tools and software used for implementation and process of creating prototype.

- *Chapter six* covers the security of hashgraph. Byzantine fault tolerance theorem has been done. Mathematical proof of hashgraph sums up this chapter.
- *Chapter seven* evaluates and give overview about DDoS attack, defence mechanism for DDoS attack. Implementation of ping of death on hashgraph, then implementation of DDoS attack on hashgraph and sybil resilience of hashgraphy has been explained.
- *Chapter eight* is the application of hashgraphy to real time tracking application. It gives the architecture and flow chart, list of tables and explains the working of tracking application.
- *Chapter nine* is the final chapter, which covers the conclusion, discussion, and directions for future works.

1.8 Chapter Summary

Concept related to Cryptography has been explained and described in this chapter. It also explains the Elliptic Curve Discrete Logarithm Problem. It also specified about cryptanalysis which was the invention of cipher text- a major game changer in security. The various tools for cryptanalysis have been listed out as well. Furthermore, this chapter illustrates the Research motivation, Research contribution and Structure of Thesis.

The next chapter provides knowledge about Crypto-economy.

CHAPTER 2

CRYPTO-ECONOMY

This chapter covers the details of Crypto-economy. Section 2.1 explains the definition of Crypto-economy. Section 2.2 describes the structure of crypto-economy concept and includes consensus protocol, crypto-economic application design and state channels. Section 2.3 and 2.4 describes the benefits and risks involved in Crypto-economy. Section 2.5 covers crypto-economy based Distributed Ledger Technologies (DLTs). Section 2.6 gives the overview of the different types of attacks on Crypto-economy. Section 2.7 explains the existing prevention and defence mechanisms and propose mitigation techniques.

2.1 Definition of Crypto-economy

Crypto-economics originates from two words: Cryptography and Economics.

Ethereum engineer Vlad Zamfir states crypto-economy as “A formal discipline that studies protocols that govern the production, distribution, and consumption of goods and services in a decentralized digital economy. Cryptoeconomics is a practical science that focuses on the design and characterization of these protocols.” [9]

2.2 Structure of Crypto-economy

There are three types of structures which are being developed and referred to as “crypto-economy”.

2.2.1 Consensus protocol

Blockchains can achieve dependable agreement without depending on a central council – a result of cryptoeconomic structure. Bitcoin's answer, which we reviewed above, is designated "proof-of-work" (PoW) agreement since machines must submit work – as equipment and power – so as to take an interest in the system and get mining rewards.

Enhancing proof-of-work frameworks and structuring options in contrast to them is one dynamic region of cryptoeconomic research and plan. Ethereum's present proof-of-work protocol component incorporates numerous varieties and enhancements for the first structure, empowering quicker open events and being progressively not allowing events information to pass through to the mining centralization that can result from application-specific integrated circuits(ASICs).

Soon, Ethereum intents moving to "proof-of-stake" (PoS) agreement model known as Casper. This is a choice to PoW that does not require "mining" in the typical sense: there is no requirement for particular mining equipment or tremendous consumptions of power.

Keep in mind that the general purpose to purchase machine equipment and spend power is to force an expense on mineworkers, as a method for raising the aggregate expense of attempting a 51 percent attack adequately high that it turns out to be excessively costly. The thought behind PoS frameworks is to utilize stores of cryptographic money to make a similar disincentive, rather than misusing equipment and power.

So as to mine in a PoS framework, you should submit a specific measure of ether into a security-savvy contract. Just like in PoW, this raises the expense of a 51 percent attack – an attacker would need to submit a lot of ether to effectively attack the system, which they would then lose for eternity.

2.2.2 Crypto-economic application design

When we have tackled the essential issue of blockchain agreement, we can assemble applications that sit "to finish everything" of a blockchain like ethereum. The hidden blockchain gives us (1) a unit of significant worth that can be utilized to make motivators and punishments, and (2) a toolbox with which we can structure restrictive rationale as "savvy contract code." The applications we work with these instruments can likewise be a result of cryptoeconomic plan.

For example, a decentralized market prediction protocol like Augur requires cryptoeconomic systems so as to work. Utilizing its local token REP, Augur makes an arrangement of motivating forces that rewards clients for detailing "reality" to the application, which is then used to settle bids in the auction forecast. This is the advancement that makes a decentralized forecast showcase conceivable. Another forecast showcase, Gnosis, utilizes a comparable technique, however additionally gives clients a chance to indicate different instruments for deciding genuine results (normally called "prophets").

Cryptoeconomics is likewise connected to structure token deals or ICOs. Gnosis, for example, utilized a "Dutch closeout" as a model for its token sale, on the hypothesis this would result in an all the more reasonable assumptions (a trial that had blended outcomes). We referenced before that one zone where instrument configuration has been connected is in the plan of closing, and token deals give us another chance to apply a portion of that hypothesis.

These are an alternate sort of issue than building the hidden consensus protocols, yet they share enough likeliness that both can be well observed as cryptoeconomic. Building these applications requires a comprehension and keen-eye of how motivating teams shape clients' conduct and structure of financial systems that can dependably create a specific outcome. They additionally require a comprehension of the capacities and restrictions of the fundamental blockchain on which the application is fabricated.

Numerous blockchain applications are not results of cryptoeconomics; for example, applications like Status and Metamask – wallets or stages allow clients to collaborate with the ethereum blockchain. These don't include any extra cryptoeconomic systems past those that are as of now part of the hidden blockchain

2.2.3 State Channels

Cryptoeconomics likewise incorporates the act of planning a few arrangements of associations between people. The most eminent of these are state channels. State channels are not an application but rather an important system that can be utilized by most blockchain applications to enable progressive productivity.

An essential confinement of blockchain applications is that blockchains are costly. Sending and receiving information requires charges, and utilizing ethereum to run savvy contract code is similarly exorbitant to different sorts of calculation. The thought behind state channels is that we can make blockchains increasingly effective by moving numerous procedures off-chain, while as yet holding blockchains trademark dependability, using cryptoeconomic structure.

Envision Aek and Ben need to trade a substantial number of little instalments of digital currency. The typical route for them to do this is send exchanges to the blockchain. This is wasteful – it requires paying exchange expenses and sitting tight for the affirmation of new squares.

Rather, envision that Aek and Ben sign exchanges that could be submitted to the blockchain however are most certainly not. They pass these forward and backward between each other, as quick as they need – there are no expenses, since nothing is really hitting the blockchain yet. Each refresh "signals" the last one, refreshing the consensus between the exchanges.

Whenever Aek and Ben have completed the process of trading little instalments, they "finish off" the channel by presenting the last state (for example the latest marked exchange) to the blockchain, paying just a solitary exchange charge for a boundless number of exchanges between themselves. They can confide in this procedure on the grounds that both Aek and Ben realize that each refresh go between them could be sent to the blockchain. In the event that the channel is appropriately structured, there is no real way to

cheat – state, by attempting to present a past refresh just as it were the latest – since response to the blockchain is constantly accessible.

For illustrative purposes, you can think about this as like how we connect with other known sources, similar to a legitimate framework. At the point when two parties sign an agreement, more often than not they never need to indict that agreement and request that a judge solve and uphold it. On the off chance that the agreement is legitimately planned, the two parties basically do what they guaranteed to do, and never collaborate with the courts by any stretch of the imagination. The way that either gathering could go to the court and have the agreement authorized is sufficient to make the agreement helpful. [10]

Later on, most blockchain applications will utilize state channels in some shape. It is quite often a strict enhancement to require less on-chain activity, and numerous things done on-chain today can be moved into state channels while as yet protecting an adequately high certification to be helpful.

The portrayal above skirts numerous critical details and details of how state channels function. For an increasingly definite depiction, Ledger Labs constructed a toy execution the previous summer that shows the essential idea.

2.3 Crypto-economy Benefits

The decentralized nature of crypto-economy brings with it a plethora of benefits.

Performance: Overall performance is increased because the computational load is spread across various nodes.

Reliability: If one node goes down, performance demands on other nodes goes up and work continues.

Scalability: Can adjust number of nodes working depending on high demand / low demand. Save power and wear on the system.

2.4 Crypto-economy Risks

If $>2/3$ nodes go down then system may crash.

As it is a decentralized setup, nodes work on various tasks and may contain different data at different times, before sharing with other nodes. Failure in a node before data sharing may result in loss of data. [11]

2.5 Crypto-economy based Distributed Ledger Technologies (DLTs)

The Figure 2.1 classifies the popular DLTs as Blockchain, Hashgraphy and Tangle.

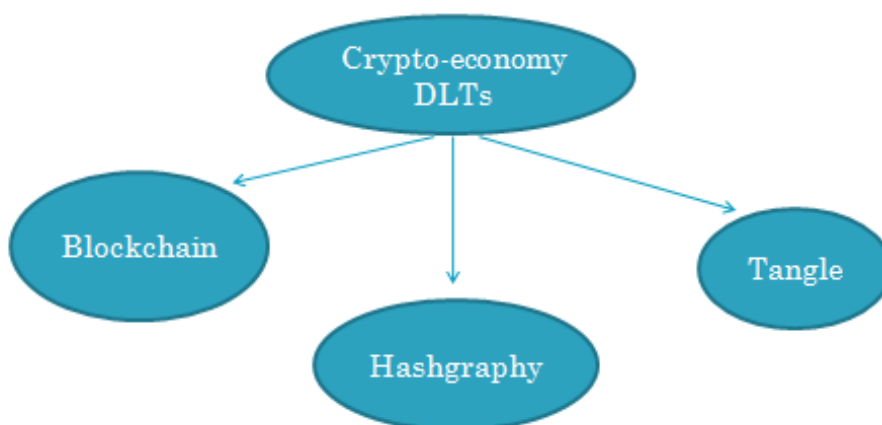


Figure 2.1 Popular Distributed Ledger Technology (DLTs) of crypto-economy

The examination underneath in Figure 2.2, demonstrates the abundance of real Blockchain wallet users for the years 2015 – 2018 in a quarterly period. This likewise outlines the measure of these budgetary segments of Blockchain.

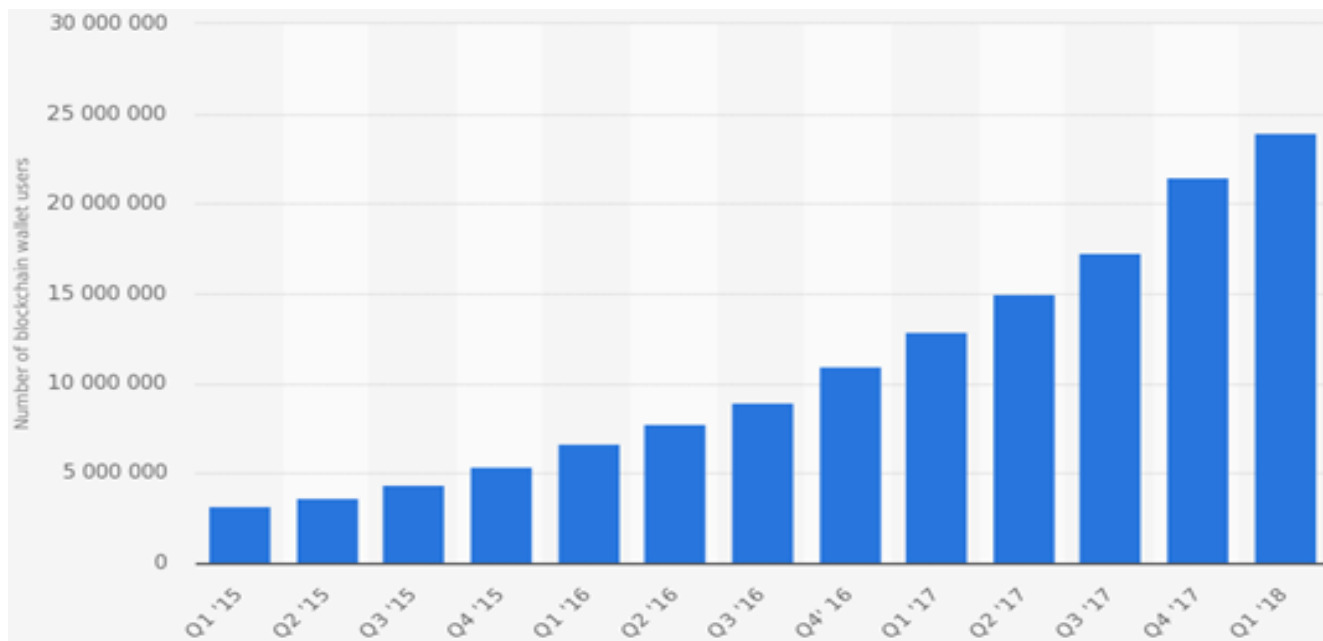


Figure 2.2 Statics depicting increase in Crypto-economy based Blockchain users from 2015-2018 [81]

2.6 Overview of the different types of attacks on Crypto-economy

2.6.1 Crypto-hacking

It is a software attack on the system itself.

A malware known as “crypto clipboard hijackers“, that attacks whenever copy + paste option (when dealing with large crypto-addresses) are used during crypto-transactions, monitors the infected computer of a victim in the clipboard software and when cryptocurrency addresses are detected, the address of the victim changes to one that the attackers control.

This may be a major risk for crypto-economy DLTs if it is applied for crypto-currency based applications.

2.6.2 51 % attack

A hub has the greater part (51%) of the system's handling power. It can control the framework, embed false transactions, double up the reserves spend, or even take a benefit from others.

2.6.3 Identity theft

In spite of the fact that secrecy and protection is saved, the security of advantages relies upon wellbeing of the private key, a type of computerized character. It is very difficult for mediator to recover back a private key if it's been acquired or snatched by someone. Frauds like key loss are not relatable to hashgraphy as it utilizes Keyless Signature Infrastructure (KSI)

2.6.4 Sybil Attack

A harmful gadget misguidedly goes up against different characters. The extra characters are called Sybil nodes. [12] Hashgraph claims to be safe from DDoS and Sybil attacks.

2.6.5 DDoS Attack

An attacker floods a node with packets, to incidentally disengage it from the web. However, work on that node may not stop and may be continued as usual. An attack like DDoS on the framework would require flooding an extensive portion $>2/3$ of the nodes with packets, which is progressively troublesome. [13]

2.7 Existing prevention and defence mechanisms

Solution to mitigate attacks on Crypto-economy:-

The attacks on crypto-economy DLTs mentioned above, can be mitigated. To increase the powerful and non-deceptive potentiality of this technology, following suggestion have been put in.

1.USING RECOGNITION TECHNIQUE

In spite of the fact that blockchain innovation anticipates misconduct, it can't recognize any offense without anyone else's input. Blockchain designers must concentrate on protecting this innovation by actualizing imaginative systems and techniques that are expected to distinguish attacks. They can utilize machine learning and information gathering calculations for making new applications, for recognizing extortion and interruptions in blockchain-based exchanges. By executing systems, for example, profiling, observing, and recognizing personal conduct standards dependent on individuals' information exchanges, scientists can create administered machine learning approaches that can help in identifying exception practices. [13]

2. BUILDING UP IDENTITY IN BLOCKCHAIN TECHNOLOGY

Utilization of cryptographic keys and mysterious exchanges can make the blockchain helpless against record takeover and digital fraud. Loss of a key is equivalent to the loss of personality on the system. One arrangement is building a character and notoriety framework utilizing a blockchain that can record "unique finger impression" events. This can likewise follow life events, for example, the opening of financial balances, vehicle purchases, and so forth. These occasions recorded in the irreversible personality can turn into a computerized character that is hard to take since it is unforgeable, openly observed, and time-stamped. Regardless of whether blockchain innovation turns out to be adequately strong to prevent malicious exercises and false assaults, any instrument and assurance characteristic in the innovation won't work except if it is broadly acknowledged and received by most of the business. [13]

Mitigation Techniques

How hashgraphy DLT can help mitigate attacks-

AVOID ATTACKS USING SWIRLDS HASHGRAPH

a. How does the Swirlds platform avoid attacks?

Imagine a community of members running a "swirld"(a particular Swirlds network) for some specific purpose, such as a public ledger. It is PoS, where consensus voting is proportional to each member's ownership of some amount of a cryptocurrency, which will be called StakeCoin for this example. The ledger swirld is open, not permissioned, so we cannot trust all the members. The ledger swirld uses PoS rather than PoW, so it costs low. Question to consider is whether it can be made secure. The system will be secure if no attacker can obtain $1/3$ of the total StakeCoin owned by all the participating members put together. The ledger swirld will continue to function as long as $2/3$ of the StakeCoin is owned by members who participate and are honest. [14]

b. In what capacity would this be able to be accomplished?

One methodology is to begin with a consortium of, state, 10 huge, regarded companies or associations that are the authors. Each is given a lot of StakeCoin to begin with, and the framework is organized so the cash supply won't develop rapidly, and will have some extreme size limit. Each originator has a motivation to

take part as a part in the record swirld and the StakeCoin swirld, where StakeCoin itself is a swirld running on a hashgraph with the Swirls accord calculation. Since there is no PoW, it is economical to be a taking an interest part running a hub. The originators are sufficiently reliable that it is impossible that any substantial part of them will connive to undermine the framework. Particularly since that would demolish the estimation of the coins they hold and the record they are running. In any case, isn't that simply like a permissioned blockchain? Yes, indeed! It appears to be like a permissioned blockchain at first. After some time, different individuals can join the record swirld. What's more, other individuals can purchase StakeCoin, either specifically from the originators, or on a trade. The record could even boost individuals to take an interest by paying small measures of StakeCoin for taking an interest, to urge more individuals to join. After some time, it could turn out to be significantly more appropriated, with the stake in the long run spreading out, so it gets troublesome for anybody to corner the market, regardless of whether the authors connived. By then, the cryptographic money will have genuine esteem, the record swirld will have genuine security, the framework will be open without permissioning, and nobody should pay the expenses of squandered PoW calculations.

2.8 Chapter Summary

This chapter presented the concept of Crypto-economy. It focusses on the risks and benefits of this decentralized concept. It also discusses the different types of attacks the cryptoeconomic world is prone to and suggested some existing attack prevention and mitigation techniques.

CHAPTER 3

Distributed Ledger Technology

This chapter covers the details of Distributed Ledger Technology (DLT), which is the main concept used in this study. Section 3.1 provides the definition of DLT. Section 3.2 gives us the SWOT analysis on DLT. Section 3.3 gives us key features of DLT. Section 3.4 explains the key advantages of DLT. 3.5 give an understanding and types of DLT categories. Section 3.6 gives an understanding of different types of DLT & explains methodology of each algorithm in detail.

3.1 Definition of a Distributed Ledger Technology

The improvement of a crypto-economy has been dependent most unmistakably on the headway of Distributed Ledger Technology (DLT), including the first and related programming and computational progressions, starting from open key cryptography.

With the improvement of the web, a growing number of united frameworks were made in various regions. With a particular true objective to develop potentiality of frameworks, it needs a usual record which can guarantee straightforwardness, irrevocable, diverse yet private. Distributed record (DL) [7] allows this type of arrangement of frameworks where in particular accomplices don't need to trust in one another yet can cooperate [8].

DLT alludes to a novel and quick advancing way to deal with account and sharing information over different information stores (or records). This innovation takes into account exchanges and information to be recorded, shared, and synchronized over a conveyed system of various system members. DLT goes ahead the impact points of a few distributed (P2P) advancements empowered by the web, for example, email, sharing music or other media documents, and web communication. In any case, web based exchanges of benefit possession have for quite some time been subtle. Moreover this requires guaranteeing that an advantage is just exchanged by its actual proprietor and guaranteeing that the benefit can't be exchanged more than once, for example no double spending. The advantage being referred to could be anything of significant worth. In 2008, an winning paper created by a so far then unidentified individual using the pen name Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", suggested a paperback philosophy for trading "resources" as "Bitcoin" in a P2P way. There has been crucial development for Bitcoin portrayed out in Nakamoto's paper was named Blockchain that alludes in a particular strategy for dealing with and securing information and trades. In this manner, distinctive strategies for dealing with information and trades for asset moves in a P2P way were thought up – provoking the articulation "Distributed Ledger Technology" (DLT) suggests more of broad class of developments.

DLT implies an ordered & speedily moving approach for managing and distributing data over different data stores (records). Each have definite and similar data records and are in general kept up and constrained by a dispersed arrangement of PC servers, which are called centres. One way of considering DLT is that it is basically spread database with specific properties (mentioned in section 3). Blockchain, a particular kind in DLT, uses cryptography and logarithmic systems to make & check while continually creating and adding data structures, that shows up as a chain of indicated 'trade squares' – the blockchain – and distributes the limit for record across the chain. However, some latest developments to database has been initiated by individuals and influences others "to ruin" data, for example containing a couple of trade records. Information related to latest data square is later distributed over entire framework, holding encoded data so trade nuances are not made open, and individuals all around the framework choose the square's authenticity as shown by a pre-portrayed algorithmic endorsement method ('accord part'). Just after endorsement, all individuals contribute Latest Square for specific records. With such techniques, every change for record is replicated over entire framework & every framework part have complete, indefinite copy of the entire record whenever. This technique can be used to record trades on any advantage which can be addressed in a propelled form. The trade could be a modification in the normal for the preferred standpoint or a trade of ownership. See figure 1.

3.2 SWOT Analysis

A nitty-gritty examination of Strength Weakness Opportunity Threats (SWOT) [15] on Distributed Ledger Technology (DLT) is shown as follows in Figure 3.1.

Strengths	Weakness
<ul style="list-style-type: none">• Distributed resilience and control• Decentralized network• Open source• Security and modern cryptography• Asset provenance• Native asset creation• Dynamic and fluid value exchange	<ul style="list-style-type: none">• Lack of ledger interoperability• Customer unfamiliarity and poor user experience• Lack of hardened/tested technology• Skills scarcity and cost• Immature scalability• Wallet and key management
Opportunities	Threats
<ul style="list-style-type: none">• Reduced transaction costs• Business process acceleration and efficiency• Reduced fraud• Reduced systemic risk• Monetary democratization• New business model enablement• Application rationalization and redundancy	<ul style="list-style-type: none">• Legal jurisdictional barriers• Technology failures• Institutional adoption barriers• Divergent blockchains• Ledger conflicts/competition• Poor governance• Politics and hostile nation sided actors

Figure 3.1: Strength Weakness Opportunity Threat (SWOT) investigation on DLT [16]

3.3 Important aspects of DLT

Individual records along many external dignitary approvals are distributed, given to, & modified through an arrangement with screened individuals been existing for a long time. Anyway the possibility for decentralized, spread & invariable records were recognized suddenly along DLT. 3 features of DLT which are ordinarily seen as important for the development are mentioned below. [17]

- the disseminated nature of the record,
- the consensus mechanism, and
- Cryptographic systems.

There should moreover be amplified where DLT isn't one all around described development. Or maybe, a dignitary somebody's share of blockchains and flowed records are dynamic or are a work in advancement today and their arrangements and correct setups change dependent upon the creators' targets and the DL's inspiration and developmental stage.

3.3.1 Nature of the Distributed Ledger

Documenting has reliably been a bound together system which needs faith in data manager. Very important development about DLT is that order above the record doesn't lie with any component yet rather is with a couple or all framework individuals – depending upon the sort of DL. This isolates it from other mechanical headways, for instance, circulated registering or data replication, which are generally used in existing shared records. Acknowledged, this suggests in a DL, no single substance in the framework can address past data segments in the records and no single component can underwrite new increments to the record. Or maybe, a pre-portrayed, decentralized assentation framework (see underneath) is used to endorse new data entries that are added to the blockchain and as such shape new segments in the record. There exists,

anytime, just a single rendition of the record and each system member possesses a full and exclusive duplicate of the whole record. Each neighbourhood expansion to the record by a system member is rapidly distributed to all hubs. After approval is acknowledged, the new exchange is added to all particular/Records to guarantee information consistency over the whole system.

This diverse element of DLT permits interested members with regards to a shared system access the entire record-confirmed information in their individual records, for instance exchange records, without depending on a confided in centrally focal gathering. The elimination of the focal party can build speed and conceivably expel expenses and wasteful aspects related with keeping up the record and consequent compromises. Imperatively, it can likewise improve security in light of the fact that there is never again a solitary purpose of assault in the whole system. To degenerate the record, an assailant needs to pick up authority over the larger part of servers in the system; destroying a solitary or a few members does not trade off the framework's honesty. Be that as it may, security chances in the product application layers based on top the DL can turn into extra assault surfaces. Shortcomings in this layer can make misfortunes the clients of a DL framework, notwithstanding when the centre innovation stays sheltered and secure. Striking points of reference that caused budgetary and reputational hurts were the hacks of Mt. Gox in Japan and Bitfinex.[17]

3.3.2 Consensus Mechanism

The scattered thought of the DL requires the individuals in the framework ('centres') to accomplish a concurrence as for the authenticity of new data entries by following a ton of principles. This is practiced through an understanding instrument that is shown in the algorithmic structure of the DL and can vacillate dependent upon its appearance, reason, and major asset. In a DL, when all is said and done any of the centre points can propose an extension of another trade to the record, in any case there are utilization which propose specific employments for centre points where only a couple of centres can propose a development. An agreement part is essential to set up whether a particular trade is genuine or not, using a predefined unequivocal cryptographic endorsement procedure allocated for this DL. The understanding instrument is furthermore indispensable to manage conflicts between various simultaneous opposition segments - for example, phenomenal trades on same asset are proposed by different centres. This framework ensures right sequencing of trades and foresee expect control by dreadful on-screen characters (by virtue of approval less DL). The understanding framework and sequencing, secure against the referenced twofold spend issue. The Bitcoin blockchain uses PoW to set up accord in a worldwide decentralized framework, a thought that was first made as an adversary of spamming measure. In order to add another block to the chain, which suggests including another plan of data sections to the record, a PoW tradition is required. This is a computational test that is hard to settle (similarly as enrolling power and taking care of time) yet easy to check. The PoW is made by again and again running single direction cryptographic hashing estimations until a progression of numbers that satisfies a predefined anyway optional condition is conveyed, expressly in the Bitcoin blockchain this is a certain number of driving zeros and the path toward delivering PoW is grouped "mining". Settling this PoW baffle is a computationally troublesome process and there are no other ways, which suggest that any single centre in the framework simply has a minutely little probability of making the required PoW without utilizing a huge proportion of costly enlisting resources. The Bitcoin structure is changed in accordance with such a degree, to the point that a real PoW is conveyed around at customary interims and in case two are made in the meantime, the tradition with the higher score is recognized as largest ("the longest chain"). Each "miner" that makes an authentic PoW in the Bitcoin compose gets Bitcoins as a reward (like a trade charge), which fills in as a money related persuading power to keep up system dependability. As such, the extensive size of open, approval less structures is essential to its security. Framework security is direct related to having a considerable number of centre points in the structure that are supported to affirm new changes to the record decisively and develop an assentation over the framework to ensure data consistency. The "affirmation of work" causes an enormous computational cost on framework individuals for keeping up the DL (for instance making new data squares and adding these squares to the blockchain), which is simply required in structures with questioned individuals. Evaluations recommend that Bitcoin miners currently eat up power indistinguishable to Ireland's capacity consumption and could accomplish Denmark's measurement by 2026 (expecting the Bitcoin assentation segment remains unaltered). As demonstrated by one check, if the Bitcoin sort out were relative to the components of use of existing portion systems like Visa and MasterCard, the power required would outperform stream overall power usage. In any case, this issue is most explained for the Bitcoin blockchain. The DLT structure used by ether, displayed propelled

monetary by Ethereum, requires basically less enlisting resources and the thought-out mechanism is significantly agile. Permissioned blockchains don't ordinarily require troublesome "PoW" as an agreement part to check trades since framework individuals are pre-picked and trusted. There are in like manner diverse understandings segments, for example PoS which rewards rank over preparing power and require a PoW regarding certain advantage. [17]

3.3.3 Cryptographic systems

Cryptography is at the focal point of DLT, explicitly for blockchain use. Each new data area, for instance a trade record, is "hashed", which infers that a cryptographic hash work is associated with the main message. A hash takes data of any size data and registers an electronic exceptional check like a human finger impression that can't be changed aside from if the data itself is changed. The hash yield is a claimed 'process' of a described length which looks subjective and immaterial to the main data yet is as a general rule deterministic. This infers for one interesting data only a solitary hash is possible and it is exceedingly improbable for another commitment to have a comparable hash value. Hashing, in like manner, applies a period stamp to the principal message. These trade hashes are accumulated into a 'trade block' that can contain any number of trades but anyway normally has an obliged steady yet variable size. The hash engages revelation of any changing of the hidden trade data, as when a hash is enlisted yet again, it will convey a surprising hash in contrast with the at first created hash. The blocks are set apart with a propelled check, which binds the sender to the content of the block, much equivalent to a fault on an agreement. DLT uses 'open key cryptography' for cutting edge marks, which is a common place system that is used in a wide show of various applications, for instance, HTTPS web tradition, for affirmation in fundamental applications and moreover in chip based portion cards. Automated imprints are comprehensively recognized as proportionate to physical stamps by law in various countries. Framework individuals each have a private key, which is used for checking modernized messages and simply known by the individual customer, and an open key which is open data and is used for supporting the character of the sender of a propelled message. Individuals when all is said in done key are moreover used to perceive the recipient. These three thoughts help clear up the nuts and bolts of DLT. The system by which data is recorded in a blockchain-based mechanism, and the passed on record is simply appended to a chain of 'trade blocks' in consecutive demand that contains hash reviews of the trades (electronic messages) to be added to the record, a proof-of-work (or another assentation instrument yield), and a propelled characteristic of the hash by the sender's private key, and open keys of the sender and the normal recipient of the trade. This chain starts with the main truly segment in the record (the 'starting block') and each additional block contains hashed information of the past square, setting the successive demand of the chain. Figure 3.3 underneath depicts an instance of a blockchain structure: The last (block $n+1$) was added to a current blockchain (Block $n-1, n$. block n being the 'starting square'). Each square contains a unique "proof of-work" tradition, a reference to the past block that chooses the privilege consecutive asking for of blocks, a movement of hashed build-ups of trade information which can't be changed, and a propelled check. In this figure, square $n+1$ addresses the most present extension to this blockchain which invigorates the record. At the point when another square is added to the chain through a foreordained assentation part, the chain can't retroactively be changed and blocks can't be deleted or redressed without re-attempting the affirmation of-work tradition for each block. This suggests as the chain grows long, this ends up being powerfully progressively troublesome in light of the way that all centre points are persistently seeing for clarifying PoW puzzles and adding new blocks to the chain.

In doing this they simply consider the trade blockchain that reflects the best proportion of computational work. Each productive extension to the affix is conveyed to the entire framework and all centre points have a cutting-edge copy of the entire blockchain.

3.4 Key Advantages of DLT

In the correct setting, appropriated records can conceivably have various focal points over customary incorporated records and different sorts of shared records. The most critical potential points of interest of DLT are recorded beneath, however speculations are troublesome as a result of the extensive assortment of plans and details about permissioned & authorization less blockchains have: [17]

- **Decentralization and disintermediation.** DLT empowers coordinate exchanges of computerized esteem or tokens between two counterparties and decentralized record-continuing, evacuating the requirement for a go-between or focal specialist that controls the record. This can convert into lower costs, better versatility and quicker time to showcase.
- **Greater transparency and easier auditability.** All framework people have a full copy of the coursed record (which can be encoded). Changes must be made when accord is set up and they are incited over the entire framework persistently. This segment, joined with the nonappearance of a central pro or confined commitment of a central master, can lessen distortion and abstain from trade off expenses.
- **Automation & programmability.** DLT enables programming pre-agreed conditions that are normally executed once certain conditions hold. This is insinuated as "sharp contracts" (see zone 8), for example sales that remuneration themselves when a shipment arrives or share confirmations which subsequently send owners benefits or cash for-work programs that pay beneficiaries out once the contracted work is done. Sharp contracts should be conceivable in regular joined record systems as well, anyway the structure of united record structures requires such exercises to be executed essentially after the concerned social occasions have agreed to the concealed trade as recorded in the central system, which in a couple of settings can take upwards of multi day. On the other hand, in a DL, the counterparties by definition agree the moment the trade is done, as both have a comparative record of the trade. Moreover, the result of the execution of the "keen contract" itself will set aside additional chance to incite and be obliged in a customary record system.
- **Immutability & verifiability.** DLT can give a changeless and obvious survey trail of trades of any modernized or physical asset. While a great part of the time, immutability is charming, it can make issues related to plan of activity segments if the structure misses the mark. Constant nature of the record, in any case, does not infer that a countervailing trade to break down a discussed trade can't be made. This is as per how question objectives capacities, for example in portion card systems. The primary record would, in any case, for this circumstance still remain. Two MIT experts have starting late recorded a patent for a cryptographic course of action that would empower a go to 'open' units in a blockchain and adjust them, anyway this is especially questionable as perpetual nature is seen as one of the middle inclinations of the first blockchains.
- **Gains in speed and efficiency.** DLT offers the capacity of growing rate and cutting down inefficient perspectives by emptying or diminishing design frames in trades or in clearing and settlement shapes by removing mediators and automating design frames.
- **Cost decreases.** DLT offers the potential for basic cost reductions due to emptying the prerequisite for bargain as DLT-based systems by definition contain the "shared truth" and in this manner there is no convincing motivation to oblige one interpretation of "truth" with that of one's counterparties. Additional pros of cost decline could be cut down system costs for keeping up a DL, similarly as concessions in frames and distortion. As shown by a couple of assessments, passed on record development could save the budgetary business alone around \$15-20 billion for consistently.
- **Enhanced cybersecurity flexibility.** DLT can possibly give a stronger framework than conventional incorporated databases and offer better insurance against various kinds of digital assaults in view of its disseminated nature, which expels the single purpose of assault. On an essential dimension, DLT is an elective arrangement approach that considers a decentralized business and operational model when diverged from existing, concentrated structure approaches that can be used for near purposes. This makes possible a progressively unmistakable game plan of automation, faster getting ready, and increasingly important flexibility potential. In clear settings, a DLT-based arrangement approach can give a noteworthy number of the points of interest analysed beforehand. The underneath case for a security vault shows the contrast between DLT-based methodologies and elective structure approaches. Setting up a guaranteed agenda existing, united philosophies requires a central component to setup a dedicated stage, develop cooperation criteria, and set up rules and methods. All trades identifying with the security are taken care of on this stage and all business exercises are actuated by the united stage. This stage is made using standardized programming applications delivered for the specific business require or custom made. A DLT-

based methodology, conversely, highlights exchanges including insurance those are traded on a shared premise, with implanted, pre-decided conditions, for example, released data and standards relating to inability to reimburse a fundamental credit. There is no compelling reason to setup any concentrated framework and the business rules relating to specific insurance can be customized dependent on the particular understanding between counterparties. In a permissioned DL, there can be an executive that sets up interest criteria and on sheets new members. In any case, rather than the incorporated element in a conventional execution, the job of the head in a DLT-based framework would be negligible. Business exercises can be event driven and can be initiated with no prerequisite for additional external mediations. Setting up another assurance library using a DLT-based strategy can possibly be snappier and continuously versatile as the benefits required at the administrator level are incredibly irrelevant, the handling load is spread over all members, and the business rationale for insurance exchanges can be custom-made and redid dependent on the particular needs of the counterparties.

3.5 Categories of DLT

Passed on record briefs can be open (approval less) or permissioned, and there are essential differentiations between the two. Bitcoin and Ethereum are the most obvious occurrences of absolutely approval less blockchains, where sort out individuals can join or leave the framework openly, without being pre-attested or considered by any component. All that is required to join the framework and add trades to the record is a PC with the critical programming. There is no central owner and random copies of the record are flowed to all framework individuals. In permissioned DLs people are pre-picked by someone – an owner or a regulator of the record – who controls access and sets the rules of the record. This unwinds for different concerns governments and controllers have about approval less passed on records, for instance, identity affirmation of framework people, whom to allow and oversee, and real obligation regarding record. Regardless, it also reduces a primary favoured outlook of permission less blockchains: the ability to work without the prerequisite for any single substance accepting an arranging work, which on a very basic level requires distinctive individuals to trust in this component. Regardless, even in permissioned DLs, when all is said in done there is no necessity for a chief for the execution of trades. Permissioned DLs, which coordinate framework get to, ordinarily don't require a figuring power-concentrated confirmation of-work to affirm trades anyway rely upon different algorithmic models to set up accord among people. In permission less DLs, which don't oversee orchestrate access, there is no need of any trust between the individuals and a tangled proof of-work is from this time forward used to create understanding about record sections. Strangely, by virtue of a permissioned DL, the administrator bears the commitment to ensure that the individuals in the DL are strong. In permissioned DLs, any centre point can propose a development of a trade, which is then rehashed to various centres, perhaps even with no understanding segment. In reality, this is unquestionably not a twofold course of action anyway the dimension of straightforwardness and decentralization of appropriated record systems falls on a range with totally open, approval less blockchains, for instance, Bitcoin toward one side of the range and permissioned blockchains encouraged by private substances on the other, and the correct features change from time to establish. DLT designs can be portrayed the extent that different estimations: access to the framework (open/close) versus occupations inside the framework (constrained/boundless) – see logical order in Figure 5. Various associations use a dignified strategy where they give the advancement to permissioned frameworks to be founded on open blockchain establishment and thusly limit employments in a DLT structure with open access. Some industry players make a capability between open/private (with respect to get to) and permissioned/permission less (to the extent employments) spread records. Surge, for example, has a permissioned record anyway the data is endorsed by all individuals; along these lines their structure can be seen as an open, permissioned record. A permissioned DLT where the data is affirmed just by a great deal of individuals would be seen as a private, permissioned record. More than likely, both open DLs and permissioned DLs will have significant applications. The development is still at a starting time of headway and there are particular future circumstances: some trust the business will at last meet to one generally speaking open blockchain (like one in general web) and a wide scope of private blockchains (much equivalent to different private intranets), while others believe that few open blockchains will continue existing beside one another. At first, the web was a home of information, which had the effect of democratizing access to information. A possible future circumstance of the blockchain could capturing of critical information, democratizing access and limit of automated assets. [18]

Since Bitcoin's start in 2009, in excess of 600 unmistakable open and private coursed record frameworks have risen, anyway only a pack have achieved scale and a further created period of enhancement. Most blockchain applications (see underneath) depend on open blockchains – commonly Bitcoin and Ethereum. The Committee on Payment and Market Infrastructures (CPMI) of the Bank for International Settlements (BIS), in its progressing generation on DLT proposed a demonstrative structure for looking at DLT applications in portions and settlements. This is, in any case, a summed up framework and is significant for a wide scope of uses of DLT in the monetary fragment. The framework proposes the going with various nonexclusive occupations for a centre point:

- **System executive:** This activity incorporates picking who can get to the framework, keeping up and coordinating discussion objectives measures and performing lawful authority limits. This activity isn't required in permission less DLT.
- **Asset backer:** The centre points accepting this activity are responsible for issuing new "tokens" used in the framework. In the Bitcoin blockchain, there is no component accepting this activity, the system itself makes new bitcoins reliant on express models. A token is a depiction of an automated asset. It routinely does not have innate regard yet rather it is associated with the fundamental asset, which could be anything of huge worth.
- **Proposer:** This activity incorporates proposing new trades for linking in with the record.
- **Validator:** This activity incorporates favouring requesting for development of trades in the record. In permission less DL, this activity is performed by a decentralized agreement part.
- **Auditor:** Permitted to see the record yet not allowed to make changes. This could be used for performing audits and besides be used by controllers and chiefs. Money related organizations, which are overwhelming clients of databases, are so far not indicating much enthusiasm for open, authorization less blockchains because of the trouble of consenting to existing administrative and consistence structures. Further worries by the money related division identify with the open access and the trouble of personality check in consent less frameworks, which are frequently inconsistent with existing business rehearses that require keeping up security of exchanges. Monetary establishments are making critical ventures into looking into permissioned DLs as a mechanical answer for lessening costs and expelling grindings in cross-outskirt instalments, journalist managing an account, clearing and repayments forms, syndicated advances and exchange fund.

3.5.1 Open or Permission-less DLT

This kind of blockchain is open and is available to all. The member must have assets like processing force and programming to approve exchanges. Bitcoin and Ethereum are equivalent instances. In basic words, it implies that anybody in the system (hubs) can join the system and approve the blocks; anybody can study the chain and add new blocks to it.

3.5.2 Permissioned DLT

This sort of framework depends on a consortium of trusted validators. One needs the approval to study the data in the chain which isn't the situation with permission less blockchain. One of the real disadvantages which hold the majority of the organizations to make permission less blockchain a piece of their venture arrangement is that it requires enormous processing capacity to accomplish agreement. Every hub in the system takes care of a complex numerical issue with the verification of work accord component to guarantee exchange legitimacy. The purpose of concern is the straightforwardness of the framework. It is noticeable to everybody which makes the authorization less blockchain an uncertain undertaking for the organizations. With regards to the permissioned framework, at that point it is very adaptable. The consensus models can be based on the evidence of-stake convention. In spite of the fact that the member get to is the key differentiator between the permissioned and authorization less framework, both are similarly effective and share comparative characteristics. It likewise adds to the upsides of both these frameworks.

3.6 Distributed Ledger Technologies (DLT) Types & Methodology of each Algorithm in detail.

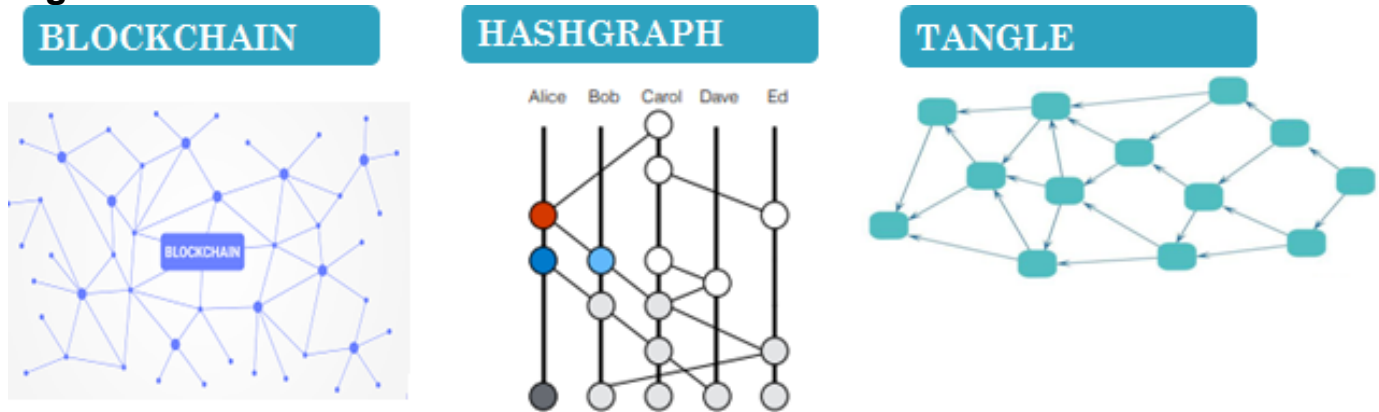


Figure 3.2 Structure of Blockchain[71], Hashgraph[72], Tangle[73].

Distributed Ledger Technologies (DLT) is primarily classified into the following major Algorithm's: [19]

Blockchain

A blockchain system relies upon Distributed Ledger Technology (DLT). It handles and offers trade records over an consensus of customers. An exchange contains an identifier, information, Output and a timestamp. Preceding an exchange being enlisted in the blockchain it is constantly checked and grasped by the mediators of the structure. On demand it is analysed if two hubs directing trade related data between each other is real as demonstrated by the tradition.

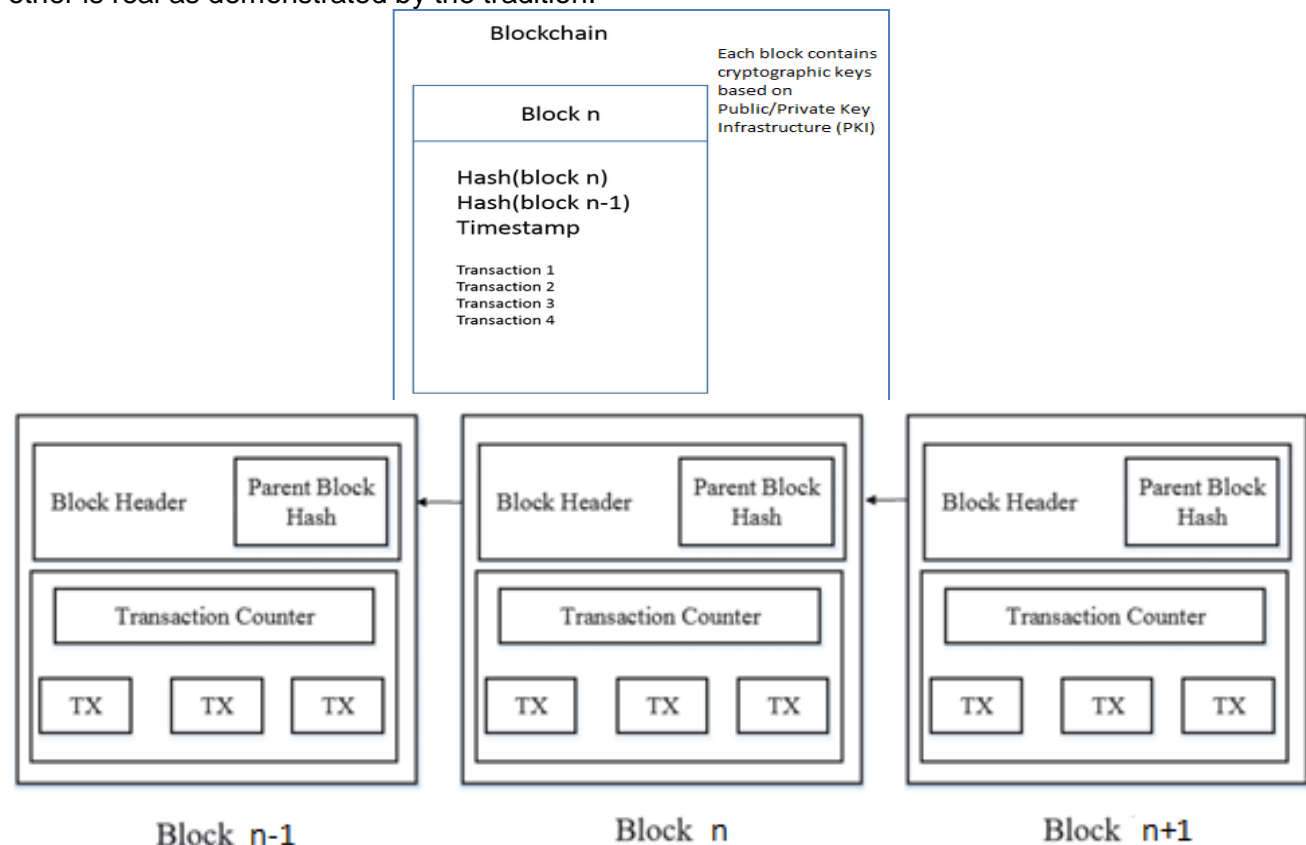


Figure 3.3: A block structure and formation of a chain of blocks [11]

The Figure 3.3, illustrates a sample creation of a block and how the blocks come together to form a chain. In approval, which block will be appended to the blockchain is chosen on the basis of consensus

methodology. The last block of consensus later will be connected to blockchain. Cryptography engages the traits of ceaseless quality, lack of definition and change opportunity. The two rule used thoughts of cryptography are hashing and verified sign impression. A blockchain uses hashing to relate blocks. As an agreement, signatures gets used for public as well as personal or private keys. Private Key being confidential to the customer is always required to acknowledge exchange and open key to check and open information of user.

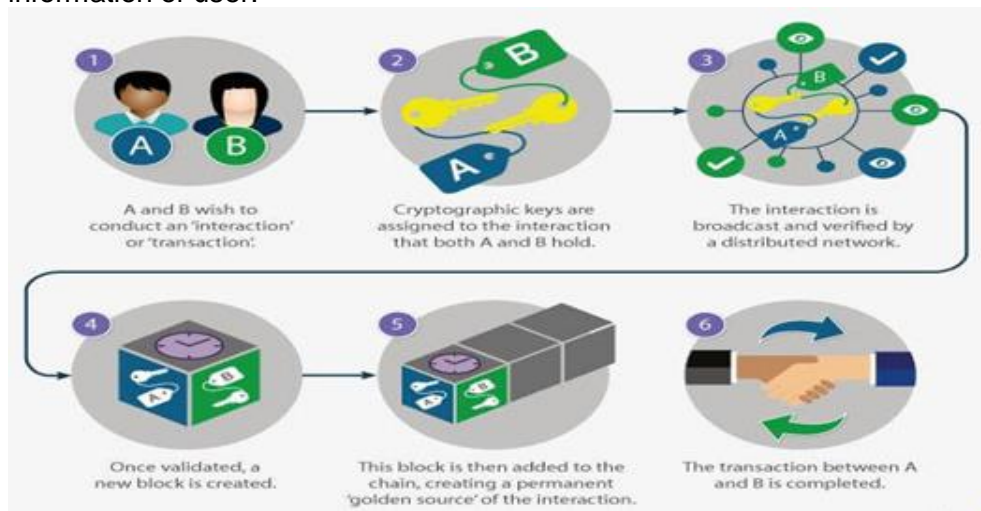


Figure 3.4 Blockchain transaction

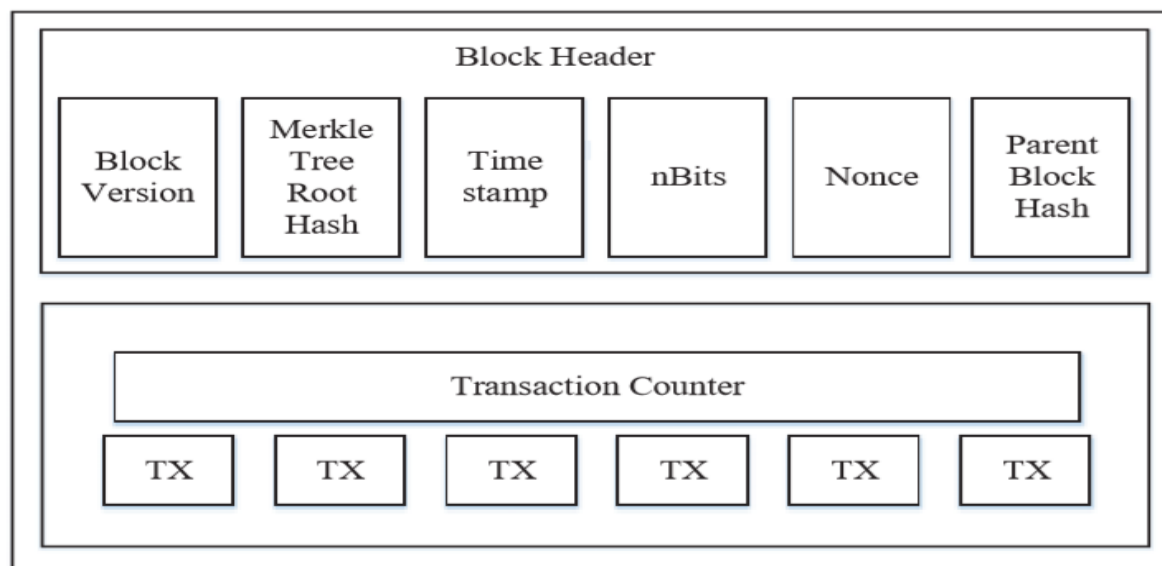


Figure 3.5: Block structure [20]

A block involves the block header and the block body as showed up in Figure 3.5. In particular, the block header highlights:

- (I) Block version: exhibits which set of square endorsement fundamentals to seek after.
- (ii) Merkle tree root hash: the hash estimation of the extensive number of trades in the square.
- (iii) Timestamp: current time as seconds in across the board time since January 1, 1970.
- (iv) nBits: target limit of a genuine square hash.
- (v) Nonce: a 4-byte field, which typically starts with 0 and additions for each hash figuring.
- (vi) Parent block hash: a 256-piece hash regards those concentrations to the past square.

The block body is made out of a trade counter and trades. The best number of trades that a block can contain depends upon the block size and the degree of each trade.

The Table 3.1 depicts the varying groupings of blockchains to be explicit, Public Blockchain, Consortium Blockchain, and Private Blockchain. Furthermore, the capabilities and likenesses between every sort are

helpfully delineated. This would give a pervasive valuation for which type of blockchain is better appropriate and for what reason.

Table 3.1: Types of Blockchain Networks; Public-type, Consortium-type, Private-type mechanisms [21]

Public	Consortium	Private
<ul style="list-style-type: none"> Participation in a network (building a consensus and conducting mining) is open to anyone. 	A blockchain is used while building a consensus only among members who can be trusted with each other to some extent, such a member of a specific company group.	A blockchain is used only within a specific organization.
<ul style="list-style-type: none"> Methods of building a consensus are important in order to eliminate malicious participants. 	Building a consensus is easier as participants are all identified.	Building a consensus is quite easy as the mechanism is open only to the relevant organizations.

Open Blockchains

Anyone can examine an open blockchain, send trades to it, or appreciate the agreement system. They are seen as "permission less." [6] Every trade is open, and customers can remain mysterious. Bitcoin and Ethereum are discernible examples of open blockchains. [22]

Private Blockchains

Private Blockchains are constrained by a singular affiliation that makes sense of who can examine it, submit trades to it, and share in the understanding procedure. Since they are 100% bound together, private blockchains are important as sandbox conditions, anyway not for genuine creation. Semi-private blockchains are constrained by a lone association that stipends access to any customer who satisfies pre-developed criteria. Regardless of the way that not truly decentralized, this kind of permissioned blockchain is drawing in for business-to-business use cases and government applications. [23]

Consortium Blockchains

In a consortium blockchain, [24] the understanding method is constrained by a pre-chosen group – a social gathering of organizations, for example. The benefit to examine the blockchain and submit trades to it may be open or constrained to individuals. Consortium blockchains are seen as "permissioned blockchains" and are most proper for use in business. [24]

Comparison between the three types of blockchains

Table 3.2 gives us the correlation among the three sorts of blockchains. [23]

Property	Public Blockchain	Consortium Blockchain	Private Blockchain
Consensus determination	All miners	Selected set of nodes	One organization
Read permission	Public	Could be public or restricted	Could be public or restricted
Immutability	Nearly impossible to tamper	Could be tampered	Could be tampered
Efficiency	Low	High	High
Centralized	No	Partial	Yes
Consensus process	Permission-less	Permissioned	Permissioned

Each property as mentioned in table 3.2 is described as follows:

- Consensus confirmation. Out in the open blockchain, each centre could take part in the understanding methodology. Besides, only a picked set of centre points are responsible for favouring the square in consortium blockchain. Regarding private chain, it is totally constrained by one affiliation and the affiliation could choose the final consensus.
- Read permission. Trades in an open blockchain are clear to the all-inclusive community while it depends concerning a private blockchain or a consortium blockchain.
- Immutability. Since records are secured on a sweeping number of individuals, it is practically hard to change trades in an open blockchain. In a surprising way, trades in a private blockchain or a consortium blockchain could be adjusted viably as there are required amounts of individuals.
- Efficiency. It requires a great deal of speculation to spread trades and blocks as there are an extensive number of centre points on open blockchain mastermind. Thusly, trade throughput is limited and the inactivity is high. With less validator, consortium blockchain and private blockchain could be more efficient.
- Centralized. The key refinement among the three sorts of blockchains is that open blockchain is decentralized, consortium blockchain is deficiently united and private blockchain is totally bound together as it is constrained by a single social occasion.
- Consensus process. Everyone on the planet could join the understanding method of individuals as a rule blockchain. According to publications in connection to open blockchain, both consortium blockchains and private blockchains are permissioned. [23]

With a fantastically spread out data collecting structure, trades in Bitcoin network[24] could happen with no unapproachable and inside advancements to gather Bitcoin to being a blockchain,[25] as first proposed in 2008 and afterward completed in the year 2009 [26]. Every single vital rely upon the planet is by and by investigating the use related to blockchain development. Every single important rely on the planet is starting at now investigating the usage of blockchain progression. In August 2016, UBS, Deutsche Bank, Bank of Santander and Bank of New York Mellon together winning with respect to making electronic cash structure along blockchain movement to connect with budgetary trade to upgrade the part pace. Bank of Santander, the best bank in Spain, expects if all banks on the planet use the blockchain, they can save about \$20 billion reliably. World Economic Forum figures about 10% of the world's Gross Domestic Product (GDP) will be secured on the blockchain sort before 2027. [27] [28]

Blockchain can be used in various financial relationship, for instance, modernized assets, settlement and online part [29], [30]. Likewise, it can in like way be connected into various fields including smart contracts [31], open affiliations [32], Internet of Things (IoT) [33], hypothesis structures [34] and security affiliations [35].

The examination underneath in Figure 3.6, shows the surge of genuine Banks enthusiasm for Blockchain development for the next couple of years. This in like manner portrays the proportion of trust budgetary fragments have in Blockchain.

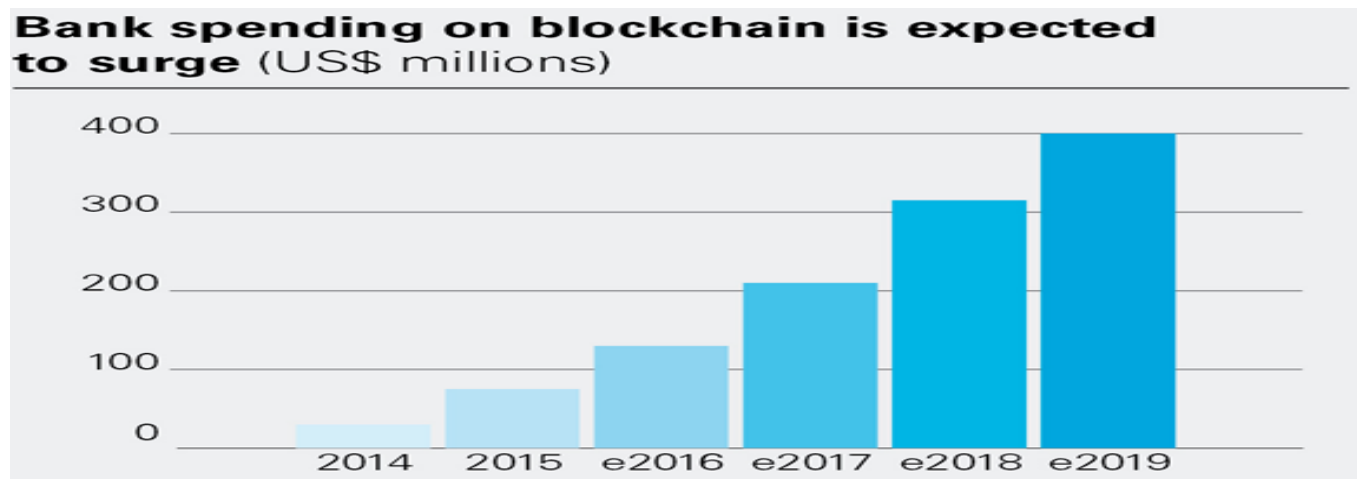


Figure 3.6: Gradual increase of Banks interest in Blockchain innovation from 2014 to 2017[24]

Consensus Algorithms

Blockchain Consensus

It isn't sufficient to guarantee that each part knows each occasion. It is likewise important to concur on a direct requesting of the occasions, and along these lines of the exchanges recorded inside the occasions. Most Byzantine's adapt to internal failure mechanisms and rely upon individuals transmitting internal votes, without a leader. Therefore to achieve a solitary Yes/No inquiry n individuals may need $O(n^2)$ to cast vote which has to be transmitted in the system, because each part informs every other about their vote. A portion of such conventions needs proof about the votes been sent over to all, so they become $O(n^3)$. What's more, they may require various rounds of casting a vote, which further expands the quantity of casting vote messages sent. [36]

Table 3.3. Typical consensus algorithms comparison

Property	PoW	PoS	PBFT	DPOS	Ripple	Tendermint
Node identity management	Open	Open	Permissioned	Open	Open	Permissioned
Energy Saving	No	Partial	Yes	Partial	Yes	Yes
Tolerated power of adversary	<25% computing power	<51% Stake	<33.3% faulty replicas	<51% validators	<20% faulty nodes in UNL	<33.3% byzantine voting power
Example	Bitcoin	Peercoin	Hyperlodger Fabric	Bitshares	Ripple	Tendermint

The consensus algorithms concepts as shown in Table 3.3 are described as follows:

Concept Proof of Work in Blockchain

PoW (Proof of work) being an master plan procedure is utilized in the Bit coin arrangement . For suburbanized system, a character must be chosen for noting about trades. To do this one of most effortless way is subjective choice. In any case, irregular choice is helpless against assaults. Now if a hub requires distributing a block of exchanges, lot of effort is done to demonstrate that the hub isn't probably going to assault the system. For the most part the work implies PC estimations. In PoW, every hub in system is ascertaining hash estimation about the header. Block header contains a nonce and miners would change the nonce much of the time to get diverse hash esteems. Agreement necessitates about the determined esteem should be equivalent or littler than a specific given esteem. When one centre point accomplishes the goal regard, it would convey the block to various centre points and each and every other centre point ought to regularly confirm the rightness of the hash regard. If the block is endorsed, distinctive diggers would join this new block to their own special blockchains. Centre points that figure the hash regards are called miners and the PoW system is called mining in Bitcoin. In the decentralized framework, considerable blocks might be delivered in the meantime when various centres find the sensible nonce nearly meanwhile. Consequently, branches may be delivered as showed below in Figure 3.7 . In any case, it is inconceivable that two fighting forks will create Next block in the meantime. In PoW tradition, a bind that ends up being longer from that point on is settled on a choice as the real one. Consider two forks made by at the same time endorsed blocks U4 and B4. Miners keep mining their block until the point that an increasingly drawn out branch is found. B4, B5 shapes an increasingly broadened chain, so the miners on U4 would change to the more expanded branch. Miners then need to finish a huge amount of PC calculations in PoW, yet these works waste unnecessarily resources. To lighten the hardship, some PoW traditions in which works could have some side-applications have been organized. For example, Prime-coin [19] searches for remarkable prime number chains which can be used for logical research.

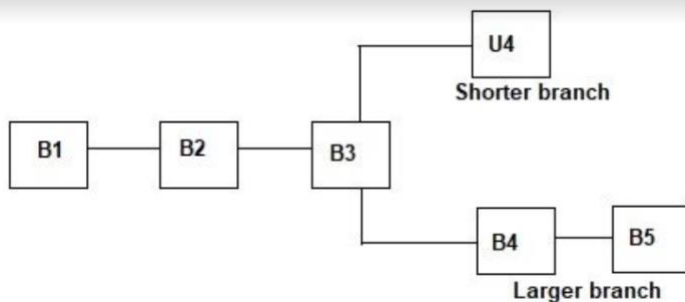


Figure 3.7: Structure of blockchain branches

Concept of Proof of Stake in Blockchain

PoS (Proof of stake) is a vitality sparing option in contrast to PoW. [37] Miners in PoS need to demonstrate the responsibility for measure of money. There's a belief about individuals that have more monetary forms would be more averse for assaulting the system. The decision subject to record balance is entirely unmerited because the absolute most lavish individual will without a doubt be overpowering in the framework. In this manner, various courses of action are proposed with the blend of the stake size to pick which one to form the accompanying block. In particular, Blockchain uses randomization to envision the accompanying generator. It uses a formula that looks for the most diminished hash, and gives an incentive in mix with the degree of the stake. Peer coin favours coin age based assurance. In Peer-coin, increasingly prepared and greater courses of action of coins have a progressively imperative probability of mining the accompanying block. As stand out from PoW, PoS save greater imperativeness and are continuously effective. Disastrously, as the mining cost is around zero, attacks may come as a result. Various blockchains grasp PoW toward the begin and change to PoS relentlessly. For instance, ethereum is wanting to move from Ethash (a kind of PoW) [39] to Casper (a kind of PoS) [40].

Proof of Stake is one of the most used parts in understanding traditions inside blockchain advancement. This proof of stake and how it functions truly has been exhibited in this thesis.

Proof of stake is the agreement estimation used by advanced monetary forms to favour blocks. In 2011 the structure was proposed first and then in 2012 the essentials for advanced monetary was implemented. Security & imperativeness capability are the main advantages of proof of stake.

To resolve issues related to Byzantine Fault Tolerance, we use proof of stake agreement, as the network allows us to trace all the validators that are present & those who have familiar identities. More than half of validators needs to be real & honest as required by Byzantine Fault Tolerance, checking these individual identities keeps up a pragmatic existing condition.

Unplanned system helps to determine the blocks who have created for proof of stake organization, which in part, gives us information about the number for cryptocurrency individual user holds & for most cases it also tells for how much time is the currency being retained. Instead of computational influence, like the case in PoW, the probability of making a square and getting the related prizes is comparing to a customer's holding of the underlining token or cryptographic cash on the framework.

The randomization in a proof of stake structure envisions centralization; for the most part the most luxurious individual in the system would constantly be making the accompanying square and dependably growing their wealth and in this way their control of the structure. The essential favoured angle of check of stake, over a structure, for instance, proof of work, is that it uses widely less imperativeness and along these lines is all the more fiscally aware. It is all around announced that each Bitcoin exchange, which uses a proof of work structure, can require as much power as a typical Dutch nuclear family does in around fourteen days. This is both inadequate and unsustainable.

In such way affirmation of stake can be seen as a prevalent understanding tradition as it requires far less capacity to run. In addition, as the confirmation of stake structure is significantly increasingly pragmatic there is to a lesser degree a need to release an unnecessary number of new coins as a strategy for boosting diggers to keep up the framework. This keeps the expense of a particular coin continuously consistent.

PoS tradition is amazing in not simply encouraging individuals to partake in the structure yet furthermore shielding any individual from controlling the framework. In order to finish a 51% attack an individual or social event would need to have the bigger piece of coins on the framework.

At first, it would be exorbitant to get enough coins to go wherever close doing in that capacity since various individuals would likely leave the cash if a lone get-together begun buying everything, while others would expand the expense to dampen an unpleasant takeover. Additionally, it would be absolutely counterproductive to attack the framework as it would unbelievably reduce the estimation of the coins that the attacker is holding. Essentially, the customers with the most significant stake in advanced cash have the most eagerness keeping up and tying down the framework because any ambushes would diminish the reputation and cost of the cryptographic cash that they hold.

In any case, affirmation of stake has its downsides, one of them being a "nothing being referred to" issue. Block generators supports fluctuating blockchains, where the consensus fails in the events that occur, which then prevents the difference of resolving the issues.

All things considered, the affirmation of stake accord tradition is a lively structure that effectively and capably fulfils its proposed reason. Regardless, this has not kept associations from changing and upgrading the tradition. [53]

Tangle

Refinement of a tangle to blockchain is accessible in the information piece which suggests that on execution surface Directed Acyclic Graph (DAG) framework may likewise hold blockchain parts.

As showed up in the framework in Figure 4, the tangle is a DAG in which within centres address trades and the edges show the course of clarification between two trades. [41]

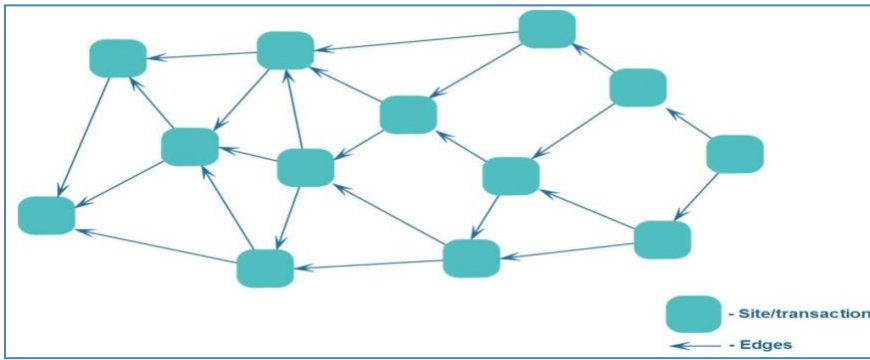


Figure 3.8: Structure of Tangle [41]

The validation of trades is not synchronous which awards parallel support and no specific time periods between affirmations. In context of this, the period of trade accreditations and conviction of trades i.e., demonstrates the endorsement dimension of trades. Tangle trades are critically essential.

In case it is roundabout connected with all centre's that have no moving toward composed edge (tips) are down and out to the level of the tangle. [41]

The tangle is what is known as a Directed Acyclic Graph (DAG): a data structure that moves in a solitary bearing without hovering back onto itself. Like the blockchain and hashgraphy, the tangle is a circled record, in which an arrangement of independent records performs trades among themselves, accomplishing understanding about who has what without depending upon incorporated expert. In the tangle, every contraption endeavours to keep up the record. Every hub has some degree of mining.

Here's the way it works: each time a hub needs to trade some esteem, it must support two past exchanges. This endorsement requires proximity of PoW remembering that the ultimate objective to stay is the framework, inferring that exchanges are not by any means free. Since there is no unmistakable class of miners that must be reviewed, in any case, there are no charges. As a tangle exchange gets supports, and the exchange supporting it receives endorsements consequently, the "total weight" of that exchange creates. Like certifications for a bitcoin trade, higher total loads show even more constantly perpetual exchanges.

Tangle's key goal dependent on IoT is making DL arrange for Internet-of-Things (IoT). With a theorized 18 billion gadgets by 2022 [27], Internet of Things (IoT) has transformed into an advancement with broad effect transversely over various vertical markets.

The graphical depiction underneath in Figure 3.9, exhibits the surge of number of Internet of Things customers in a scope of six years starting 2013 to 2019. This is a positive sign for Tangle system as it relies upon the possibility of IoT. In addition, the speedy advancement of development gadget customers shows a promising and splendid future for Tangle

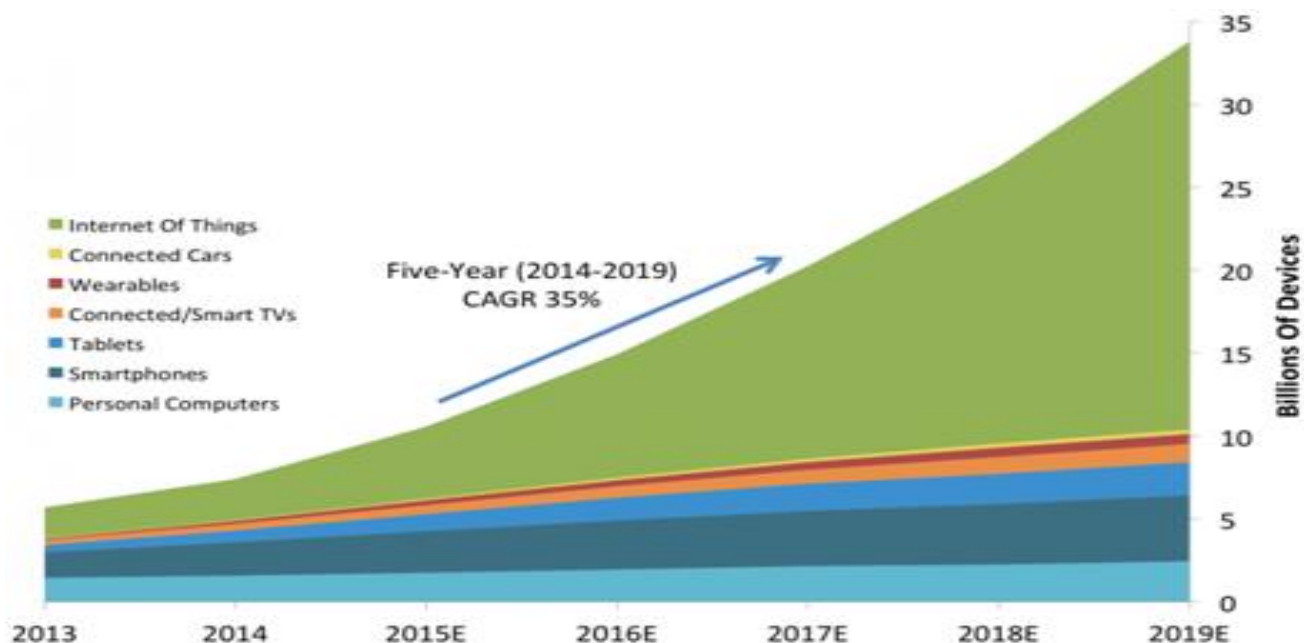


Figure 3.9: Rise in number of Internet of Things users from 2013 to 2019 [42]

Hashgraph

Hashgraph is an approved estimation on DaG to enable distributed methodology with DDoS attack resistance, high exchange throughput, low unresponsiveness and reasonable all around requesting of exchanges. Also, it is non-concurrent and non-deterministic achieving simultaneousness with probability [29]. It likewise utilizes a gossip convention that detects an exchange with respect to a chatting technique. Most recent certainties will be spread out over the system by carelessly assembling individuals and posting every one of them with all the data. The rationale is to start dialog and convey the hashgraph without anyone else's input. It solidifies each part's exchange occasions which guarantees Byzantine Fault Tolerance. No under 0 exchanges can be encouraged in the charge of occasion which actuates Aek to not mean the exchanges rather the occasion itself. As section of hashgraph the most recent exchange is in a split second sent over the structure. In light of the hashgraph, in the memory Ben can figure the vote of Aek to achieve Byzantine concurrence with no transmission limit being utilized. Also, if Aek and Ben can enrol Cate's virtual vote both find a similar course of action on demand. This is the key component of the legitimate certification of BFT along probability one.

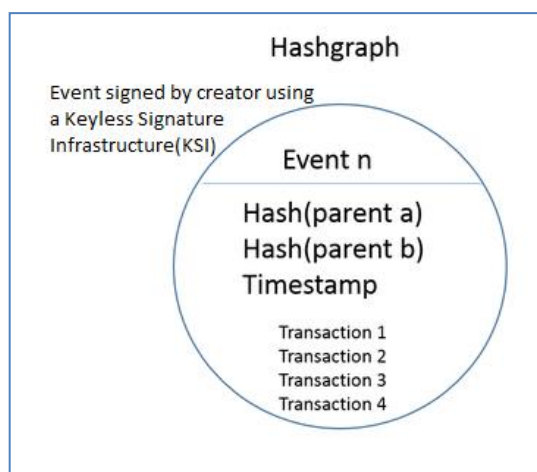
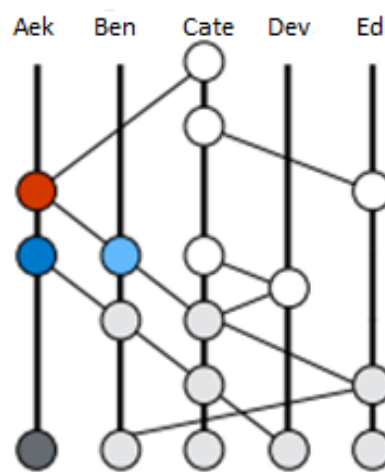


Figure 3.10: Event specifications of Hashgraph



3.11: Structure of Hashgraph [43]

As appeared in the above outline in Figure 3.10, in a Hashgraph network, at whatever point a member gossips (i.e., conveys their information regardless the information received) about gossip (i.e., information

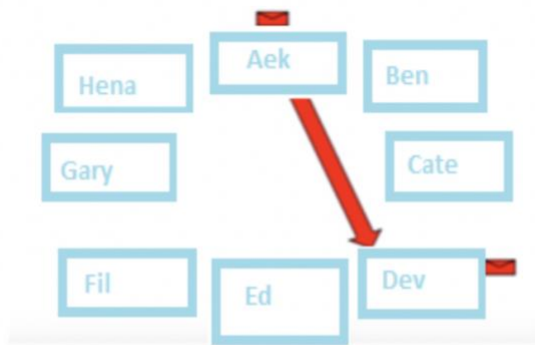
acquired), they make an occasion called Event . This event is set apart by the member and relies upon the possibility of Keyless Signature Infrastructure (KSI). Also, the event contains a timestamp, the exchanges being conveyed, and two hashes called gossip (information of when and from whom the last information was obtained).

As depicted in Figure 3.11, Ben (light blue) gossips to Aek and uncovers to her everything that he knows. Aek directly makes an event (red). This event contains: Hash of light blue (data acquired), Hash of dim blue (information Aek starting at now had), current information, timestamp. Just by including these two hashes- this entire graph is in memory. Likewise, everybody approaches the graph, of exactly, how every part chatted with one another part. Aek signs this new event (red) when made. Moreover, presently this new occasion is gossiped erratically to various members. This is the methods by which a hashgraph is formed.



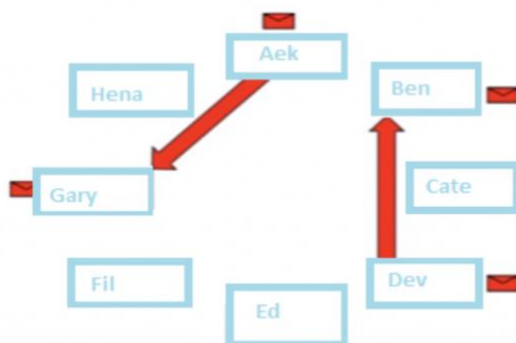
Aek has a message

Figure 3.12 : Step 1 of working of Hashgraph



Aek picks someone in random , in this case it is Dev . She sends the message to him. Now two people have the message.

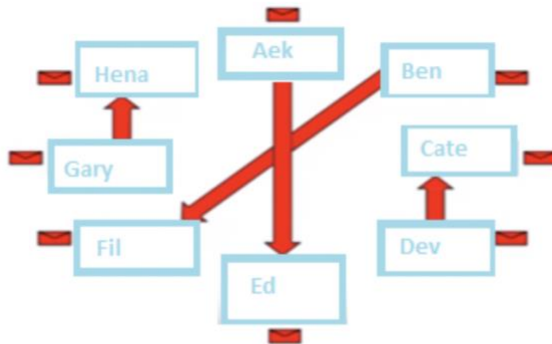
Figure 3.13 : Step 2 of working of Hashgraph



Now, they both pick someone at random. Aek picks Gary and sends the message, Dev picks Ben and sends the message.

And now, four people have the message.

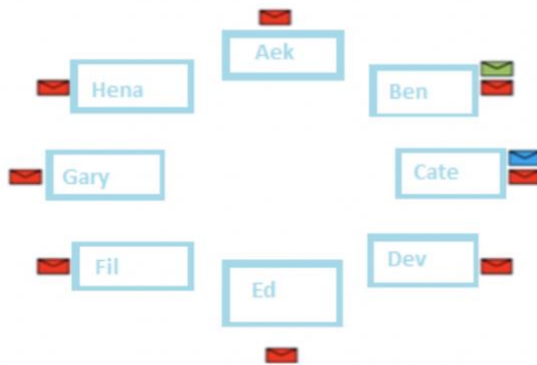
Figure 3.14 : Step 3 of working of Hashgraph



The four people randomly pick different people to send the message. Now, 8 people have the message. And then 16 people will have it, then 32 people and so on. Hence hashgraph is exponentially fast.

Even if one computer is down, it still sends messages exponentially fast and doesn't hurt the functioning in any way. It is incredibly resilient and there are no bottlenecks. There is no one person (leader) who will get the information and distribute it to everyone else.

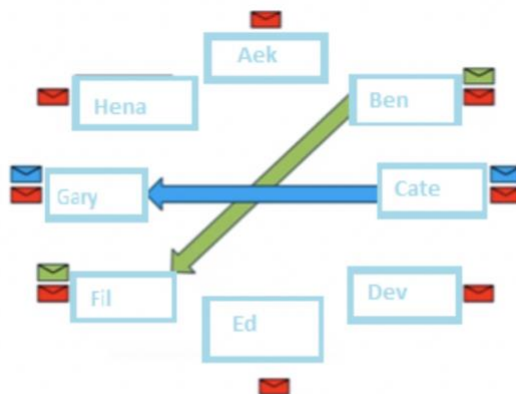
Figure 3.15 : Step 4 of working of Hashgraph



Ben and **Cate** want to send some information at the same time.

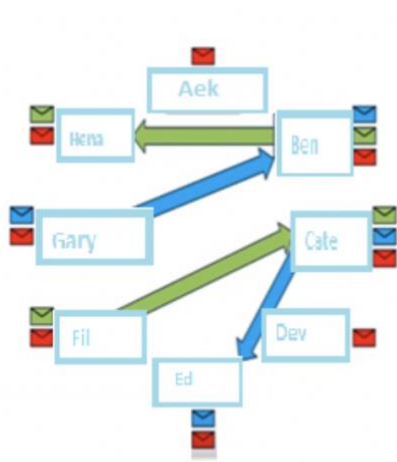
One approach could be they take turns. **Ben** sends it first and **Cate** can send it next. But this would be slow.

Figure 3.16 : Step 5 of working of Hashgraph



Rather than taking turns, the best approach would be that **Ben** picks someone at random. **Cate** picks someone at random and they both can distribute their message at the same time.

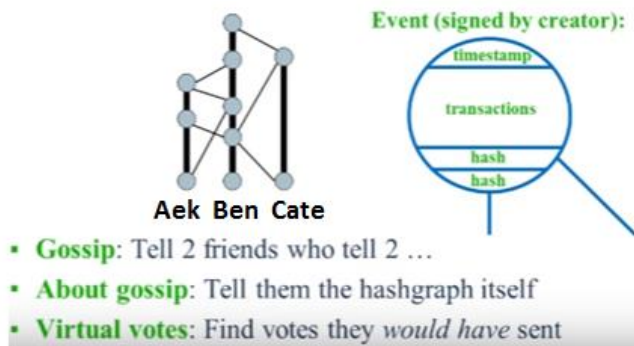
Figure 3.17 : Step 6 of working of Hashgraph



Hashgraph claims to facilitate the fastest transactions over the internet, with each person distributing information with a signature and a timestamp.

Figure 3.18 : Step 7 of working of Hashgraph Figure 3.19 : Step 8 of working of Hashgraph [36]

Gossip about gossip with virtual votes



Information is distributed in the most efficient way

+

A tiny bit of extra information is added to it

Figure 3.20 : Step 9 of working of Hashgraph [72]

Hashgraph Consensus

The hashgraph consensus calculation is totally non-concurrent, is nondeterministic, and accomplishes Byzantine concurrence with probability 1. Hashgraph consensus does not utilize a leader, and is versatile to repudiating of administration assaults on little subsets of the individuals. Up to just less of 1/3 of the individuals can be untrustworthy, they can conspire, and they can erase or postpone messages between legitimate individuals without any limits on the message delays. The assailants can control the system to postpone and erase any messages, however whenever, if a legit part over and again sends messages to another part, the assailants should in the end permit one through. It is expected that safe computerized marks exist, so assailants can't imperceptibly alter messages. It is expected that safe hash capacities exist, for which crashes will never be found.

```

run two loops in parallel:
  loop
    sync all known events to a random member
  end loop
  loop
    receive a sync
    create a new event
    call divideRounds
    call decideFame
    call findOrder
  end loop

```

Figure 3.21. The Swirlds hashgraph consensus calculation. [72]

Every part more than once calls different individuals picked indiscriminately, and adjusts to them. In parallel with the active adjusts, every part gets approaching matches up. At the point when Aek adjusts to Ben, all the occasion is shared to Ben when Aek perceives that ben is not aware of. Ben then adjoins such occasion with hashgraph, accepting just occasions to legitimate marks containing substantial hashes of parent occasions he has. Every single realized occasion are then partitioned into rounds. At that point the first occasions by every part for every single encircle ("spectators") gets selected based on whether they are well known, via absolutely nearby Byzantine concurrence along virtual voting. At that point the absolute request is found on those occasions for which enough data is accessible. On the off chance that two individuals autonomously allocate a situation in history to an occasion, they are ensured to distribute a similar position, and ensured to never show signs of change it, even as more data comes in. Besides, every occasion is in the long run relegated such a situation, with probability 1. [43]

In the first place, the accepted round is determined. Occasion x has a acquired round of r if that is the first round in which all the exceptional renowned observers were relatives of it, and the popularity of each observer is chosen for rounds not exactly or equivalent to r . At that point, the acquired time is determined. Assume occasion x is presented an information about I , then Aek makes an extraordinary well known observer y for information I . Its estimation locates z , one of the untimely predecessors that y had knowledge about x . Let t be considered the timeframe which Aek feeds z when z is first made. T should be viewed as the time at that instance where Aek professes of knowing about x . The acquired time for x is the middle of all such timestamps, for every one of the makers of the extraordinary renowned observers in round r . At that point the agreement arrange is determined. All occasions are arranged by their acquired round. On the off chance that two occasions have the equivalent acquired round, they are arranged by their acquired time. In the event that there are still ties, they are broken by basically arranging by signature, after the mark is brightened by XORing with the marks of all the extraordinary renowned observers in the acquired round.

```

procedure divideRounds
  for each event  $x$ 
     $r \leftarrow$  max round of parents of  $x$  (or 1 if none exist)
    if  $x$  can strongly see more than  $2n/3$  round  $r$  witnesses
       $x.\text{round} \leftarrow r+1$ 
    else
       $x.\text{round} \leftarrow r$ 
     $x.\text{witness} \leftarrow$  ( $x$  has no self parent)
      or ( $x.\text{round} > x.\text{selfParent}.\text{round}$ )

```

Figure 3.22. The divide Rounds methodology. [72]

When an occasion x is known, it is relegated a round number x round, and the boolean esteem x witness is determined, showing whether it is a "witness", the first occasion that a part made in that round.


```

procedure decideFame

for each event x in order from earlier rounds to later
  x.famous ← UNDECIDED
  for each event y in order from earlier rounds to later
    if x.witness and y.witness and y.round > x.round
      d ← y.round - x.round
      s ← the set of witness events in round
           y.round-1 that y can strongly see
      v ← majority vote in s (is TRUE for a tie)
      t ← number of events in s with a vote of v
      if d = 1 // first round of the election
        y.vote ← can y see x?
      else
        if d mod c > 0 // this is a normal round
          if t > 2*n/3 // if supermajority, then decide
            x.famous ← v
            y.vote ← v
            break out of the y loop
          else // else, just vote
            y.vote ← v
        else // this is a coin round
          if t > 2*n/3 // if supermajority, then vote
            y.vote ← v
          else // else flip a coin
            y.vote ← middle bit of y.signature

```

Figure 3.23. To choose Fame system. [72]

For each observer occasion (i.e., an occasion x where x witness is valid), choose whether it is celebrated (i.e., allot a boolean to x famous). This choice is finished by a Byzantine acceptance convention dependent on virtual casting a ballot. Every part runs it locally, all alone duplicate of the hashgraph, with no extra correspondence. It treats the occasions in the hashgraph as though they were sending votes to one another; however the computation is simply next to a part's PC. The part allots votes to the observers of each round, for a few rounds, until more than $2/3$ of the populace concurs. To find the popularity of x , re-run this more than once on the developing hashgraph until x famous gets esteem.

```

procedure findOrder

for each event x
  if there is a round r such that there is no event y
    in or before round r that has y.witness=TRUE
    and y.famous=UNDECIDED
  and x is an ancestor of every round r unique famous
    witness
  and this is not true of any round earlier than r
  then
    x.roundReceived ← r
    s ← set of each event z such that z is
        a self-ancestor of a round r unique famous
        witness, and x is an ancestor of z but not
        of the self-parent of z
    x.consensusTimestamp ← median of the
        timestamps of all the events in s

return all events that have roundReceived not UNDECIDED,
  sorted by roundReceived, then ties sorted by
  consensusTimestamp, then by whitened signature

```

Figure 3.24. The find Order system. [72]

When every one of the observers in round r have their notoriety chosen, find the arrangement of acclaimed observers in that round, at that point expel from that set any celebrated observer that has indistinguishable maker from others in that set. The staying well known observers are the extraordinary popular observers.

They go about as the judges to allocate prior occasions around acquiring an agreement timestamp. An occasion is said to be "acquired" in the first round where all the remarkable well known observers have acquired it, if every prior round have the distinction of all observers chose. Its timestamp is the middle of the timestamps of those occasions where every one of those individuals first obtain it. Once these have been determined, the occasions are arranged by round acquired. Any ties are sub arranged by accord timestamp. Any residual ties are sub arranged by brightened signature. The brightened mark is the mark "XORed" with the marks of all remarkable well known observers in the acquired round.

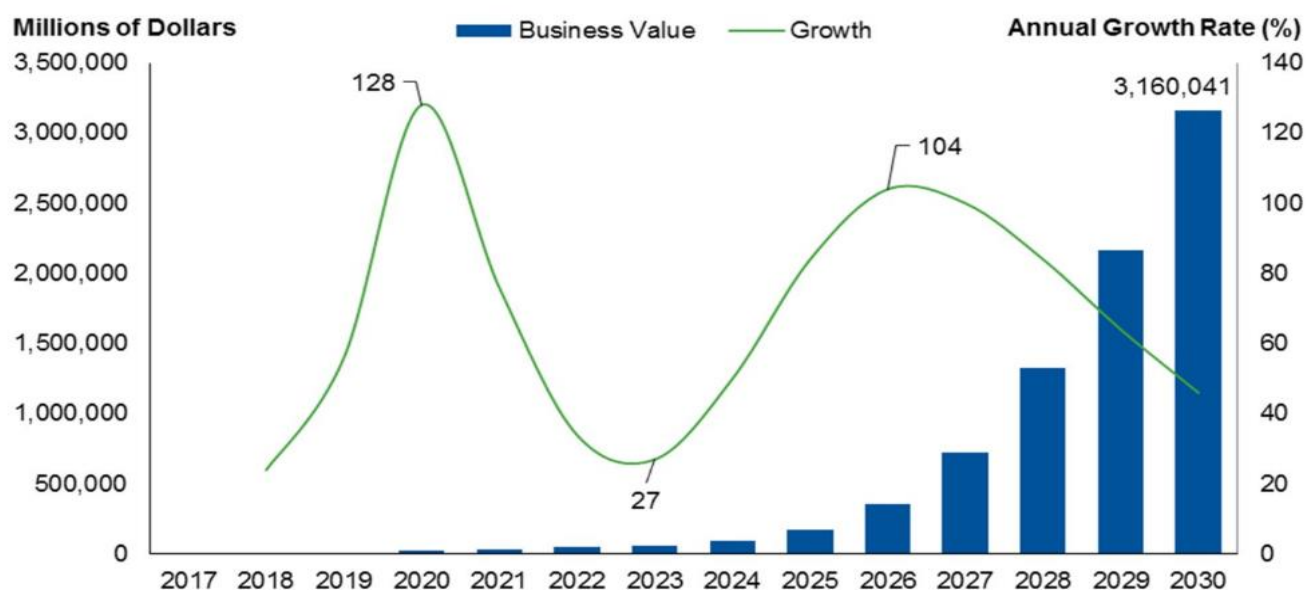


Figure 3.25: Impressive estimated results for a Hashgraph business application. [16]

A persistent advancement in the business is foreseen with the use of Hashgraphy. As per figure 3.25, there is suspected to be a 128% improvement rate (green) in year 2030 that is in game plan to the estimation of the business (blue). Hashgraphy is ascending out to be the fastest, secure, powerful figuring over interchange DLTs and along these lines has an extraordinarily positive and beguilement changing future due its striking features.

With a sheltered, brisk, open record, the possible destiny of Hashgraph could tremendously improve new and existing scattered applications, as the intersection purpose of flowed record advancement and AI meet in new ways.

Each DLT based computation has been cleared up in the most capable and correct way. Regardless of the way that they rely upon equivalent foundation in spite of all that they differ on some huge characteristics of a decentralized system.

The given Table 3.4 (underneath) profitably isolates the three standard DLTs specifically Blockchain, Tangle and Hashgraph.

The characteristics thought about to do this connection are: Data structure, Ledger create Permission, Anonymity, Consensus, Efficiency, and Central Authority and in end copyright.

Table 3.4: Comparison of mainstream DLTs Blockchain, Tangle and Hashgraph [44]

	Blockchain	Tangle	Hashgraph
Data structure	Blockchain	DAG	DAG
Ledger type	Public	Public	Private
Permissioned	No	No	Yes
Anonymous	Yes	Yes	No
Consensus	PoW, PoS	PoW	GaG, VV
Efficiency	Low	High	High
Central authority	No	Yes	No
Copyright	Open source	Open source	Proprietary

Beside twofold growing, which will dependably yet reliably be attainable for computerized money applications, assaults will limit to an extent of character ambushes (i.e., client side security), arrange attacks, (for instance, DDoS, sybil) and mining attacks, (for instance, >50%, square disposing of, and Brute power).

3.7 Complete outline of the potential security dangers alongside their effects on different elements in a Distributed Ledger Technology (DLT)

The details in Table 3.5 (underneath) gives a total framework of the potential security threats close by their consequences for various components in a Distributed Ledger Technology (DLT) and their possible game plans that exist recorded as a hard copy as of recently.

Table 3.5: Potential security threats on mainstream DLTs and counter measures [45]

Attack	Description	Primary target	Adverse effects	Possible countermeasure s	Affects Blockchain	Affects Hashgraph	Affects Tangle
DDoS	Community assault to deplete arrange resources	Distributed Ledger Technology(DLT) organize, organizations, mineworkers, and users	deny administrations to genuine clients/ex cavators, confine or head out the miners	Proof-of-Activity (PoA) convention, quick check signature based authentication	No	No	No
Sybil	adversary makes numerous virtual identities	DLT arrange, excavators, users	facilitates time jacking, DDoS, and twofold	Xim (a two-party blending convention)	No	No	No

			spending assaults, debilitate s client privacy				
Time lifting	adversary speed the dominant part of excavator's clock	Miners	isolate a minework er and waste its assets, impact the mining trouble figuring process	limitation resistance ranges, organize time convention (NTP) or time examining on the qualities obtained from confided in peers	Yes	No	Yes
Twofold spending or Race assault	spent the same bitcoins in numerous exchanges, send two clashing exchanges in fast succession	sellers or merchants	sellers lose their items, head out the legitimate clients, make blockchai n forks	inserting spectators in arrange, conveying twofold spending alarms among peers , close-by associates ought to advise the shipper around a progressing twofold spend at the earliest opportunity, vendors should handicap the immediate approaching associations	Yes	No	No

3.8 Chapter Summary

This chapter described the details of the Distributed Ledger Technology and its advantages and disadvantages. The most popular DLTs are Blockchain, Hashgraph and Tangle and this chapter gives a detailed analysis of the working mechanisms, consensus logic, structures, and efficiency for these algorithms. Proof-of-work and proof-of-stake concepts too have been discussed in further detail.

The next chapter covers the methodologies and techniques employed in this study.

CHAPTER 4

METHODOLOGIES

This chapter covers different methodology and types of techniques used in this study. Section 4.1 describes Hypothesis used for this research. Section 4.2 explains the two types of methodologies, which focuses on the Quantitative approach & Agile methodology which is the used in this study. Section 4.3 covers two types of data collection processes, the literature review process, and the experimental data gathering process. Also there is a discussion in regard to the process of generating legitimate traffic, attack traffic & evaluating traffic has been given.

4.1 Research Hypothesis

Increment in number of exchanges every second will enhance the profitability/productivity of the framework. According to research, the quantity of exchanges in the extremely prominent Bitcoin application dependent on Blockchain is constrained to 6-7 exchanges for each second [46]. Be that as it may, there is no such constraint on Hashgraphy as it is completely subject to the transfer speed.

On Hashgraphy,

$$\text{Number of transactions per second} = \frac{\text{Total Bandwidth (Megabits per second)}}{\text{Number of bits in a packet}}$$

For instance, for 1000 megabits for every second data transfer capacity with bundle estimates around 40 bits, the quantity of exchanges will be roughly 25. According to explorations on multi-modular fiber lines [47] with most extreme data transfer capacity of 32 terabytes for every second and parcel measure 40 bits, the quantity of exchanges will be roughly 6,400,000.

Thusly, on the off chance that every part has enough transfer speed to download 4,000 transactions for each second, that is the capacity of the framework can deal with. That would probably require just a couple of megabits for each second, which is a common home broadband connectivity. What's more, it would be quick enough to deal with the majority of the exchanges of the whole Visa card systems, around the world. The Bitcoin furthest reaches of 7 exchanges for every second can obviously be enhanced in different ways.

4.2 Method Used for Study

4.2.1 Quantitative Research

This proposition work uses the quantitative research approach in light of composing examination. As appeared by Matthews and Ross, quantitative research procedures [34] are essentially connected with the social event of information that is made and which could be tended to numerically. Everything considered, quantitative information is gathered when specialist has acquired a hands-on on the confirmed epistemological procedure and information is assembled that can be probably be secluded by calculations.

The examination strategy is picked as a result of the freshness of the point thought about. By directing the proposition work, the idea of Hashgraphy can be connected to a constant following framework. It is also possible to significantly take cue of the advancement and analyse use cases and recommendations caused by it.

I have set up a simulated environment utilizing hashgraphy to quantify speed, productivity, security of the framework. A group of members (based on the concept of hashgraphy) have been thought to be vehicles. The hashgraphy calculation has been utilized in following different vehicles and checking their exact area utilizing Global Positioning System (GPS) and report some continuous data like climate, traffic, conveyance subtleties, etc.

A member communicate data arbitrarily to different members. Any member, after getting data will make an event. This event contains a timestamp, data, and two hashes - one self-made and one from the other hub. Events in this manner develop a Hashgraph, and this is added to history. In this manner, each member can have access to every communication that has been shared. With the well-established concept of virtual voting, a group of members can foresee on how another member will cast a vote. There is no chance to get of any member lying in light of the fact that everything is put away in history, in this manner creating a proof-of-stake with minimal effort. After some time, members ceaselessly refresh the chart with constant data they will get.

In the same way as other different applications currently following this framework depend on consensus mechanisms. The use of crypto-economy implies decentralization of this war room and appropriation between all the members .The strategy was considered as the most reasonable model for this examination.

4.2.2 Agile Methodology

This thesis also uses Agile Methodology based on Scrum model for monitoring and tracking the progress of the tracking application prototype. The various features of the tracking application have been estimated across time required to develop them. These features are categorized into phases known as Milestones. Milestone 1 incorporated all the wireframing and design of what features should be included and how the application's screens should look like. Milestone 2 focussed on the Backend Development. And this phase included data collection and database creation utilizing the entity- relationship concepts. Milestone 3 focussed on integrating the backend of milestone 2 with the front end screens thereby creating our tracking application working prototype. Milestone 4 incorporated the Hashgraphy concept to our working prototype. Milestone 5 focussed on the overall testing of the application to ensure smooth work flow and functioning of the application as desired.

Additional features can be added or any features can be removed during any of the Milestone phases, and this is the most flexible feature of using Agile. But any change falls in a cycle of build->test->raise defects->debug/fix->build, and the cycle repeats. Therefore, Agile is observed to be a iterative yet incremental model.

Agile SDLC methodology is a blend of iterative and consistent process models with focus on process adaptability and client/customer's satisfaction aided by quick delivery of working software programming products. These Agile based methods break the working software products into minimal builds. These builds are then delivered to the client in cycles thereby making it an iterative and incremental methodology.

Figure 4.1 portrays the agile fundamental lifecycle delineated by the Disciplined Agile Delivery (DAD) framework. The lifecycle is DAD's Scrum-based, or "key", light-footed movement lifecycle yet it in like manner reinforces a lean/Kanban kind of lifecycle and a steady moving lifecycle as well. [48]

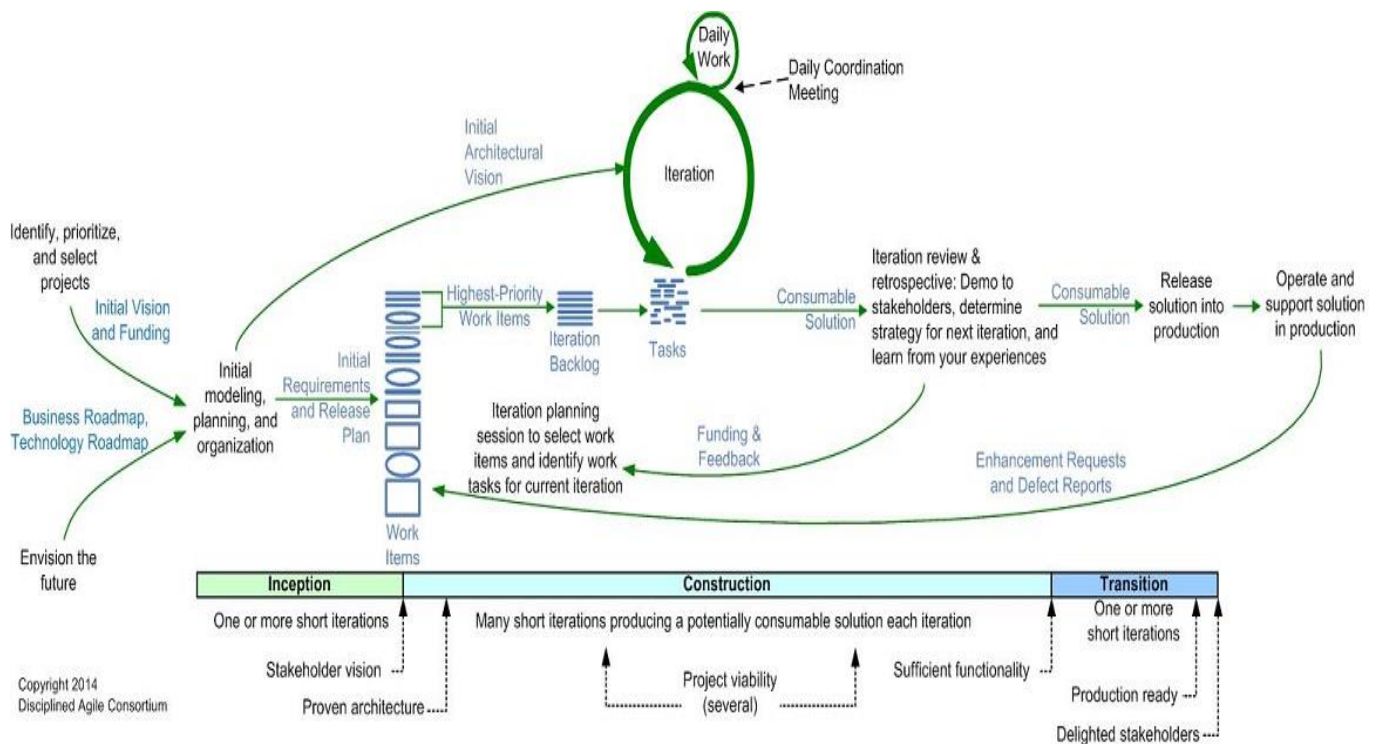


Figure 4.1. The DAD agile life cycle.

1. Garnering beginning help and financing for the undertaking. This stage centres around what we will be getting, what amount will cost, and how much time it is going to take. We should have the capacity to give sensible, albeit possibly developing, responses to these inquiries in case we will motivate authorization to take a shot at the task. We may need to legitimize ventures by means of a practicality reasoning.

2. Passionately working along partners to first structure the extent of framework. Agilists will do some underlying prerequisites displaying with their work partners to recognize the underlying, but abnormal state, necessities for the framework. To advance dynamic partner investment comprehensive instruments ought to be utilized, for example, file cards and white sheets to do this displaying. The subtleties of these necessities are demonstrated on the nick of time premise in structured raging showdowns amid advancement period.

3. Beginning to assemble the group. In spite of the fact that group will advance after some time, toward the start of an improvement venture there will be a need to begin recognizing key colleagues and begin bringing them onto the group. Now there will be designers, the task mentor/supervisor, and at least one partner agents.

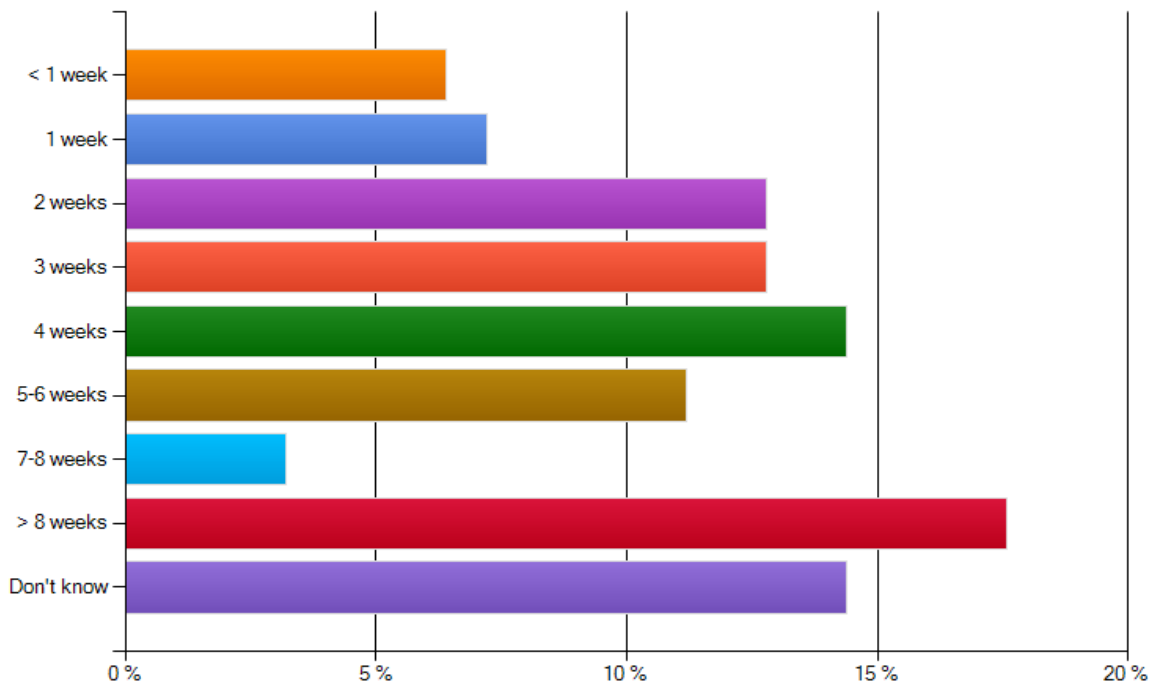
4. Modelling an underlying engineering for the framework. From the get-go in the venture it is a great idea to have a general thought of how the framework will be fabricated. Is it a Java application? A Go Programming based function? J2EE? Anything different? The objective is to recognize a building methodology. We need to work with planned subtleties later amid improvement periods in structural raging sessions and by means of Test Driven Development (TDD).

5. Setting up the environment. There will be requirement for structure and advancement apparatuses to construct the task model.

6. Estimating the task. An underlying appraisal for your nimble venture will be put dependent on the underlying prerequisites, the underlying engineering, and the aptitudes. This gauge will advance all through the venture. [48]

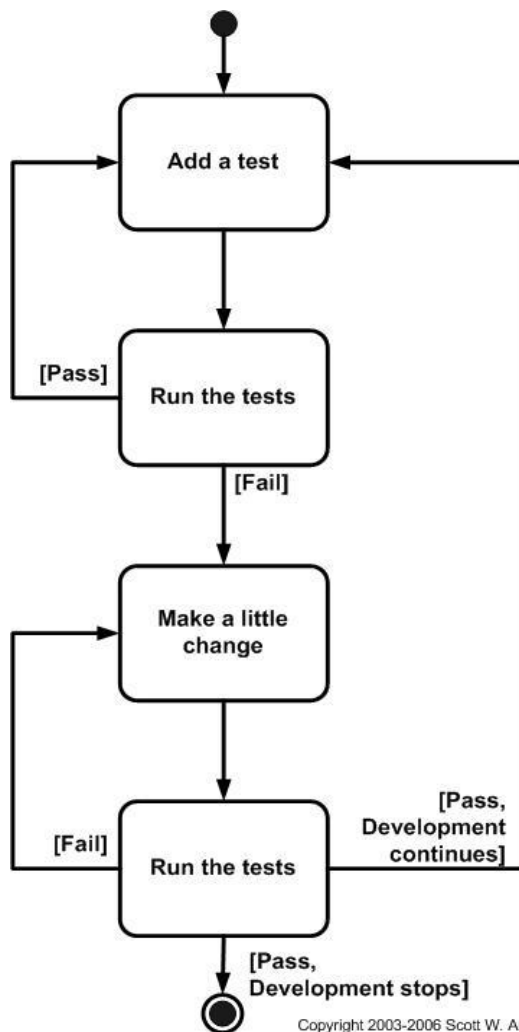
The 2013 Agile Project Initiation Survey found that the normal time to start a deft venture took 4.6 weeks. Figure 4.2 aims at the scope of inception periods with % of work completion versus Time (in weeks). Contrasts are the after effects of the unpredictability of the area/issue space, specialized multi stakeholder nature of what you're aiming to achieve, accessibility of partners, capacity of partners to come to understanding with regards to the extension. also, capacity of the group to shape itself and to get

fundamental assets. [49]



% of work completion versus Time (in weeks)

Figure 4.2. Agile Testing -flow chart



Copyright 2003-2006 Scott W. Ambler

Figure 4.3. Taking a "test first" approach to Agile project development[49]

1. Final testing of the framework. Last framework and acknowledgment testing ought to be performed now, in spite of the fact that the larger part of testing ought to be finished amid advancement cycles (rerun your relapse test suite). On the other hand, pilot/beta tests your framework with subdivision of the possible end clients.

2. Modify. There is no esteem testing the framework on the off chance that you don't organize to follow up on deformities that you find. You may not address all deformities, but rather you ought to hope to fix some of them.

3. Fulfilment of any framework & client attestation. Some attestation may have been composed amid development cycles, however it normally isn't finished until the point when the framework discharge itself has been concluded to keep away from superfluous revamp. Documentation is dealt with like some other prerequisite: it ought to be cost, organized, and made just if partners are happy to put resources into it. Agilists trust that on the off chance that partners are savvy enough to win the cash, they should likewise be sufficiently shrewd to put in suitably.

4. Guidance. We guide end clients, tasks staff, & care staff to work successfully with own framework.

5. Deploy the framework. We discharge the arrangement into generation

4.3 Data Collection Process

This segment portrays the information gathering step, which is utilized to acquire information for this thesis. There were two kinds of information accumulation techniques in this thesis: the literature review and an experimental information gathering process. The latter gives both the learning and data required for the examination. Findings for the literature review was obtained from journals, papers, books and other trusted sources from the Internet. The information gathering process for Hashgraphy and constant data building for application's execution was done utilizing my own database in MySQL. The entities and relationships graph of this database has been clarified in later sections.

4.3.1 Literature Review Process

The literature review is the initial process that enhances the researcher intellectually while reading and analysing various relevant works. Also it provides us with knowledge base, and information needed for this study. In this research, all literature was gathered from different credible resources such as academic databases, academic peer- reviewed journals, library references, and appropriate credible association websites. Table 4.1 is an example of the solid assets, where data for this exploration was recovered.

Table 4.1: Credible Resources

Resources	From
Academic Database	IEEEExplore
Books	Unitec library, and Google Book
Web Search Engine	Google scholar, Swirls website, Medium website, Hackathon, Blockgeeks

The resources mentioned in Table 4.1 prompted look into papers significant to the examination, as they are notable and believable for data innovation based research. After the writing is explored and fundamentally investigated, it prompts the following procedure which is data gathering for the application setup. The following sections describe the experimental data gathering process, which explains how the data were collected and analysed.

4.3.2 Data Gathering Process for Implementation of Design and Real-time tracking application

The fundamental asset for information gathering for this examination was making my very own database in MySQL for the improvement. A working model of Hashgraph arrangement was set up in my own PC utilizing the accompanying programming techniques:

1. Programming Language: Java Programming, Java SE 8: Java SE Development Kit 8 (or Java EE 8), Swirlds SDK
2. Editor/Integrated Development Environment: Eclipse Oxygen 4.7
3. Security: Java 8 security

A working model of Blockchain organization was set up in my own PC utilizing the accompanying programming techniques:

1. Programming Language: Go Programming
2. Command Prompt to associate with neighbourhood have on port 8080 and TCP server of Go (Command utilized: telnet <server ip address><port>)
3. Editor: Sublime Text

Tools and software utilized for attack-assaults:

1. Windows 8 Command Prompt
2. Nemesis

Various tests were kept running so as to screen Hashgraph speed and proficiency. To exhibit the strength of DDoS and sybil attacks on the framework a ping of death attacks on the IP locations of the target individuals and a DDoS attack on the system itself has been executed. Which were all gathered by utilizing tools and modern techniques. Table 4.2 demonstrates the Hashgraph assessment measurements and the devices utilized in the information gathering process. Moreover, the way toward gathering this information will be clarified in the following area.

Table 4.2: Hashgraph Evaluation Metrics and Tools Used For Collecting Data

Hashgraph Evaluation Metrics	Calculations & Tools For Collecting Data
Speed	Number of transactions per second depending on system bandwidth
Efficiency	Increase in bandwidth improves the efficiency of the system.
Security	Launch ping of death, DDoS and Sybil attacks to check Hashgraph resilience
Tracking	GPS tracking system

4.3.3 Process of Generating Legitimate Traffic

The general correspondence between the individuals/hubs of the framework and the transmission of parcels of information over the hubs is all genuine traffic. By producing these assault traffic beneath we are attempting to control the real traffic stream and watch on the off chance that it is altering the framework in any way. The way toward producing genuine traffic of the framework can be seen in the Hashgraph usage area of this thesis.

4.3.4 Process of Generating Attack Traffic

The ping of death attack and DDoS attack produce assault traffic on the IP address of the person in question and system individually. Definite portrayal of the effect of these assaults and their usage is on the Attacks on Hashgraph segment of this thesis.

% of companies experiencing some form of DDoS attack (last 12 months)

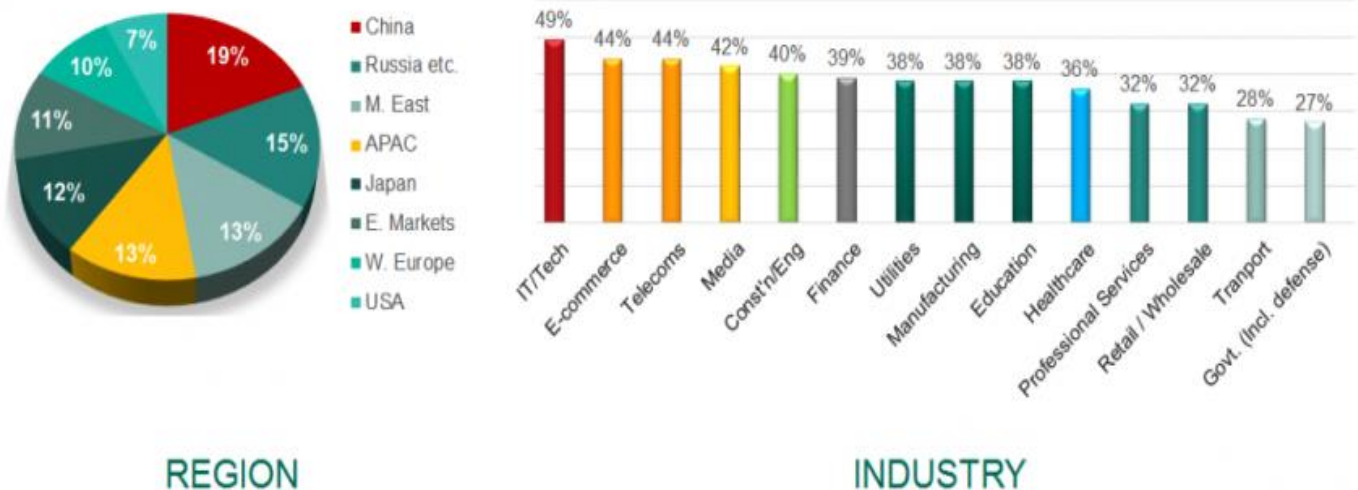


Figure 4.4. statistics of percentage of companies experiencing DDoS attack [50]

The above chart Figure 4.4 demonstrates the measurements of the % of organizations encountering some type of DDoS assault over the most recent one year. The % of DDoS assaults is higher in IT/Tech organizations principally in the China locale, trailed by online business which is another mainstream industry. Since hashgraph claims protection from assaults, ideally it is the distinct advantage innovation with a battle back reaction to these assaults.[50]

4.3.5 Process of Evaluating Defences

The hashgraph white paper professes to be safe of DDoS and Sybil assaults. Along these lines, assaults are being actualized on the framework to watch its versatility and its effect on the general consensus mechanism. The definite system of assaults usage has been portrayed in later parts.

4.4 Chapter Summary

This chapter covers research hypotheses, methodology of study, and data collection process. The pre-defined hypotheses showed the scope or the boundary of this analysis, while a quantitative analysis method was chosen as the main methodology and used to manage the research from beginning to end; agile methodology was used for implementation design and real-time tracking application. A database was created and working hashgraph prototype has been setup to gather quantitative data. This chapter also detailed the most common attacks such as DDoS, Ping of death and Sybil evaluation methods, which are implemented on the system and in theory. In addition, the Hashgraphy prototype itself was selected as an evaluation method since it provided a more realistic evaluation environment compared to the simulation and theory methods. The end of this chapter presented the procedures of data collection, and the steps involved in the process of both the literature review, and the experiment for this research, with examples of graphs also presented.

The next chapter covers the implementation of design.

CHAPTER 5

Implementation of Design

This chapter gives & describes the implementation of Design used in this research to obtain the results. Section 5.1 covers the implementation set-up of blockchain along with its advantages, disadvantages, tools and software used. It also explains core concepts of proof-of-stake in relation to blockchain, its architecture & the process of creating prototype. Section 5.2 covers the implementation of hashgraphy, tools & software used and the process of creating prototype. it also give the consensus information & derivation from its implementation.

5.1 Implementation Set-up of Blockchain.

Making a Blockchain Proof-of Stake in a simulation domain.

In Proof of Stake, blocks are either "stepped" or "created" in perspective of proportion of tokens each centre is anxious to set up as protection. These centre points are known as validators. More the number of tokens then each validator is glad to assemble as protection, more noticeable is its possibility that needs to have formed the accompanying block & get rewarded. You can consider this store interest. The additional money you cash up in your speculation account at bank, more noticeable is the month to month premium portion you get. Correspondingly, your probability of creating the accompanying block grows the more tokens which is set up as assurance. You are Claiming your share, which is why this type of process is known as Proof of Stake [POS]. [51]

Advantages:-

1. Bitcoin which is based over Blockchain's POS concept has gained so much popularity that customary individuals have not possessed the capacity to mine without using anyone else's PCs in years. Hence, numerous individuals contend that POS in reality have more flexibility since anybody at any rate can take a PC without arranging a mining rig. Costly equipment's are not required, only few enough tokens to claim are sufficient. This cost effective feature of POS is an advantage.

2. POS has been utilized for a long time in Nxt [52], and has not been broken regardless of having the third most noteworthy market capitalisation, so it seems to be secure. On the other side, had it not been secure, somebody would have broken it at this point. History indicates that forks happen once in a while and multiply in exponents, so it achieves agreement. In principle, a shortcoming is that individuals can vote in favour of the two sides of a fork. By and by, that doesn't appear to occur. The gain from doing as such would be minor (there's no square reward in Nxt), and the loss of security abundant, so nobody does it. With Proof of Stake, the general population who secure the block chain the most are likewise the general population who have the most coins, so they have the most motivator to protect the respectability of the cash. To put it another way, in Nxt manufacturing is done to anchor the block chain, not to make a benefit.

Disadvantages:-

One disadvantage is that in unadulterated POS, the best way to secure coins is from somebody who as of now has them. This can prompt issues with the conveyance. For instance, in Nxt the whole coin supply was at first appropriated to 73 "authors", and a portion of those individuals still possess critical parts of the supply, giving them riches and impact. All things considered, none currently possess as vast a part as Satoshi claims of Bitcoin. It appears that the conveyance issue illuminates itself after some time, as the authors have an enthusiasm for spending their prosperity to help the coin. [53]

5.1.1 Tools & Software used

Programming Language: Go Programming

Command Prompt to connect to local host on port 8080 and TCP server of Go (Command Used telnet <server IP address><port>)

Editor: Sublime Text

5.1.2 Core concepts of this proof-of-stake blockchain

This blockchain will actualize the centre ideas of Proof of Stake.

- Full shared usage. Systems administration is mimicked and the focal blockchain state is held by a solitary Go TCP server. In this instructional exercise, the state is communicated to every hub from the single server.
- Wallet and parity following. I have executed a token usefulness as a wallet in this code. Hubs are spun up in the system and the token sum is inputted in 'stdin'. So you can type in any sum you need. A full execution would connect every hub with a hash address and monitor token adjusts in each.

5.1.3 Architecture

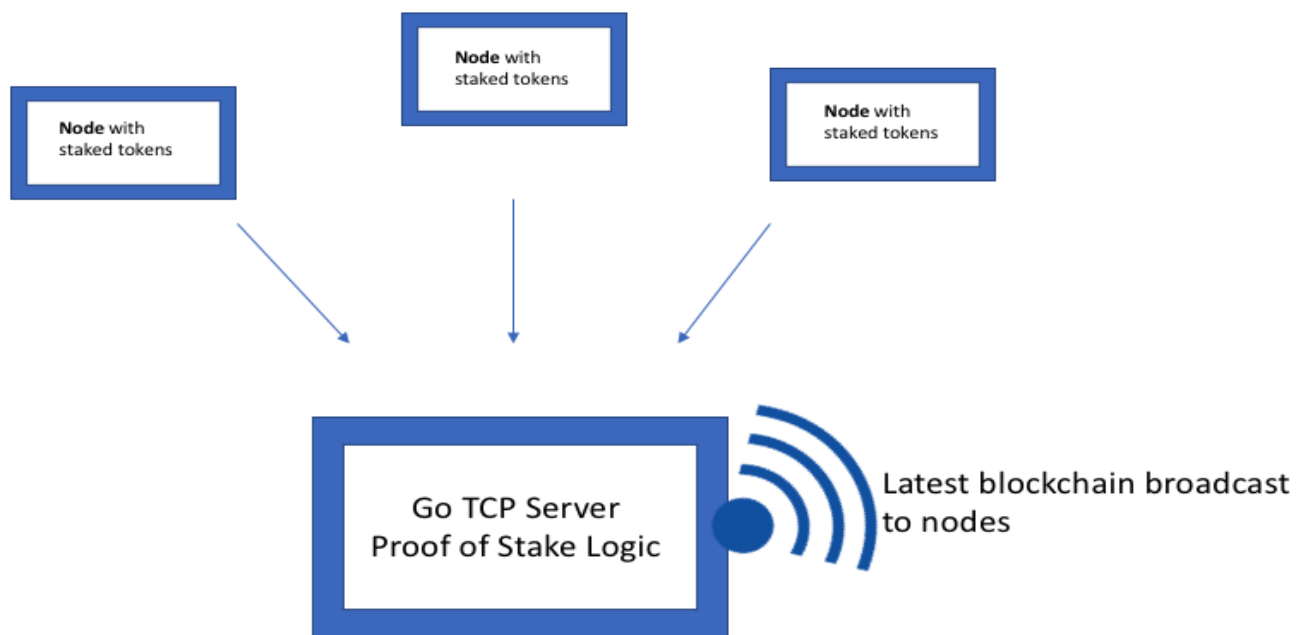


Figure 5.1. Design of Proof-of-Stake of blockchain [55]

- I have setup a TCP server which is Go-based through which different hubs (validators) can interface.
- Most recent blockchain gets communicated to every hub intermittently.
- Every single hub will suggest latest blocks.
- Established on quantity of shares claimed every hub, 1 from all hubs will then be haphazardly picked as champion, and its blocks will be attached and added to the blockchain. [55]

5.1.4 Process of creating prototype

Step 1. Setup & Imports:

- Environment variable needs to be set up, & only TCP server knows which one to use. A .env file is created in the operational database of the code with single line in it: ADDR=8080. The Go program reads through the document & will now expose port 8080 so that all nodes gets connected.
- Next a main.go file is created for the operational database and start POS coding.
- spew is an advantageous bundle that prints our blockchain to the terminal.
- godotenv enables us to peruse from our .env record we made beforehand.

```

package main

import (
    "bufio"
    "crypto/sha256"
    "encoding/hex"
    "encoding/json"
    "fmt"
    "io"
    "log"
    "math/rand"
    "net"
    "os"
    "strconv"
    "sync"
    "time"

    "github.com/davecgh/go-spew/spew"
    "github.com/joho/godotenv"
)

```

Figure 5.2: Step 1 Setup & Import

Step 2. All global variables are declared.

Next, all global variables are declared.

- Block is the content of each block
- Blockchain is our official blockchain, that is simply a series of validated blocks.
- The PrevHash in each block is compared to the Hash of the previous block to make sure our chain is robust.
- tempBlocks is simply a holding tank of blocks before one of them is picked as the winner to be added to Blockchain
- candidateBlocks is a channel of blocks; each node that proposes a new block sends it to this channel
- announcements is a channel where our main Go TCP server broadcasts the latest blockchain to all the nodes
- mutex is a standard variable that allows us to control reads/writes and prevent data races
- validators is a map of nodes and the amount of tokens they've staked

```

// Block represents each 'item' in the blockchain
type Block struct {
    Index      int
    Timestamp  string
    BPM        int
    Hash       string
    PrevHash   string
    Validator  string
}

// Blockchain is a series of validated Blocks
var Blockchain []Block
var tempBlocks []Block

// candidateBlocks handles incoming blocks for validation
var candidateBlocks = make(chan Block)

// announcements broadcasts winning validator to all nodes
var announcements = make(chan string)

var mutex = &sync.Mutex{}

// validators keeps track of open validators and balances
var validators = make(map[string]int)

```

Figure 5.3: Step 2 Global variables are declared.

Step 3. Basic Blockchain functions are coded.

Next, the Basic Blockchain functions are coded.

- `calculateHash` takes in a string and returns its SHA256 hash representation.
- `calculateBlockHash` hashes the contents of a block by concatenating all its fields.
- `generateBlock` is how a new block is created. The important fields we include in each new block are its hash signature (calculated by `calculateBlockHash` previously) and the hash of the previous block `PrevHash` (so we can keep the integrity of the chain). We also add a `Validator` field so we know the winning node that forged the block.

We hash data for 2 main reasons:

- To spare space. Hashes are obtained from every one of the information that is on the block. For our situation, we just have a couple of information focuses however envision we have information from hundreds, thousands or a great many past blocks. It's considerably more productive to hash that information into a solitary SHA256 string or hash the hashes than to duplicate every one of the information in going before squares again and again.
- Preserve uprightness of the blockchain. By putting away past hashes as we do in the chart beneath, we're ready to guarantee the blocks in the blockchain are organized appropriately. In the event that a malevolent gathering were to come in and attempt to control the information (for instance, to change our pulse to fix life coverage costs), the hashes would change rapidly and the chain would "break", and everybody would know to not believe that noxious chain.

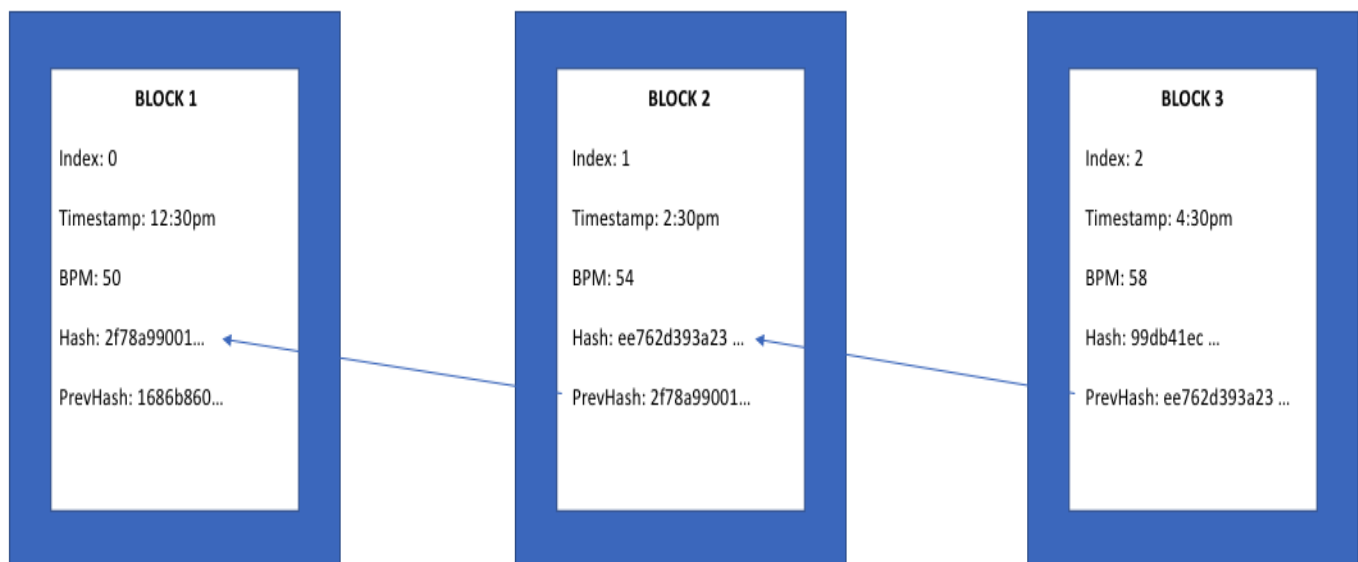


Figure 5.4 Different blocks for above function [55]


```
// SHA256 hashing
// calculateHash is a simple SHA256 hashing function
func calculateHash(s string) string {
    h := sha256.New()
    h.Write([]byte(s))
    hashed := h.Sum(nil)
    return hex.EncodeToString(hashed)
}

//calculateBlockHash returns the hash of all block information
func calculateBlockHash(block Block) string {
    record := string(block.Index) + block.Timestamp + string(block.BPM) + block.PrevHash
    return calculateHash(record)
}

// generateBlock creates a new block using previous block's hash
func generateBlock(oldBlock Block, BPM int, address string) (Block, error) {

    var newBlock Block

    t := time.Now()

    newBlock.Index = oldBlock.Index + 1
    newBlock.Timestamp = t.String()
    newBlock.BPM = BPM
    newBlock.PrevHash = oldBlock.Hash
    newBlock.Hash = calculateBlockHash(newBlock)
    newBlock.Validator = address

    return newBlock, nil
}
```

Figure 5.5: Step 3 Basic Blockchain functions are coded

- isBlockValid performs the Hash and PrevHash check to make sure our chain has not been corrupted.

```
// isBlockValid makes sure block is valid by checking index
// and comparing the hash of the previous block
func isBlockValid(newBlock, oldBlock Block) bool {
    if oldBlock.Index+1 != newBlock.Index {
        return false
    }

    if oldBlock.Hash != newBlock.PrevHash {
        return false
    }

    if calculateBlockHash(newBlock) != newBlock.Hash {
        return false
    }

    return true
}
```

Figure 5.6 performing hash & prevhash check.

Step 4. Validator

When a validator connects to the TCP server, we need to provide it some functions that achieve a few things:

- Allow it to enter a token balance you want the validator to stake
- Receive a broadcast of the latest blockchain
- Receive a broadcast of which validator in the network won the latest block
- Add itself to the overall list of validators
- Enter block data BPM—remember, this is each validator's pulse rate
- Propose a new block


```
Telnet 192.168.178.23
Enter token balance:2
Enter a new BPM:75
Enter a new BPM:
    winning validator: 649d978d433e7cc53383ce5e2b0be723471233edec3c2
f8320c32649ba20bae9
[{"Index":0,"Timestamp":"2019-01-19 12:19:04.4301275 +1300 NZ
DT m=+0.005005001","BPM":0,"Hash":"96a296d224f285c67bee93c30f8a309157f0daa35dc5b
87e410b78630a09cfc7","PrevHash":"","Validator":""}, {"Index":1,"Timestamp":"2019-
01-19 12:19:47.1400088 +1300 NZDT m=+42.714886301","BPM":60,"Hash":"300097a3e735
b8b15f7b5b1d9fb10212474c54df2cb24b6185f8db2a4944b779","PrevHash":"96a296d224f285
c67bee93c30f8a309157f0daa35dc5b87e410b78630a09cfc7","Validator":"0c8b59d023d1309
721b6eee046e049d43cd27cbece6721b2a70ffe79655063e7"}, {"Index":2,"Timestamp":"2019
-01-19 12:21:42.4974845 +1300 NZDT m=+158.072362101","BPM":75,"Hash":"9f8a923235
4b84ac4e10d10d4689ae79d19ac0f3a13dc387f13de1d41c085ab0","PrevHash":"300097a3e735
b8b15f7b5b1d9fb10212474c54df2cb24b6185f8db2a4944b779","Validator":"649d978d433e7
cc53383ce5e2b0be723471233edec3c2f8320c32649ba20bae9"}]
```

Figure 5.7 Describing step 4 Validator

- We then enter BPM which is the validator's rate and create a separate Go routine to process our block logic.

Step 5. Picking a Winner

This is the main feature of Proof of Stake logic that illustrates how a winning validator is chosen; the higher the number of tokens they stake, the higher their probability should be to be chosen as the winner who gets to forge their block.

- In my code, we will only make validators who propose new blocks eligible to be chosen as the winner. In traditional Proof of Stake, a validator can be chosen as the winner even if they don't propose a new block. Remember, Proof of Stake isn't a definition, it's a concept; there are lots of different implementations of Proof of Stake
- We pick a winner every 30 seconds to give time for each validator to propose a new block. Then we need to create a lotteryPool that holds addresses of validators who could be chosen as our winner. Then we check to see there actually are some blocks proposed in our temporary holding tank of proposed blocks with if `len(temp) > 0` before proceeding with our logic.
- We check to make sure we haven't already come across the same validator in our temp slice. If we do, skip over the block and look for the next unique validator.
- We make sure the validator we get from our block data in temp is actually an eligible validator that sits in our validators map. If they exist, then we add them to

our lottery Pool.

- We fill our lotteryPool with copies of the validator's address. They get a copy for each token they've staked. So a validator who put in 100 tokens will get 100 entries in the lotteryPool. A validator who only put in 1 token will only get 1 entry.
- We randomly pick the winner from our lotteryPool and assign their address to lotteryWinner.
- We then add their block to our blockchain and announce the winner to the rest of the nodes who won the lottery
- We clear out our tempBlocks holding tank so it can be filled again with the next set of proposed blocks.

```

C:\ Telnet 192.168.178.23
Enter token balance:5
Enter a new BPM:60
Enter a new BPM:
    winning validator: 0c8b59d023d1309721b6eee046e049d43cd27cbece672
1b2a70ffe79655063e7
[{"Index":0,"Timestamp":"2019-01-19 12:19:04.4301275 +1300 NZDT m=+0.005005001","BPM":0,"Hash":"96a296d224f285c67bee93c30f8a309157f0daa35dc5b87e410b78630a09cfc7","PrevHash":"","Validator":""}, {"Index":1,"Timestamp":"2019-01-19 12:19:47.1400088 +1300 NZDT m=+42.714886301","BPM":60,"Hash":"300097a3e735b8b15f7b5b1d9fb10212474c54df2cb24b6185f8db2a4944b779","PrevHash":"96a296d224f285c67bee93c30f8a309157f0daa35dc5b87e410b78630a09cfc7","Validator":"0c8b59d023d1309721b6eee046e049d43cd27cbece6721b2a70ffe79655063e7"}]
[{"Index":0,"Timestamp":"2019-01-19 12:19:04.4301275 +1300 NZDT m=+0.005005001","BPM":0,"Hash":"96a296d224f285c67bee93c30f8a309157f0daa35dc5b87e410b78630a09cfc7","PrevHash":"","Validator":""}, {"Index":1,"Timestamp":"2019-01-19 12:19:47.1400088 +1300 NZDT m=+42.714886301","BPM":60,"Hash":"300097a3e735b8b15f7b5b1d9fb10212474c54df2cb24b6185f8db2a4944b779","PrevHash":"96a296d224f285c67bee93c30f8a309157f0daa35dc5b87e410b78630a09cfc7","Validator":"0c8b59d023d1309721b6eee046e049d43cd27cbece6721b2a70ffe79655063e7"}]
    winning validator: 649d978d433e7cc53383ce5e2b0be723471233edec3c2f8320c32649
ba20bae9
[{"Index":0,"Timestamp":"2019-01-19 12:19:04.4301275 +1300 NZDT m=+0.005005001","BPM":0,"Hash":"96a296d224f285c67bee93c30f8a309157f0daa35dc5b87e410b78630a09cfc7","PrevHash":"","Validator":""}, {"Index":1,"Timestamp":"2019-01-19 12:19:47.1400088 +1300 NZDT m=+42.714886301","BPM":60,"Hash":"300097a3e735b8b15f7b5b1d9fb10212474c54df2cb24b6185f8db2a4944b779","PrevHash":"96a296d224f285c67bee93c30f8a309157f0daa35dc5b87e410b78630a09cfc7","Validator":"0c8b59d023d1309721b6eee046e049d43cd27cbece6721b2a70ffe79655063e7"}, {"Index":2,"Timestamp":"2019-01-19 12:21:42.4974845 +1300 NZDT m=+158.072362101","BPM":75,"Hash":"9f8a9232354b84ac4e10d10d4689ae79d19ac0f3a13dc387f13de1d41c085ab0","PrevHash":"300097a3e735b8b15f7b5b1d9fb10212474c54df2cb24b6185f8db2a4944b779","Validator":"649d978d433e7cc53383ce5e2b0be723471233edec3c2f8320c32649ba20bae9"}]
[{"Index":0,"Timestamp":"2019-01-19 12:21:42.4974845 +1300 NZDT m=+158.072362101","BPM":75,"Hash":"9f8a9232354b84ac4e10d10d4689ae79d19ac0f3a13dc387f13de1d41c085ab0","PrevHash":"300097a3e735b8b15f7b5b1d9fb10212474c54df2cb24b6185f8db2a4944b779","Validator":"649d978d433e7cc53383ce5e2b0be723471233edec3c2f8320c32649ba20bae9"}]

```

Figure 5.8 shows how the winner is picked

```

C:\ Command Prompt - go run src\main\BlockPOS.go
C:\GoCode\testProject>go run src\main\BlockPOS.go
(main.Block) {
  Index: (int) 0,
  Timestamp: (string) (len=53) "2019-01-19 12:19:04.4301275 +1300 NZDT m=+0.005005001",
  BPM: (int) 0,
  Hash: (string) (len=64) "96a296d224f285c67bee93c30f8a309157f0daa35dc5b87e410b78630a09cfc7",
  PrevHash: (string) "",
  Validator: (string) ""
}
2019/01/19 12:19:04 HTTP Server Listening on port : 8080
map[0c8b59d023d1309721b6eee046e049d43cd27cbece6721b2a70ffe79655063e7:5]
map[0c8b59d023d1309721b6eee046e049d43cd27cbece6721b2a70ffe79655063e7:5 649d978d433e7cc53383ce5e2b0be723471233edec3c2f8320c32649ba20bae9:2]

```

Figure 5.9 Efficiency of Blockchain

- From the above code screen shots we can see the efficiency of the Blockchain system as only 3 transactions per second(tps) fixed for a bandwidth of 100 Mbps which is the regular broadband bandwidth. In the next part of this chapter we will observe the efficiency of Hashgraph system in tps, and derive the better algorithm in terms of efficiency and speed on this basis.

Step 6. Proposing a polluted block

The following line is important:

- Delete(validators, address)

On the off chance that the validator endeavours to propose a corrupt block, for our situation, a BPM that isn't a number, that tosses a blunder and we quickly erase the validator from our rundown of validators. They are never again qualified to manufacture new blocks and they lose their parity.

```
C:\>
Enter token balance:9
Enter a new BPM:2.3

Connection to host lost.

C:\Users\Sushmitha>
```

Figure 5.10 Proposing a polluted block

- This potential to lose your token equalization is a noteworthy motivation behind why Proof of Stake is commonly secure. On the off chance that you attempt to modify the blockchain for your advantage and you get captured, you lose your whole staked token parity. It's a noteworthy hindrance for badly performing experts.

5.2 Implementation Set-up of Hashgraphy.

Creating a Hashgraph in a simulated environment. Illustrating how Proof-of-stake concept is already a part of Hashgraph.

The Swirlds hashgraph concord design is described via a simulated environmental set up, by coding a real-time hashgraph in Java programming language. This wraps the crucial calculation, from making exchanges, via finding their agreement arrange and timestamps.

5.2.1 Tools & Software used

Programming Language: Java Programming, Java SE 8: Java SE Development Kit 8 (or Java EE 8), Swirlds SDK Editor/Integrated Development Environment: Eclipse Oxygen 4.7
Security: Java 8 security: JCE Unlimited Strength Jurisdiction Policy Files

5.2.2 Process of creating prototype

Step 1. Members of the Hashgraph:

In this Hashgraph simulation 4 participants or members are present in this network.

- The members are Aek, Ben, Cate, Dev, are represented by unique identifiers namely 0,1,2,3.
- Each member has an IP address and communicates with other members over a network by connecting to that members port. Since it is a simulated environment all the Members are created on my local machine hence, they all have static IP's.
- I have setup specific ports for every member such as Aek:50204, Ben: 50205, Cate: 50206, Dev: 50207

The members in the simulated environment are configured using Java programming as shown below. I have extended the Address book from Swirlds SDK to configure them.

Swirlds	Addresses	Network	Security
0 A	Aek	[10.128.131.108] 50204	[10.128.131.108] 50204
1 B	Ben	[192.168.1.22] 50205	[192.168.1.22] 50205
2 C	Cate	[10.128.131.107] 50206	[10.128.131.107] 50206
3 D	Dev	[192.168.1.21] 50207	[192.168.1.21] 50207

The above are all the member addresses. Each address includes the nickname, name, internal IP address/port and external IP address/port.

Figure 5.11 Members of hashgraph

Step 2. Members creating events:

- Every participant initiates by generating an occasion, in a small information design storage, as shown below grey dot.

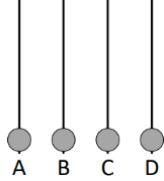


Figure 5.12 Participants initiate an occasion

- Each event is a container for zero or more transactions.
- Every event contains the following:
 - The 2 hashes of its 2 occasion.
 - Can optionally contain zero or more transactions
 - Timestamp when event was created

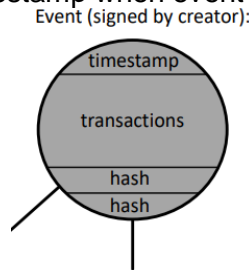


Figure 5.13 Event description

Step 3. Members Gossip:

A Section generates skinder rules, meaning that every participants randomly call all other participants to talk with them.

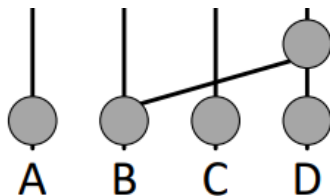


Figure 5.14 Members Gossip randomly

For example, participant B indirectly calls participant D. They are then connected over the internet & participant B shares every occasion which participant D doesn't know about. Here, this is just the single occasion that participant B has started.

- Dev records the way this match up occurred by making another occasion. This is the new circle, which has lines going straight down to his own last occasion, and slantingly down to Ben's last occasion. Therefore, the chart of occasions frames a record of how the individuals have conveyed.
- From a specialized viewpoint, Ben can abstain from sending Dev occasions he definitely knows. Ben first discloses to Dev what number of occasions he thinks about that were made by every part (i.e., 4 whole numbers). Dev discloses to Ben the equivalent. At that point they will both know precisely which occasions each ought to send the other. On the off chance that Ben has 13 occasions by Aek and Dev has 10, Ben sends Aek's last 3 occasions.

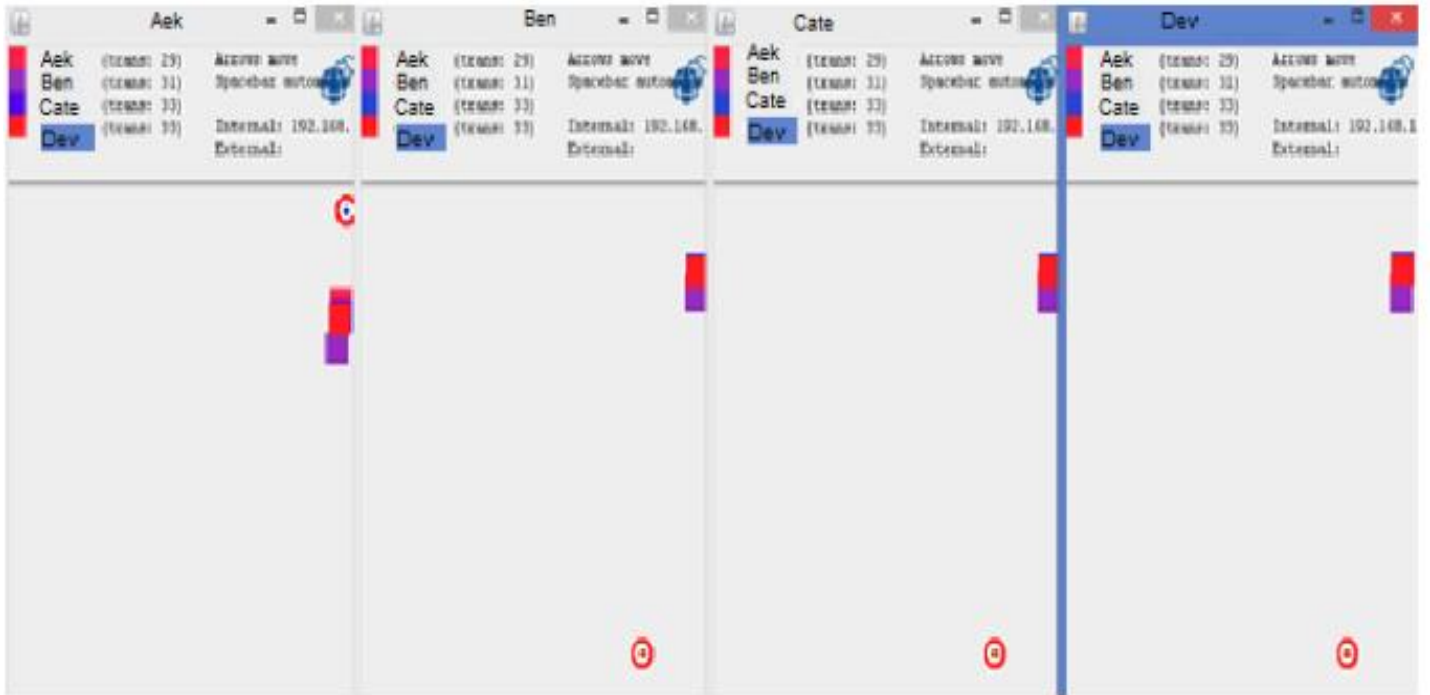


Figure 5.15 Live Demonstration of Members gossiping.

Step 4. Hashgraph Formation:

Dev then transmits Ben all his occasions (newly created included). Ben later produces a new occasion noting the fact they synced counting the hashes of the latest occasion without anyone else's input and the latest occasion by Dev.

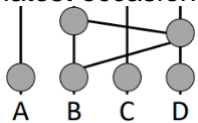


Figure 5.16 Hashgraph Formation

Ben then haphazardly picks Aek, and sends her every one of the 4 occasions he thinks about. She makes another one.

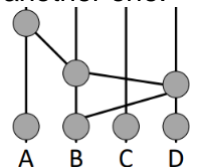


Figure 5.17 Hashgraph Formation proceeds everlastingly

This proceeds everlastingly, growing a coordinated non-cyclic chart upwards until the end of time. This is a diagram associated by cryptographic hashes, so it is known as a hashgraph.

Each occasion contains the hashes of the occasions underneath it and is carefully marked by its maker. So the whole diagram of hashes is cryptographically secure. It can generally develop, however the more established parts are permanent, as solid as the cryptographic hash and mark framework utilized. "Live Hashgraph depicting Round creations, Consensus order and Consensus timestamp, Creators ID for all events."



Figure 5.18 Live Demonstration of hashgraph formation

Working Code Snippet for the statistics and hashgraph picture, and appears in the window below all the settings, right below "display last ____ events".:-

```
/**
 * This panel has the statistics and hashgraph picture, and appears in the window below all the
 * settings, right below "display last ____ events".
 */

private class Picture extends JPanel {
    private static final long serialVersionUID = 1L;
    int ymin, ymax, width, n;
    double r;
    long minGen, maxGen;
    /** row to draw next in the window */
    int row;
    /** column to draw next in the window */
    int col;
    /** font height, in pixels */
    int textLineHeight;

    /**
     * find x position on the screen for the given event event
     *
     * @param event
     *         the event (displayed as a circle on the screen)
     * @return the x coordinate for that event
     */
    private int xpos(Event event) {
        return ((int) event.getCreatorId() + 1) * width / (numColumns + 1);
    }

    /**
     * find y position on the screen for the given event event
     *
     * @param event
     *         the event (displayed as a circle on the screen)
     * @return the y coordinate for that event
     */
    private int ypos(Event event) {
        return (event == null) ? -100
            : (int) (ymax
                - r * (1 + 2 * (event.getGeneration() - minGen)));
    }
}
```

Figure 5.20 Code snippet for x & y coordinates of the Hashgraph created in the above step.

Step 5. Creation of rounds:

- The dark occasion can be seen by each acclaimed observer in cycle 2. The red, green, and blue ways show how A2, B2, and D2, individually, would all be able to see the dark occasion. This only requires seeing, not unequivocally observing. This just requires seeing by the acclaimed observers. It doesn't make a difference whether C2 can see the dark occasion, on the grounds that C2 isn't renowned. Since the dark occasion is seen by the majority of the celebrated observers in cycle 2 (however in no prior round), it is said to have a round obtained of 2.

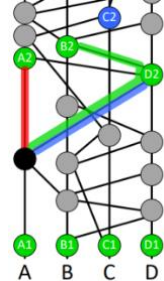


Figure 5.21 Creation of rounds logic

- This picture is a screenshot from the experimental setup Hashgraph Demo and demonstrates the piece of the hashgraph from about cycle 101 to 105. It originated from running it in moderate mode with 4 individuals Aek, Ben, Cate, Dev with the checkbox checked to demonstrate the round made.

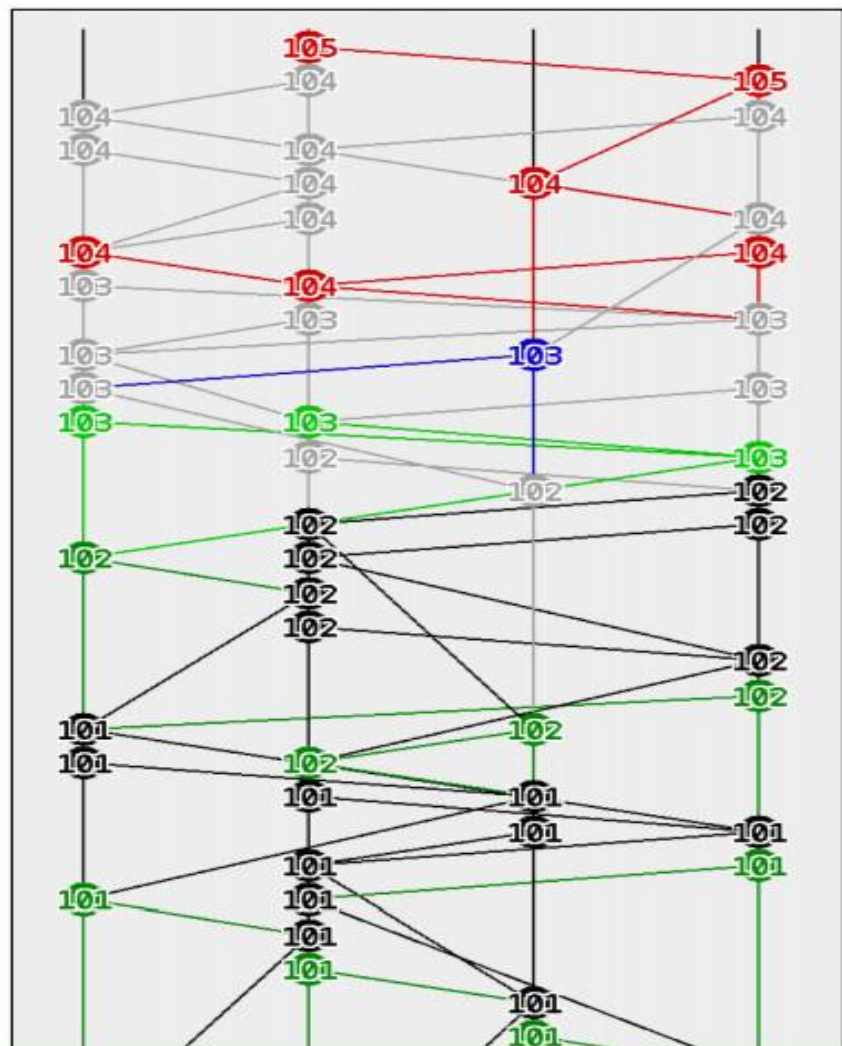


Figure 5.22 Live Demonstration of Creation of rounds

Step 6. Probability-one Theorem:

- There is a hypothesis saying that the decision will in the end (with likelihood one) as long as we include a coin round each tenth round of casting a ballot. In a piece of fortune round, gathering a supermajority makes an observer just vote (not choose). What's more, a non-supermajority makes it vote pseudo-haphazardly, by utilizing the centre piece of its own signature as its vote.

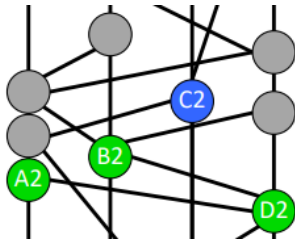


Figure 5.23 Probability- one Theorem

- Notice that in this model, we have now chosen the distinction of each observer in round 2. Once a round has the popularity chosen for the majority of its observers, it is conceivable to locate the round obtained and discover the agreement timestamp for another arrangement of occasions.

Step 7. Calculation of Consensus timestamp:

The agreement timestamp of the dark occasion can be found as follows (From the above chart):

- Find the soonest occasion X by Aek that is a predecessor of A2 and a descendant of the dark occasion.
- Similarly, locate the most punctual occasion Y by Ben that is a antecendent of B2 and descendant of the dark occasion.
- And comparatively for occasion Z by Dev.
- Take the timestamps on the occasions X, Y, Z that were placed in those occasions by their makers. Sort the majority of the timestamps all together. Take the centre one from the rundown (or the second of the two centre ones, if there is a considerably number of them). This middle timestamp is the agreement timestamp for the dark occasion.

Live Hashgraph depicting consensus and all the information as mentioned in the above image for 100 events.

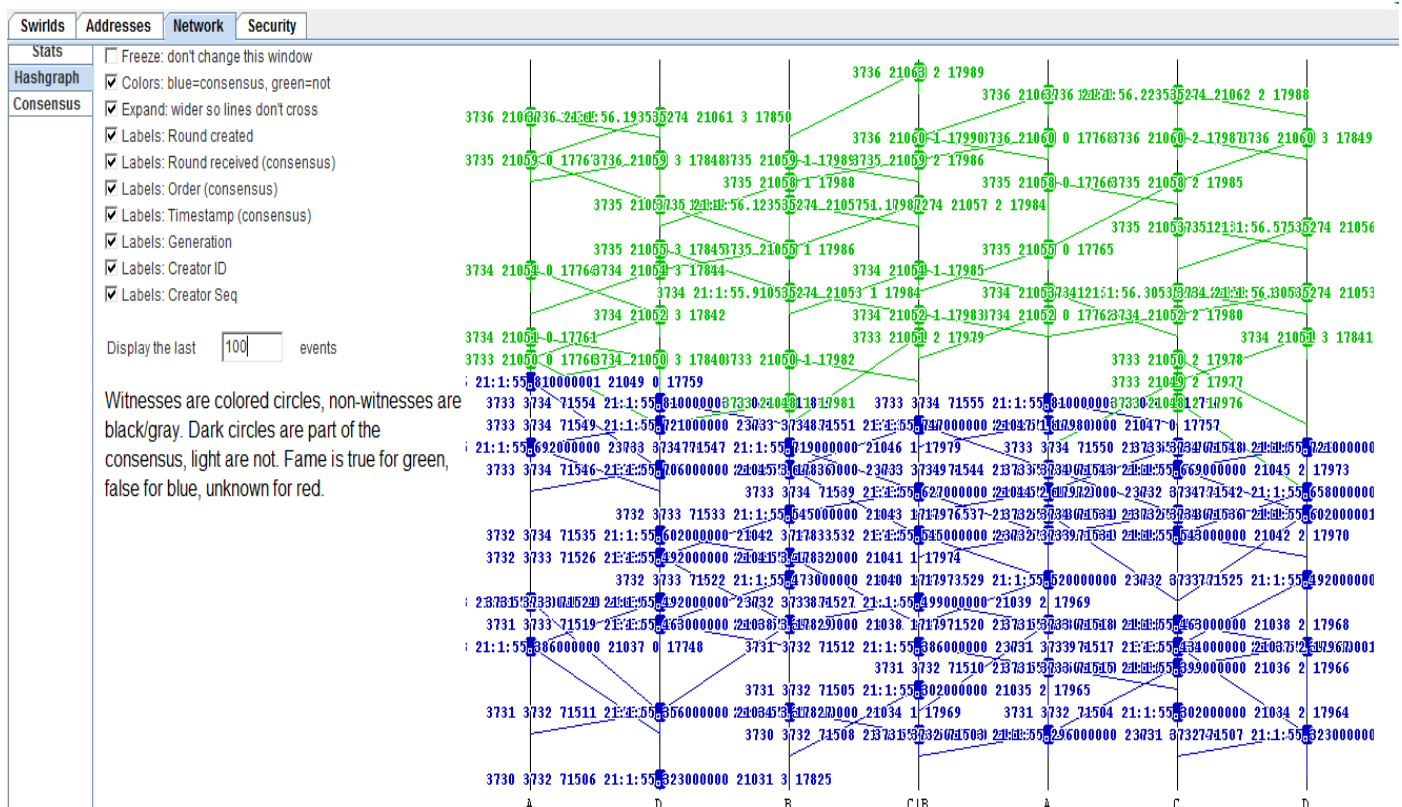


Figure 5.24 Live Demonstration describing colour for an event in the consensus algorithm.

As per Figure 5.24 the following implies:

Grey: non-witness

Green: witness(Famous)

Blue: not famous

Red: undecided fame

Dark colours (dark blue, dark green): Consensus

```
/** format the consensusTimestamp label */
DateTimeFormatter formatter = DateTimeFormatter.ofPattern("H:m:s.n")
    .withLocale(Locale.US).withZone(ZoneId.systemDefault());

/**
 * Return the color for an event based on calculations in the consensus algorithm A non-witness is gray,
 * and a witness has a color of green (famous), blue (not famous) or red (undecided fame). When the
 * event becomes part of the consensus, its color becomes darker.
 *
 * @param event
 *         the event to color
 * @return its color
 */
private Color eventColor(Event event) {
    if (simpleColorsCheckbox.getState()) { // if checkbox checked
        return event.isConsensus() ? LIGHT_BLUE : LIGHT_GREEN;
    }
    if (!event.isWitness()) {
        return event.isConsensus() ? DARK_GRAY : LIGHT_GRAY;
    }
    if (!event.isFameDecided()) {
        return event.isConsensus() ? DARK_RED : LIGHT_RED;
    }
    if (event.isFamous()) {
        return event.isConsensus() ? DARK_GREEN : LIGHT_GREEN;
    }
    return event.isConsensus() ? DARK_BLUE : LIGHT_BLUE;
}
```

Figure 5.25 Working Code Snippet that show checkboxes for every event like Round number, Consensus round received, consensus order number, consensus time stamp, generation number, member ID number, event creator sequence number:

```
// the following checkboxes control which labels to print on each vertex

/** the round number for the event */
private Checkbox labelRoundCheckbox;
/** the consensus round received for the event */
private Checkbox labelRoundRecCheckbox;
/** the consensus order number for the event */
private Checkbox labelConsOrderCheckbox;
/** the consensus time stamp for the event */
private Checkbox labelConsTimestampCheckbox;
/** the generation number for the event */
private Checkbox labelGenerationCheckbox;
/** the ID number of the member who created the event */
private Checkbox labelCreatorCheckbox;
/** the sequence number for that creator (starts at 0) */
private Checkbox labelSeqCheckbox;

/** only draw this many events, at most */
private TextField eventLimit;
```

Figure 5.26 Consensus information on every members profile in the Hashgraphy simulation application
Consensus information:

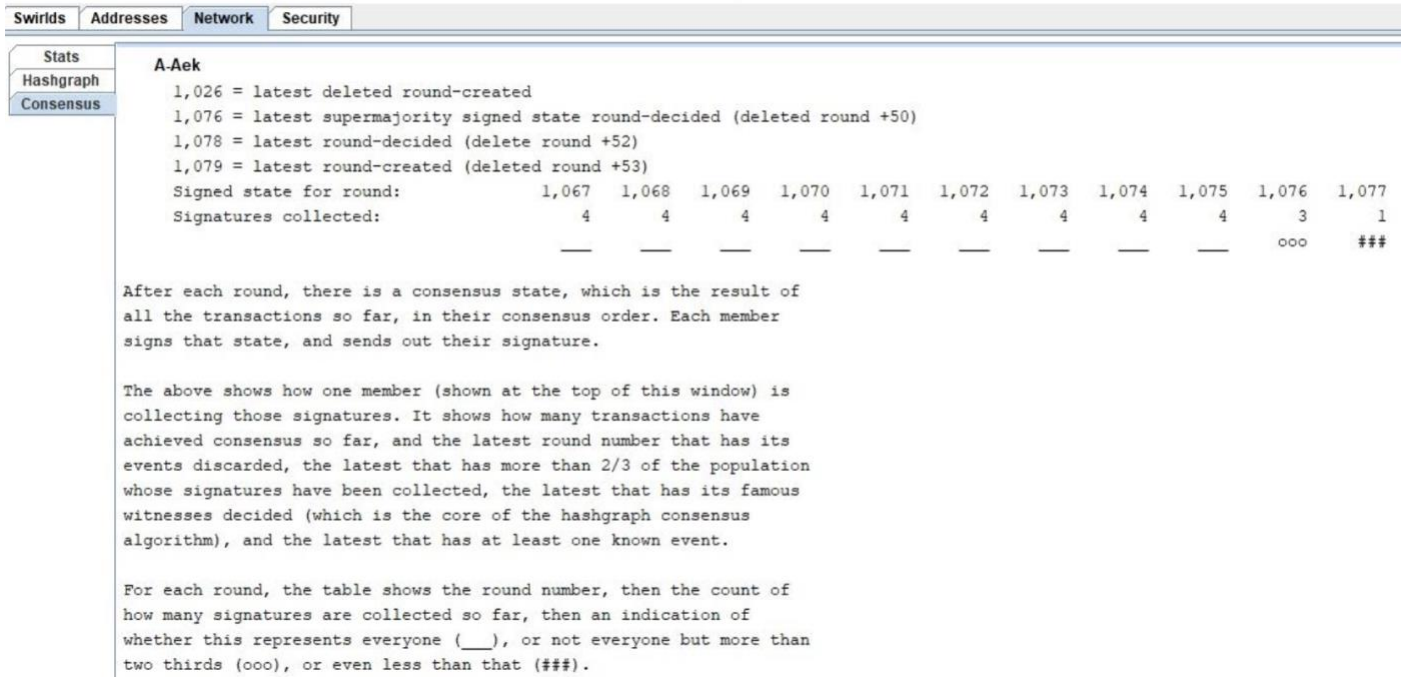


Figure 5.27 Aek consensus information

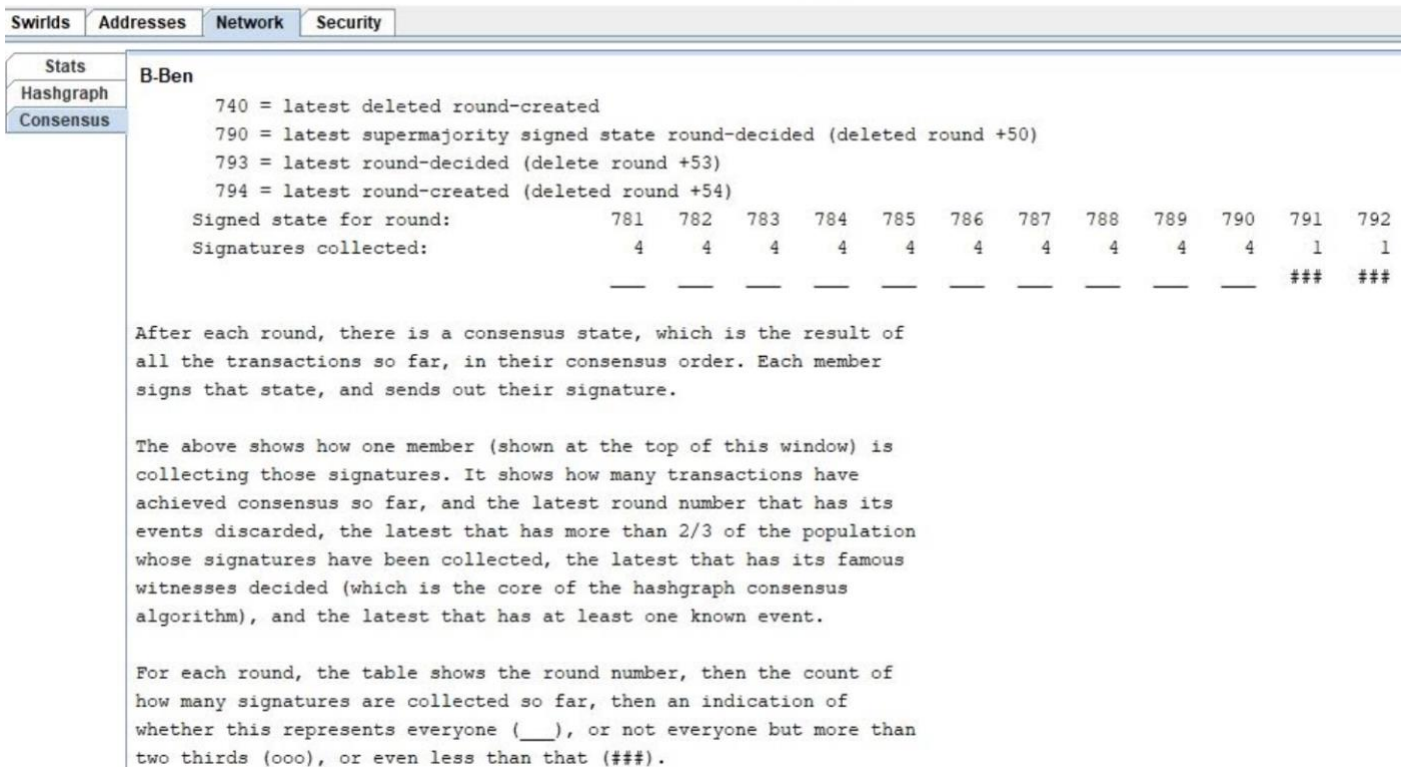


Figure 5.28 Ben's Consensus information.

Swirls	Addresses	Network	Security
Stats			
Hashgraph			
Consensus			

C-Cate												
801 = latest deleted round-created												
851 = latest supermajority signed state round-decided (deleted round +50)												
853 = latest round-decided (delete round +52)												
854 = latest round-created (deleted round +53)												
Signed state for round:	842	843	844	845	846	847	848	849	850	851	852	
Signatures collected:	4	4	4	4	4	4	4	4	4	4	1	1
	—	—	—	—	—	—	—	—	—	—	—	###

After each round, there is a consensus state, which is the result of all the transactions so far, in their consensus order. Each member signs that state, and sends out their signature.

The above shows how one member (shown at the top of this window) is collecting those signatures. It shows how many transactions have achieved consensus so far, and the latest round number that has its events discarded, the latest that has more than 2/3 of the population whose signatures have been collected, the latest that has its famous witnesses decided (which is the core of the hashgraph consensus algorithm), and the latest that has at least one known event.

For each round, the table shows the round number, then the count of how many signatures are collected so far, then an indication of whether this represents everyone (—), or not everyone but more than two thirds (ooo), or even less than that (###).

Figure 5.29 Cate's Consensus information .

SwirldsAddressesNetworkSecurity

StatsHashgraphConsensus

:D-Dev

876 = latest deleted round-created

926 = latest supermajority signed state round-decided (deleted round +50)

928 = latest round-decided (delete round +52)

929 = latest round-created (deleted round +53)

Signed state for round:917918919920921922923924925926927

Signatures collected:444444444441

— — — — — — — — — — ###

After each round, there is a consensus state, which is the result of all the transactions so far, in their consensus order. Each member signs that state, and sends out their signature.

The above shows how one member (shown at the top of this window) is collecting those signatures. It shows how many transactions have achieved consensus so far, and the latest round number that has its events discarded, the latest that has more than 2/3 of the population whose signatures have been collected, the latest that has its famous witnesses decided (which is the core of the hashgraph consensus algorithm), and the latest that has at least one known event.

For each round, the table shows the round number, then the count of how many signatures are collected so far, then an indication of whether this represents everyone (___), or not everyone but more than two thirds (ooo), or even less than that (###).

Figure 5.30 Dev's Consensus Information:

Step 8. Voting:

- Hashgraph doesn't use proof-of-work. It uses virtual-voting. There is a hypothesis that on the off chance that any observer can "choose" yes or no, that is the aftereffect of the race, and it is ensured that every single different observer will choose a similar way. In this model, B4 could choose the decision. On the off chance that it had gathered votes that were all the more on an equal basis among YES and NO, at that point it would have neglected to choose. All things considered, we can consider D4. On the off chance that D4 additionally neglects to choose, maybe A4 or C4 may choose. In the event that none of the cycle 4 witnesses can choose, every one of them will just cast a ballot as per most of the votes they gathered (casting a ballot YES if there should be an occurrence of a tie). All things considered, it will be up to the cycle 5 observers to gather cast a ballot from the cycle 4 witnesses. Maybe the cycle 5 observes will most likely choose. The casting a ballot proceeds until it inevitably achieves a round where some observer can choose the decision.

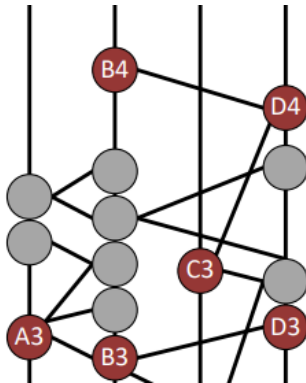


Figure 5.31 Voting method

- Below are the screen shots of Hashgraph members Aek, Ben, Cate, Dev and displaying Network based information. This information is gathered from the created Hashgraph as shown in the above screen shots and is individually passed to the respective members report in text format.

Aek:

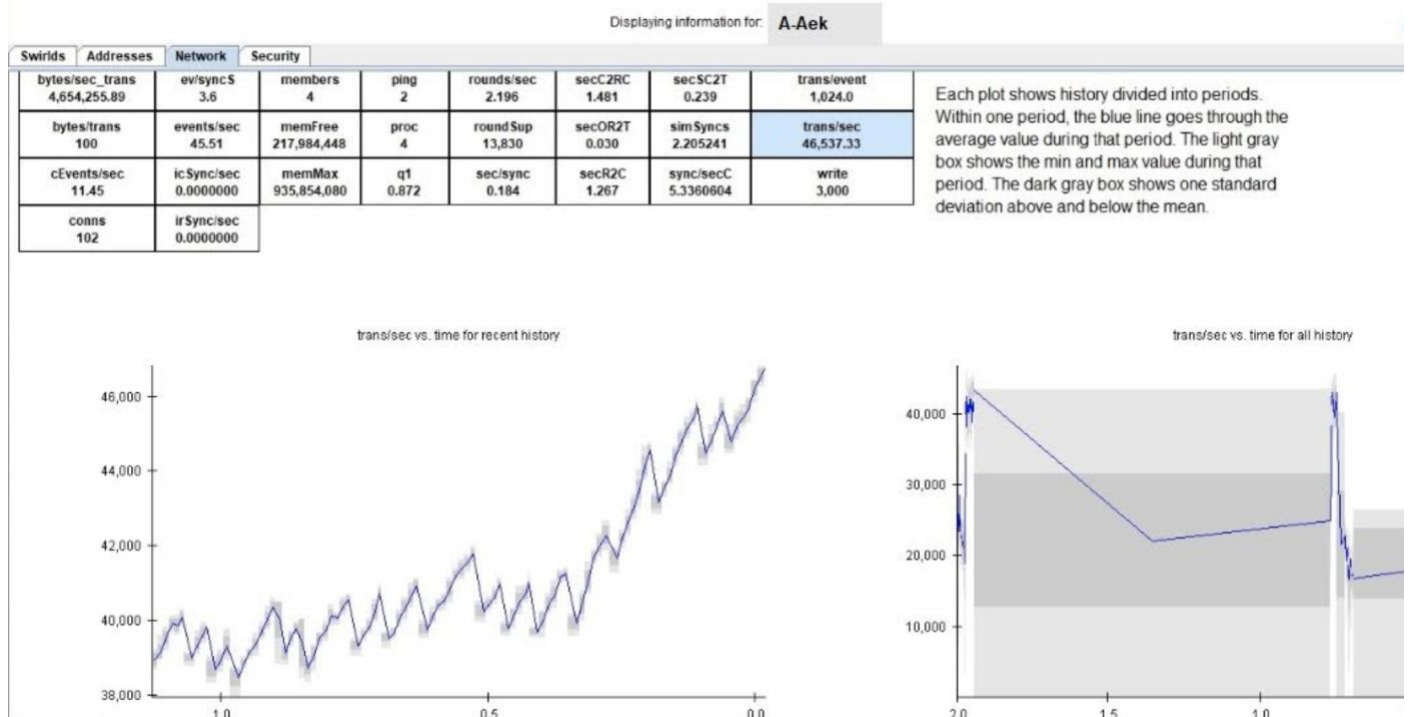


Figure 5.32 Aek trans/sec maximum 46,537

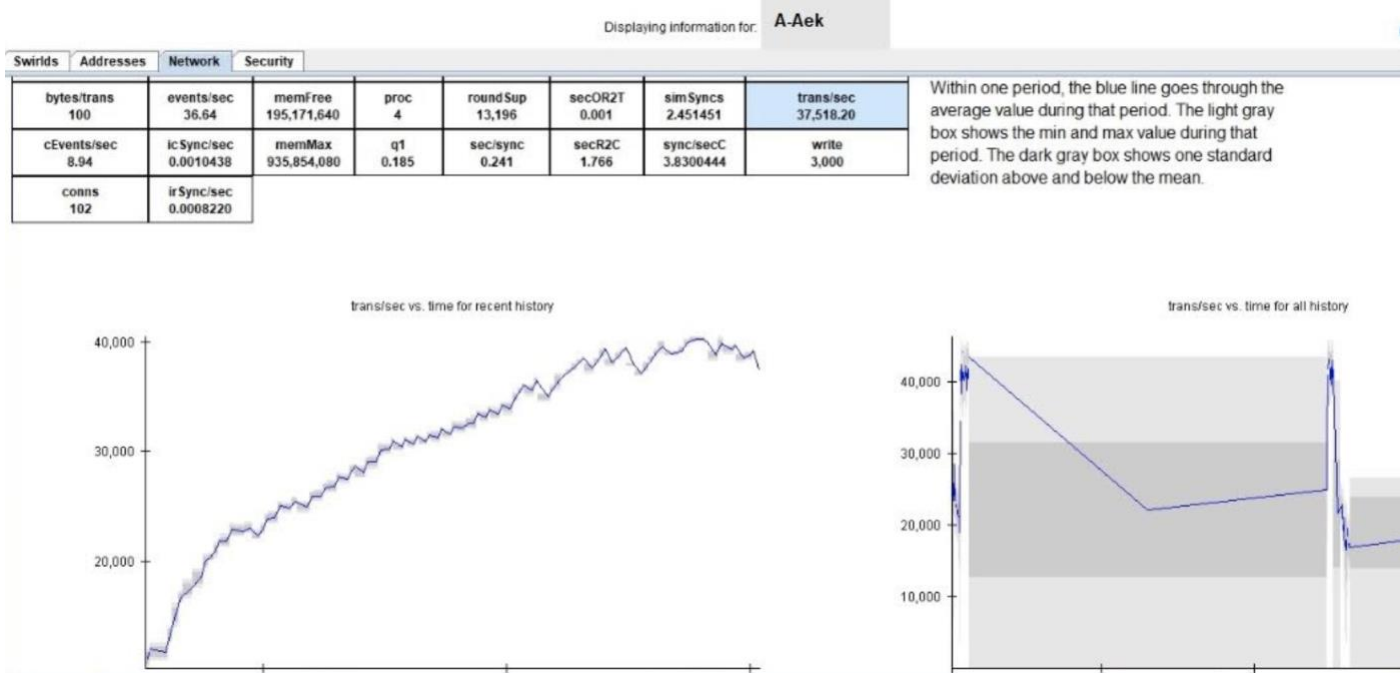


Figure 5.33 Aek trans/sec minimum 37,518

It is observed that when bytes/trans is constant as 100 for all the members, there is no fixed transactions per second transmission in any of the cases. The trans/sec is varying. For example, in both the cases of Aek, the number of transactions varies from 0.76 to 22.24. Infact the highest number of transactions have been recorded to be 35. Moreover a closer observation of the events in Sync are 3.3. The vital point is that when you get an occasion in a state of harmony, you can quickly compute its round made. What's more, any other individual accepting it will figure a similar number. This is completely assured.

Ben:

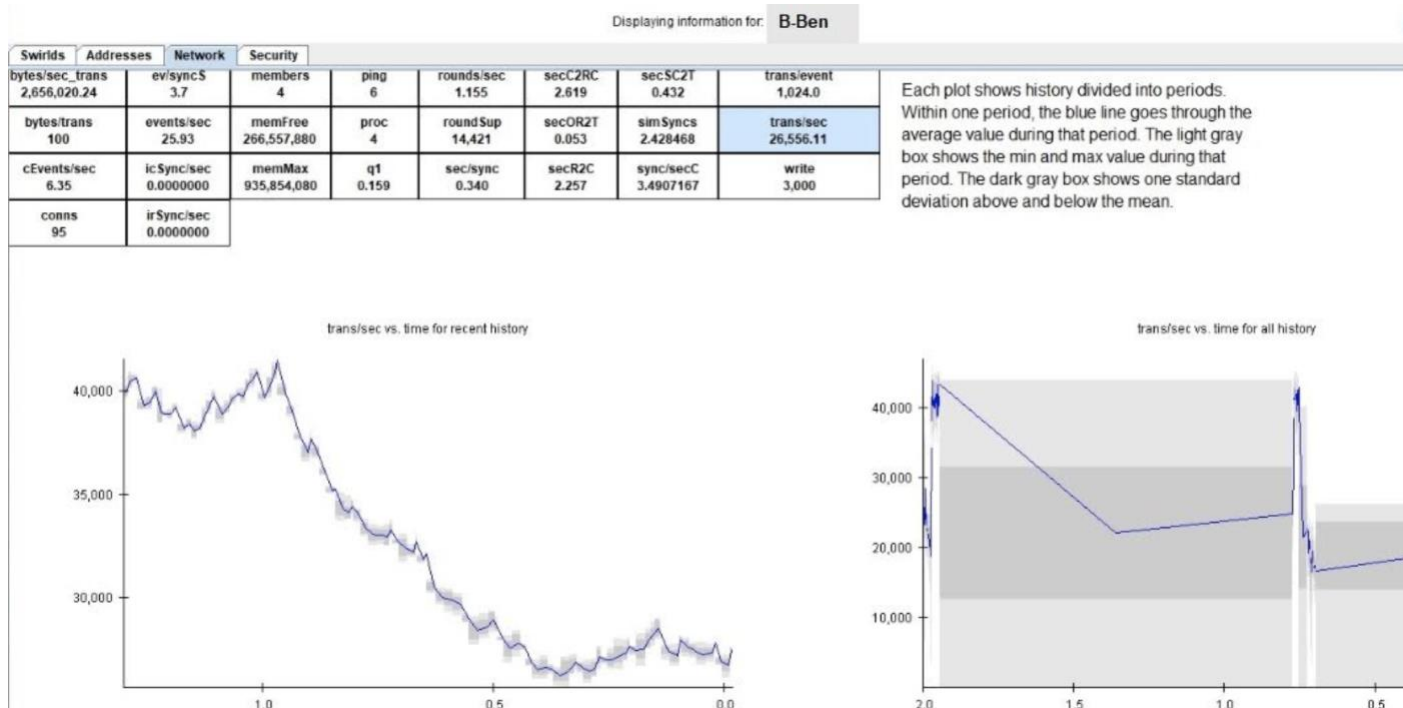


Figure 5.34 Ben trans/sec maximum 26,556

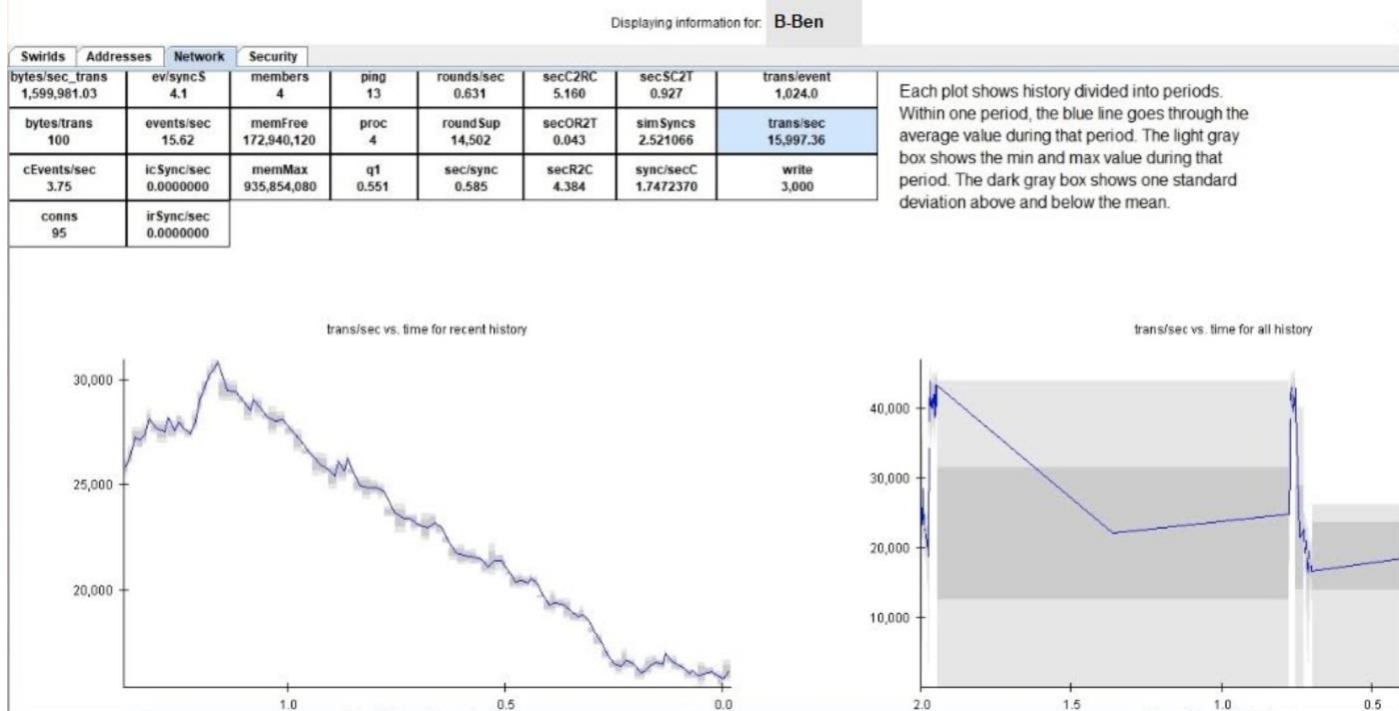


Figure 5.35 Ben trans/sec minimum 15,997

Cate:

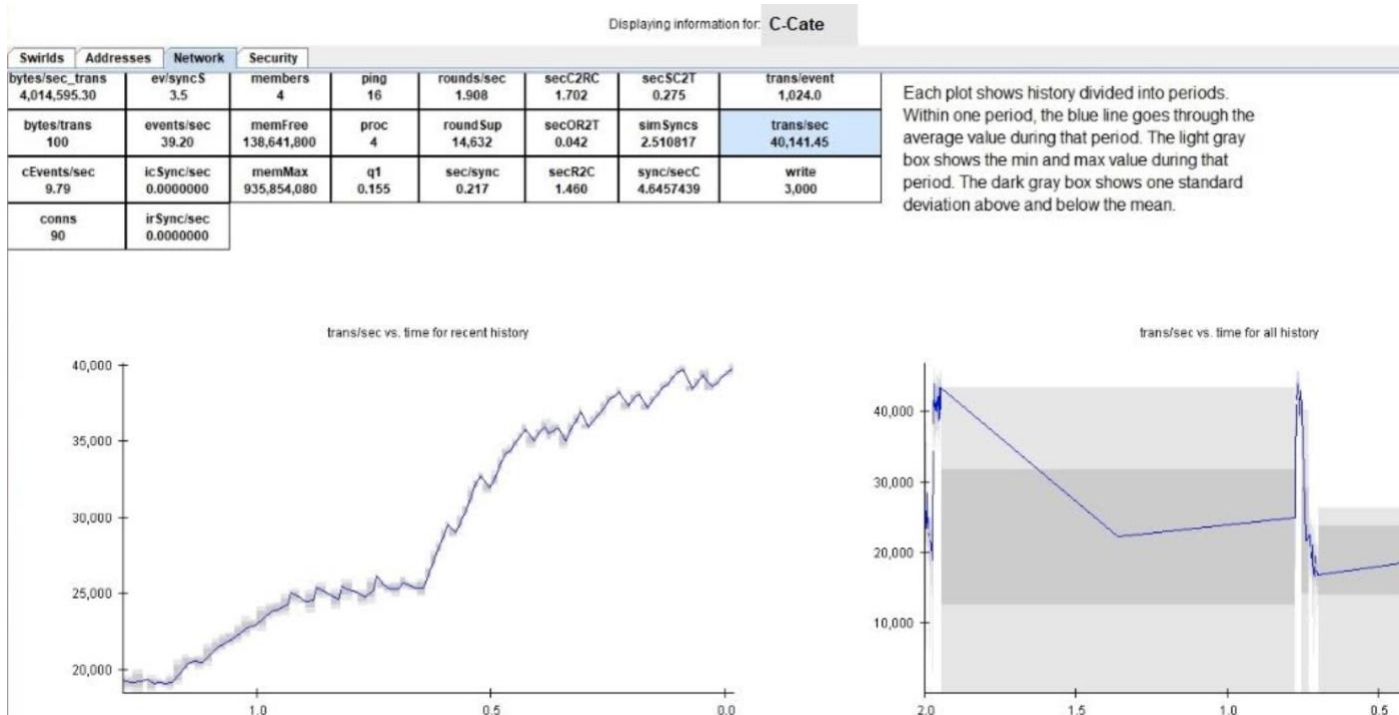


Figure 5.36 Cate trans/sec maximum 40,141

Displaying information for: **C-Cate**

Swirls	Addresses	Network	Security					
bytes/sec_trans 2,789,529.40	ev/syncS 3.8	members 4	ping 12	rounds/sec 1.133	secC2RC 2.791	secSC2T 0.440	trans/event 1,024.0	
bytes/trans 100	events/sec 27.24	memFree 172,688,328	proc 4	roundSup 14,697	secOR2T 0.057	simSynCS 2.499794	trans/sec 27,893.53	
cEvents/sec 6.92	icSync/sec 0.0000000	memMax 935,854,080	q1 0.292	sec/sync 0.330	secR2C 2.380	sync/secC 3.3581872	write 3,000	
conns 90	irSync/sec 0.0000000							

Each plot shows history divided into periods. Within one period, the blue line goes through the average value during that period. The light gray box shows the min and max value during that period. The dark gray box shows one standard deviation above and below the mean.

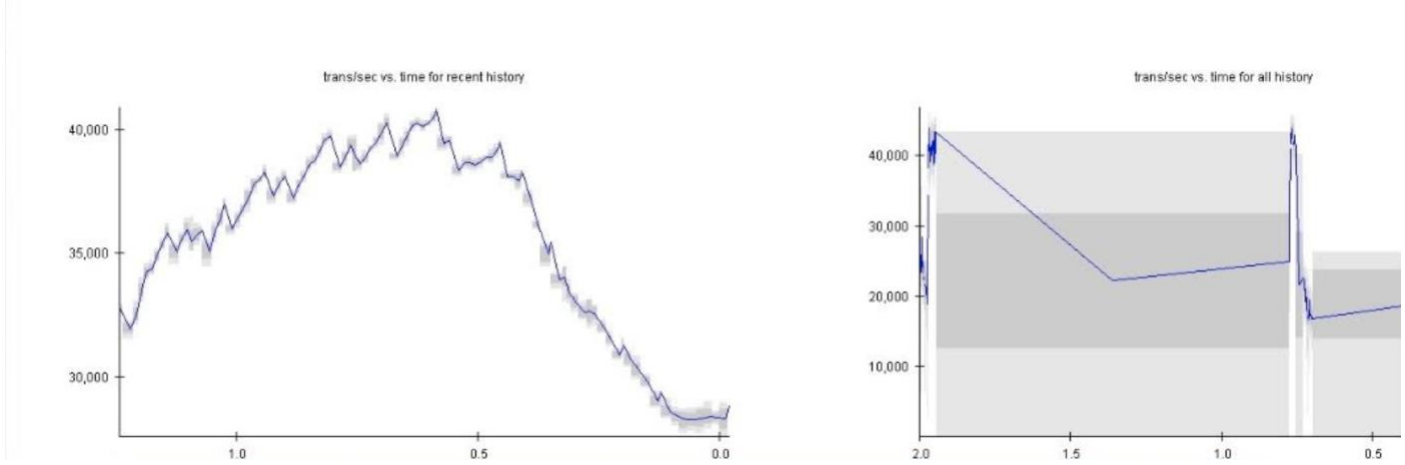


Figure 5.37 Cate trans/sec minimum 27,893

Dev:

Displaying information for: **D-Dev**

Swirls	Addresses	Network	Security					
bytes/sec_trans 4,590,204.15	ev/syncS 3.6	members 4	ping 3	rounds/sec 2.174	secC2RC 1.422	secSC2T 0.234	trans/event 1,024.0	
bytes/trans 100	events/sec 44.82	memFree 325,441,864	proc 4	roundSup 17,913	secOR2T 0.030	simSynCS 2.193821	trans/sec 45,896.22	
cEvents/sec 11.44	icSync/sec 0.0000000	memMax 935,854,080	q1 0.207	sec/sync 0.177	secR2C 1.208	sync/secC 4.8705973	write 3,000	
conns 105	irSync/sec 0.0000000							

Each plot shows history divided into periods. Within one period, the blue line goes through the average value during that period. The light gray box shows the min and max value during that period. The dark gray box shows one standard deviation above and below the mean.

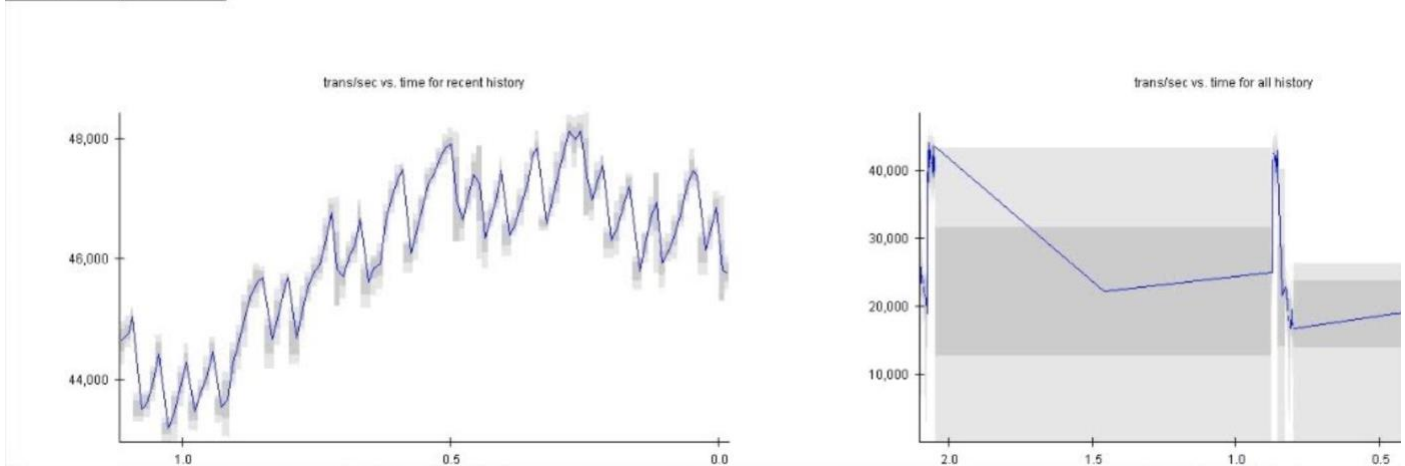


Figure 5.38 Dev trans/sec maximum 45,896

Swirls	Addresses	Network	Security				
bytes/sec_trans 1,621,602.54	ev/syncS 3.9	members 4	ping 18	rounds/sec 0.584	secC2RC 4.850	secSC2T 0.822	trans/event 1,024.0
bytes/trans 100	events/sec 15.83	memFree 159,076,688	proc 4	roundSup 14,990	secOR2T 0.169	sim Syncs 2.207526	trans/sec 16,213.21
cEvents/sec 3.92	icSync/sec 0.0000000	memMax 935,854,080	q1 0.098	sec/sync 0.541	secR2C 4.250	sync/secC 1.8375276	write 3,000
conns 97	irSync/sec 0.0000000						

Each plot shows history divided into periods. Within one period, the blue line goes through the average value during that period. The light gray box shows the min and max value during that period. The dark gray box shows one standard deviation above and below the mean.

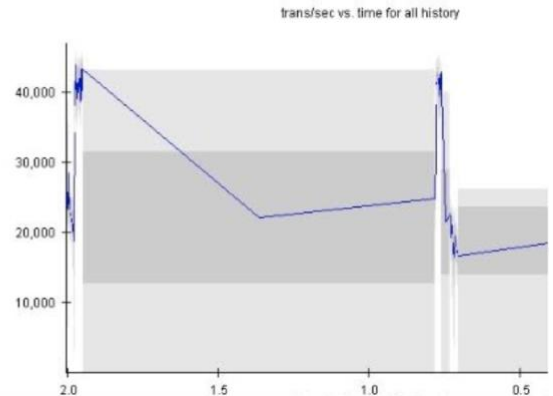
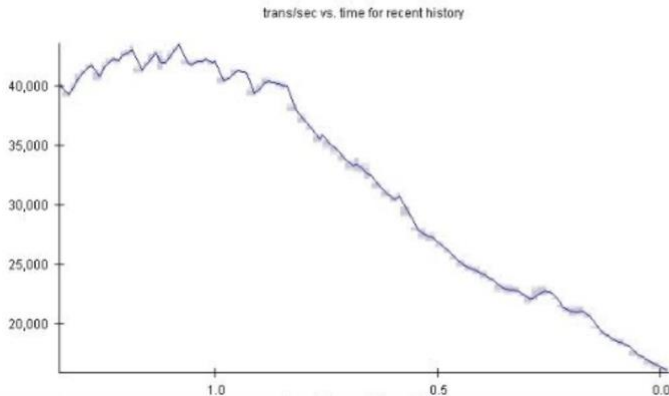


Figure 5.39 Dev trans/sec minimum 16,213

The information that is neatly arranged in a tabular format for each member respectively is called by the graphics context from the original hashgraph.

```
/**
 * called by paintComponent to draw text at the top of the window
 *
 * @param g
 *         the graphics context passed to paintComponent
 * @param text
 *         a String.format formatting string
 * @param value
 *         the value to pass to String.format to be formatted
 */
private void print(Graphics g, String text, double value) {
    g.drawString(String.format(text, value), col,
        row++ * textLineHeight - 3);
}
```

Figure 5.40 Working code snippet of the graphic context

The graphic context `g` as described above is used to retrieve the matching information for a Member id. For example, Aek has a member id 0 (zero), this 0 will be used to retrieve its information from the hashgraph and display it in a tabular format on Aek's profile. Likewise for Ben: 1, Cate: 2, Dev: 3.


```

/** {@inheritDoc} */
public void paintComponent(Graphics g) {
    super.paintComponent(g);
    g.setFont(new Font(Font.MONOSPACED, 12, 12));
    FontMetrics fm = g.getFontMetrics();
    int fa = fm.getMaxAscent();
    int fd = fm.getMaxDescent();
    textLineHeight = fa + fd;
    int numMem = platform.getState().getAddressBookCopy().getSize();
    calcNames();
    width = getWidth();

    row = 1;
    col = 10;
    double createCons = platform.getStats().getStat("secC2C");
    double recCons = platform.getStats().getStat("secR2C");

```

Figure 5.41 Working code snippet of retrieving graphic context for each member

- The code here shows the getStats() functionality's usage to retrieve trans/sec, events/sec, duplicate event %, bad events per sec.

```

print(g, "%5.0f trans/sec",
    platform.getStats().getStat("trans/sec"));
print(g, "%5.0f events/sec",
    platform.getStats().getStat("events/sec"));
print(g, "%4.0f%% duplicate events",
    platform.getStats().getStat("dupEv%"));
print(g, "%5.3f bad events/sec",
    platform.getStats().getStat("badEv/sec"));

print(g, "%5.3f sec, propagation time", createCons - recCons);
print(g, "%5.3f sec, create to consensus", createCons);
print(g, "%5.3f sec, receive to consensus", recCons);
print(g, "Internal: " + Network.getInternalIpAddress() + " : "
    + platform.getAddress().getPortInternalIpv4(), 0);
print(g, "External: "
    + (Network.getExternalIpAddress().equals("") ? ""
        : Network.getExternalIpAddress() + " : " + platform
            .getAddress().getPortExternalIpv4()),
    0);

```

Figure 5.42 Working code snippet of getStats functionality

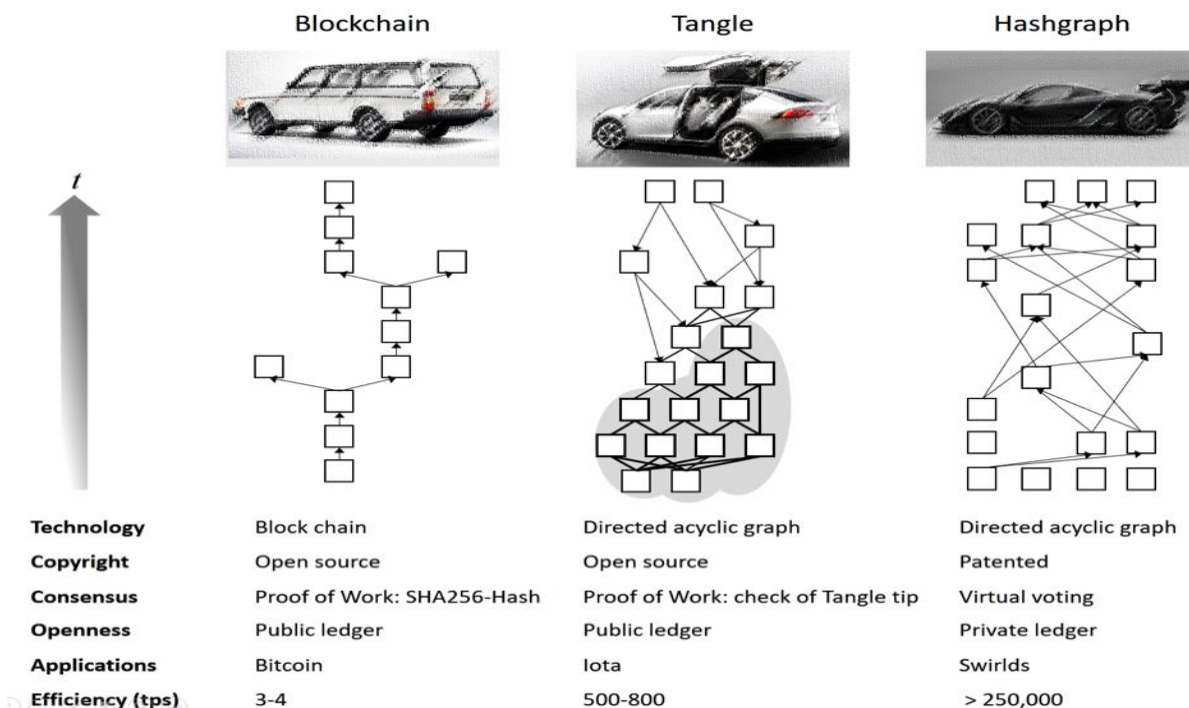


Figure 5.43 Factors of three popular DLTs.

5.3 Derivation from implementation

As per the implementation design observations in this chapter, Blockchain is capable of doing only up to 3 transactions per second (tps) for a bandwidth of 100 Mbps and packet size of 100 bytes per transaction. Therefore, 4 tps would be the efficiency of the system. And hashgraph is observed to do up to 45,000 transactions per second for a bandwidth of 100 Mbps and packet size of 100 bytes per transaction. The efficiency of the system depends on the bandwidth, which means for a higher bandwidth greater than 100 Mbps the transactions per second will surely be greater than 45,000. However, for this current implementation scenario the efficiency of the system on an average is 45,000 tps.

5.4 Chapter Summary

This chapter covered the implementation of a Blockchain Proof-of-stake(PoS). It demonstrated the coding of a Blockchain algorithm using the PoS and recorded the transactions per second of the system over a Bandwidth of 100 Mbps with 100 bytes/transaction data size. Likewise It demonstrated the coding of a Hashgraph algorithm and recorded the transactions per second of the system over the same Bandwidth of 100 Mbps with 100 bytes/transaction data size.

The result of these two observations have been compared and documented in this chapter. It is clear that Hashgraph is faster and more efficient than Blockchain. Speed of the Hashgraph as implemented in this chapter is 45000 tps compared to Blockchain which does just 3 tps for the same bandwidth. Since, Hashgraph is clearly faster, all the members reach consensus faster compared to Blockchain. And that makes Hashgraph more efficient compared to Blockchain.

The next chapter focuses on Security of Hashgraph and discusses the Byzantine Fault Tolerance theorem that makes Hashgraph resilient to attacks.

CHAPTER 6

Security of Hashgraph

This chapter gives complete description on Security of Hashgraph. Section 6.1 describes us about how the hashgraph is secured. Section 6.2 covers Byzantine fault tolerance theorem. Section 6.3 gives us the comparison between Asynchronous Byzantine fault tolerance vs Partially Asynchronous Byzantine fault tolerance. Section 6.4 gives us the mathematical proof of hashgraph being fully Asynchronous Byzantine fault tolerant & also resilient to DDoS attacks. Section 6.5 tells us how is hashgraph resilient to Sybil attacks. Section 6.6 summarizes the chapter.

6.1 Security of Hashgraph

Each mutual record, shared database or a common world in the hashgraphy setting is named as a swirld. As appeared in the code underneath, each swirld has an exceptional identifier. The identifier of the current swirld is appeared in two structures.

The first is a base-62 encoding <> in angular braced sections.

The second is an arrangement of words.

Expecting that more than 66% of the populace are straightforward, the one of a kind identifier for a given swirld will never show signs of change. Also, if the swirld ever forks or parts or branches, just a single branch will keep indistinguishable identifier from the first. So form of the swirld is the 'authority' or 'genuine' successor, and the rest are new Swirls.

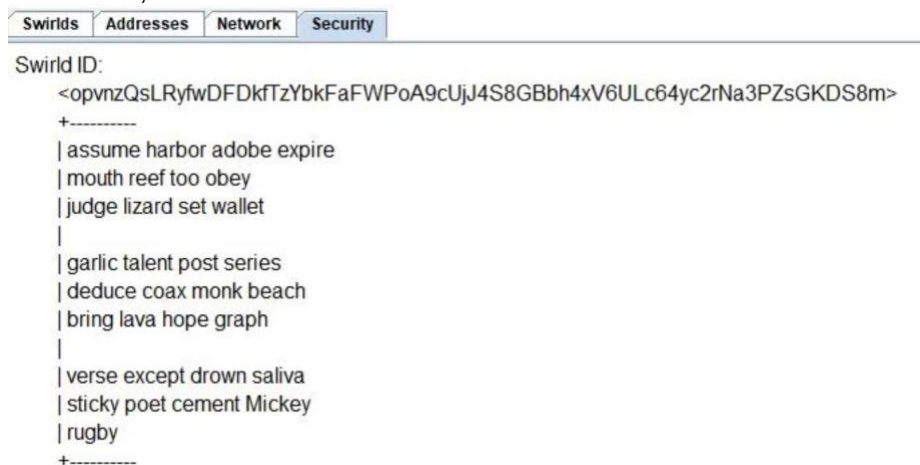


Figure 6.1 Structure of swirld identifier

6.2 Byzantine Fault Tolerance Theorem

Every occasion x made by a fair part will in the long run be served an awarded position in the all-out request of occasions, with probability 1.

Evidence: All legit individuals will in the long run learn of x , by the definition of genuine and the suppositions that the assailants who control the web should in the end enable any two legitimate individuals to impart. In this manner, there will in the end be a round where all the one of a kind popular observers are relatives of x . In this way in that round, or perhaps prior, there will be a round r where all the well-known observers are relatives of x . At that point x is granted round of r , and an accord timestamp of the middle of when those individuals first acquired it, and its agreement put in history will be fixed. Moreover, it is preposterous to later find another occasion y that will precede x in the agreement arrange. Since, to come prior in the accord history, y would must have a acquired round not exactly or equivalent to r . That would imply that all the celebrated observers in round r more likely than not acquired y . In any case, when the

arrangement of renowned observers is known for a round, the majority of their precursors are additionally known, so there is no real way to find new predecessors for them later on as the hashgraph develops. Besides, it isn't feasible for a round to increase new renowned observers later on, once the acclaim of all the known observers in that round are known. Any new round r witness that is found later on won't be a predecessor of the known round $r + 1$ witnesses (of which there are more than $2n/3$), thus the accord will promptly be achieved that it isn't celebrated. Accordingly, when an occasion is doled out a put in the absolute request, it will never show signs of change its position, neither by swapping with another known occasion, nor by new occasions being found later and being embedded before it.

At last, byzantine fault tolerant (BFT) implies three things:

- 1) We will come to an agreement;
- 2) We will realize when we've come to an agreement.
- 3) We're never wrong—it is numerically ensured that every other person will achieve precisely the same agreement. That is byzantine.

6.3 Asynchronous Byzantine Fault Tolerance v/s Partially Asynchronous Byzantine Fault Tolerance

BFT can be either asynchronous byzantine (aBFT) or in part nonconcurrent byzantine. Both are numerically ensured, with the distinction being the dimension of suspicions you're making about your condition. aBFT as in a hashgraph would expect detestable on-screen characters exist in the network since they do. Be that as it may, in case you're making broken presumptions like there are no botnets on the planet, it would be incompletely Asynchronous BFT—because botnets do exist in reality. In the event that you begin a math confirmation by expecting there are no botnets on the planet, at that point it's unlikely of what your verification implies in light of the fact that you're living in a dreamland.

Byzantine blame tolerant (BFT) is the end. Nonconcurrent versus somewhat offbeat byzantine blame tolerant (aBFT) are the suppositions toward the start.

6.4 Mathematical Proof of Hashgraph being fully Asynchronous Byzantine Fault Tolerant and resilient to DDoS and Sybil attacks

6.4.1 Hashgraph is Byzantine

- Unlike alternate frameworks, hashgraph is completely asynchronous byzantine. This implies it makes no suspicions about how quick messages are communicated across the web, making it strong against DDoS attacks, botnets, and firewalls. Hashgraph is scientifically ensured to achieve agreement and be secure in consideration of fewer than 33% of members being malicious (which is something that must dependably be expected for DLT). It is important that the expression "Byzantine" is once in a while utilized in a more fragile sense. In any case, here, it is utilized in its unique, more grounded sense that (1) each member of the Hashgraph in the end realizes consensus has been achieved (2) attackers may plan to do something illicit and (3) attackers even control the web itself (with a few breaking points). Hashgraph is Byzantine, even by this more grounded definition. [36]
- With regard to the virtual casting a ballot idea of Hashgraph. Byzantine support frameworks have been created for Byzantine understanding that commonly trade numerous messages for the individuals to cast a ballot. For n individuals to choose a solitary YES/NO inquiry, a few frameworks can require $O(n)$ messages to be sent over the system. Different frameworks can require $O(n^2)$, or even $O(n^3)$ messages crossing the system per paired choice. A calculation for a solitary YES/NO choice would then be able to be stretched out to choosing a request on a lot of exchanges, which may additionally build the vote traffic. Hashgraph sends no votes at all over the system, since all casting a ballot is virtual.

6.4.2 Blockchain is non- Byzantine

Blockchain does not have a certification of Byzantine support, in light of the fact that a part never achieves sureness that consensus has been accomplished (there's only a likelihood that ascents after some time). Blockchain is additionally non-Byzantine since it doesn't naturally manage arrange parcels. On the off chance that a gathering of miners is disengaged from whatever remains of the web, that can enable various chains to develop, which struggle with one another on the request of exchanges.

6.4.3 Hashgraph is resilient to DoS/DDoS Attack

The hashgraph is DoS/DDoS safe. Both blockchain and hashgraph are conveyed such that opposes Denial of Service (DoS) attacks. An aggressor may surge one part or mineworker with parcels, to incidentally detach them from the web. Be that as it may, the network all in all will keep on working ordinarily. An assault on the framework in general would require flooding a substantial portion of the individuals with parcels, which is progressively troublesome. There have been various proposed options to blockchain dependent on pioneers or round robin. These have been proposed to maintain a strategic distance from the evidence of-work expenses of blockchain. Be that as it may, they have the downside of being delicate to DoS attacks. In the event that the aggressor assaults the present chief, and changes to assaulting the new pioneer when one is picked, at that point the assailant can solidify the whole framework, while as yet assaulting just a single PC at any given moment. Hashgraph maintains a strategic distance from this issue, while still not requiring verification of-work. [57]

Based on the concept of Proof-of-stake, I have come up with these generalizations and enhancements based on the facts of Hashgraph's white paper.

Up until now, it has been expected that each part is equivalent. The above calculations hint to things contingent upon "more than $2n/3$ of the individuals" and "at any rate half of the celebrated observer events". They likewise utilize the possibility of a "middle" of a lot of numbers. The evidence indicates Byzantine union when more than $2n/3$ of the individuals are straightforward. It is anything but difficult to alter the calculation to enable individuals to be unequal. Every part can be accepted to have some positive number related with them, known as their "stake". At that point, the votes would be supplanted with weighted votes, and the medians with weighted medians, where votes are weighted relative to the voter's stake. In the majority of the above definitions, calculations, and confirmations, characterize "more than $2n/3$ individuals" to signify "a lot of individuals whose all-out stake is more than $2n/3$, where n is the absolute stake all things considered". The "middle of the timestamps of occasions in S " would turn into "the weighted middle of the timestamps in S , weighted by the stake of the maker of every occasion in S ". The weighted middle can be thought of as taking every occasion y in S , and putting numerous duplicates of the timestamp of y into a sack, where the quantity of duplicates parallels the stake of the part who made y . At that point take the middle of the timestamps taken care of. [36]

The Byzantine confirmation connected as long as the assailants comprised under $1/3$ of the populace. With these new definitions, it will presently apply when the aggressors together have a stake that is under $1/3$ of the complete stake all things considered.

This new verification of-stake framework is more broad than the unweighted framework. It can in any case be utilized to execute the unweighted framework, by essentially giving each part a stake of 1. Be that as it may, it can likewise be utilized to give better conduct. For instance, the stake may be relative to how much a part is trusted. Maybe individuals who have been examined here and there ought to be confided in more than others. Or on the other hand it could be utilized to give more noteworthy load to individuals who have a more prominent enthusiasm for the framework in general working legitimately. A digital money may utilize every part's number of coins as their stake, in light of the fact that those with more coins have a more noteworthy enthusiasm for guaranteeing the framework runs easily. Or then again a network could be begun by a gathering of individuals with shared trust, every one of which is given an equivalent stake. At that point, each current part could be permitted to welcome subjectively numerous new individuals to join, subject to the requirement that the inviter must part their stake with the invitee. This would dishearten a Sybil assault, where one part welcomes countless manikin accounts, so as to control the casting a ballot. The "stake record" is the rundown of individuals and the measure of stake possessed by every part. Up until this point, it has been expected that the stake record is all around known, and is perpetual. It is anything but difficult to loosen up that suspicion. Accept that there is a specific type of exchange that changes the stake record. The people group may set up tenets toward the starting, administering which

such exchanges are legitimate. For instance, every part could be permitted to welcome different individuals, up to an aggregate of at most 10 new individuals. Or on the other hand maybe anybody welcoming another part should all the while give the new part their very own bit stake. The legitimacy of such an exchange may rely upon the correct request of the exchanges in the accord arrange. For instance, if the standard is that just a single new part can be welcomed, and Aek welcomes Cate in the meantime Ben welcomes Dev, at that point then whichever welcome starts things out in the accord request will succeed, and the other will come up short.

The majority of this can be suited. At the point when the accord calculation wraps up the subject of which round r firsts are well known, right then and there it ends up conceivable to discover precisely which occasions will have an acquired round of r , and to compute their correct position in the agreement arrange. Around then, every one of the exchanges in those occasions can be prepared, and the principles can be counselled to see which are legitimate, and the substantial exchanges can be connected. This may change the stake record. On the off chance that the stake record changes, the calculation ought to be re-kept running for all occasions in round r and later. This may change the figuring of which occasions are emphatically observed, of occasion round numbers, of which occasions are observers, and of which are well known observers. Note that when choosing which round r witnesses are well known, the computations are finished utilizing the old stake record. The voting in favour of round r may proceed with a few rounds into the future, all utilizing the old stake record. Once round r is settled, the future rounds will reshuffle, and the figuring for round $r + 1$ acclaimed observer will be finished utilizing the new stake record.

This methodology enables all individuals to be in concession to precisely what stake record is being utilized for some random count. That guarantees that they will dependably concur on the aftereffects of those figuring's. What's more, Byzantine assentation will in any case be ensured with likelihood one.

6.4.4 Hashgraph is resilient to Sybil Attack

The designer of Hashgraph, Leemon Baird, distributed a paper on Swirlds and Sybil Attacks. Hashgraph is principally intended for permissioned frameworks, where the danger of sybil assaults can be expelled. In his paper, Baird plots valuable agreement for a Swirld arrange. What's more, how Proof-of-stake idea as depicted in this part expels sybil assaults.

For instance, "One methodology is to begin with a consortium of, state, 10 substantial, regarded companies or associations that are the originators. Each is given a lot of StakeCoins to begin with, much the same as a permissioned blockchain, at first. In any case, that is just to kick it off. After some time, different individuals can join the record swirld. Also, other individuals can purchase StakeCoin". [58]

The general thought is that hashgraph + PoS ought to be impervious to sybil assaults.

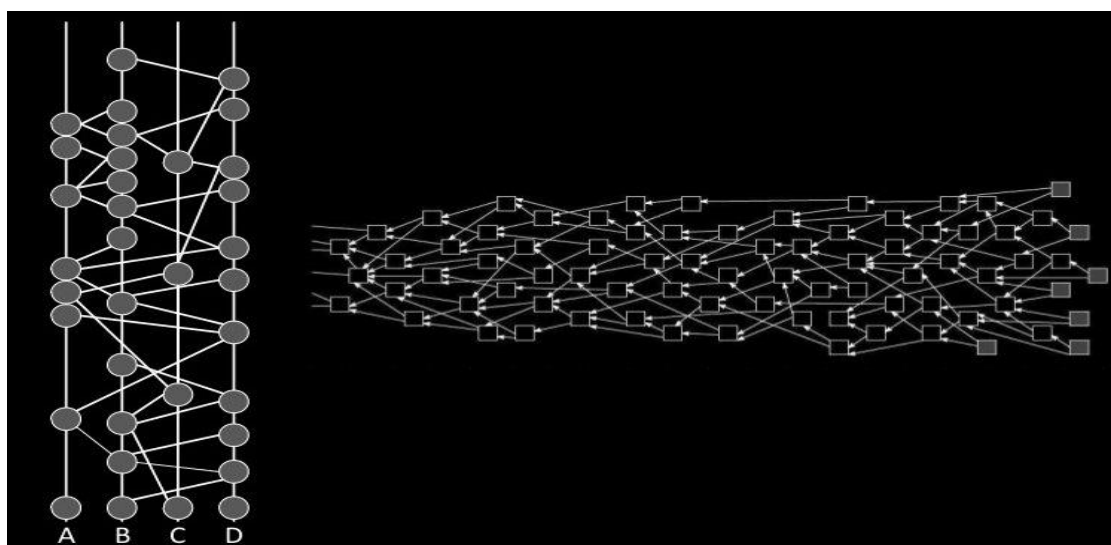


Figure 6.2 Sybil attack on hashgraph. [58]

6.5 Chapter Summary

This chapter presented the Security of Hashgraph. Each member of the Swirlds Hashgraph community has a unique identifier called Swirld ID. This ID is made up partly of a base-62 encoding and a sequence of words. This unique combination of ID is the first step of security to the Hashgraph. However, when members of the Hashgraph communicate and create events with crypto-hashes, they use the concept of Keyless Signature Infrastructure and each event is signed by the creator. This is another security layer to the hashgraph. As the transactions keep expanding the hashgraph is formed thus making it immutable, meaning it cannot be tampered. This a further security layer to the system.

This chapter also explains Byzantine Fault Tolerance theorem (BFT) and how this is a mathematical guarantee that Hashgraph is Byzantine and therefore resilient to security attacks.

The next chapter covers the evaluation of DDoS, Ping of Death & Sybil Attacks on Hashgraph.

CHAPTER 7

Evaluation of DDoS, Ping of Death & Sybil Attacks on Hashgraph

This chapter evaluates us with DDoS attack on Hashgraph. It also gives you explanation about Ping of death, & also about the Sybil attacks on Hashgraph. Section 7.1 covers the overview on DDoS attack. Section 7.2 tells us about the hardware and software specification. Section 7.3 covers the defence mechanism for DDoS attacks. Section 7.4 gives us implementation of Ping of death attack on hashgraphy. Section 7.5 covers implementation of DDoS attack Hashgraphy. Section 7.6 then summarizes the chapter.

7.1 Overview on DDoS Attack

In late 1990's prime comprehensive DDoS attack popped up against the University of Minnesota. 227 Zombies that prompted the attack (traded off PCs), close down the college's system for over two days [74]. DDoS assaults got further consideration In the year 2000, February, DDoS attacks started getting more when a programmer crushed many big online companies like Amazon, eBay , CNN interactive etc by performing DDoS attack, essentially backing them off or rendering their sites out of reach [75].

Specialists evaluated that Yahoo! was down for more than three hours, and the organization's loss of publicizing income and web based business was roughly \$500,000. The subsequent down time for Amazon.com cost them an expected \$600,000. Also there was significant drop in number of CNN clients which went down by five percent of the ordinary volume, while Buy.com went from 100% accessibility to 9.4% after these attacks [76]. Regardless of their effect, there have been no ways or procedures wherein we can control such attacks which have been ongoing from past around 10 years now. As indicated by an overall foundation security report in 2012, half of the review respondents (130 respondents altogether) demonstrated that their framework have encountered DDoS attack (e.g., switches, firewalls, and load balancers), and one-quarter experienced DDoS attacks against administrations utilized by their clients and accomplices amid the study time frame.[77].

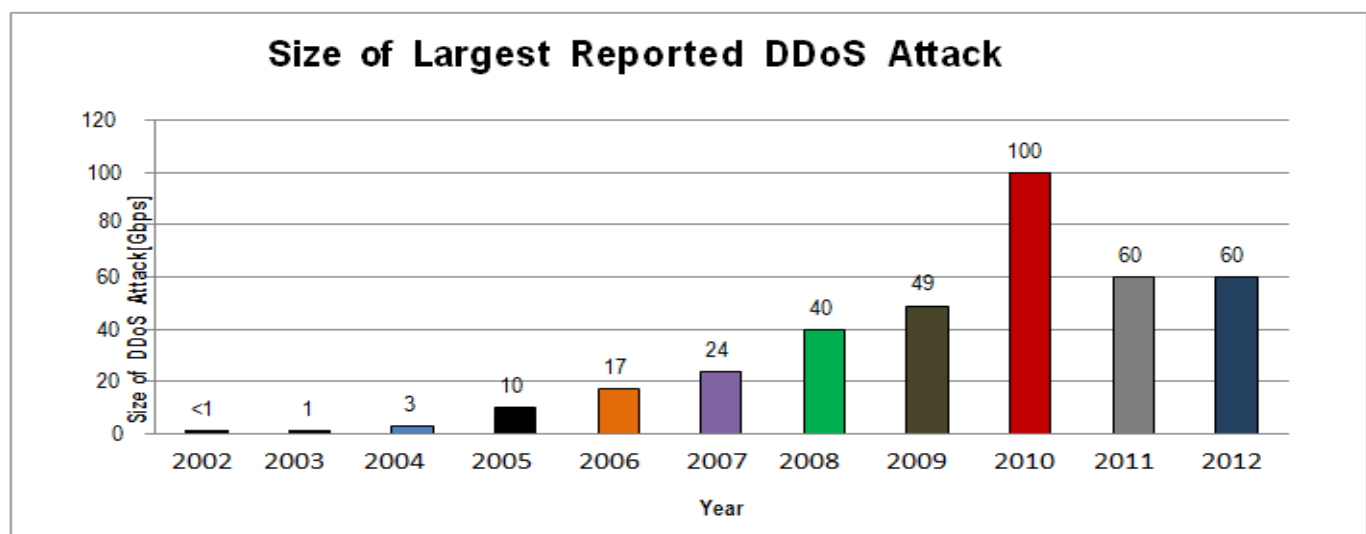


Figure 7.1: Size of Largest Reported DDoS Attack in Gbps from 2002 to 2012 [80]

DDoS attack from the time span of year 2002 to year 2012 have been illustrated in above figure 7.1[77]. It is very clear that the size of the attack was only about 500Mbps at the start. As the speed of the internet in the mid 2000's was restricted , there were relatively many procedures and framework available to be accessed over the internet.

Size of attacks if considered in 2012, expanded multiple times more than what was there in year 2002. It went up around 60Gbps in 2012 as compared to <1 in 2002. The most observable component of this diagram is that the biggest assault detailed was 100Gbps (in 2010). "This was an exceptionally critical capacity of traffic and has extra transmission capacity compared to Internet administrators had, not to mention their clients. This can be inferred that the assailants are moving to further developed danger approaches".[80]

Now being factual we know that the size of DDoS attack has been increasing day by day, we now have about 37% of the respondents (130 respondents altogether) had built up an expanded familiarity with the DDoS risk in their association, though, 63% held a similar dimension of mindfulness. In addition, the report uncovers that over 10% of the respondents did not have DDoS moderation capacities in their systems [77].

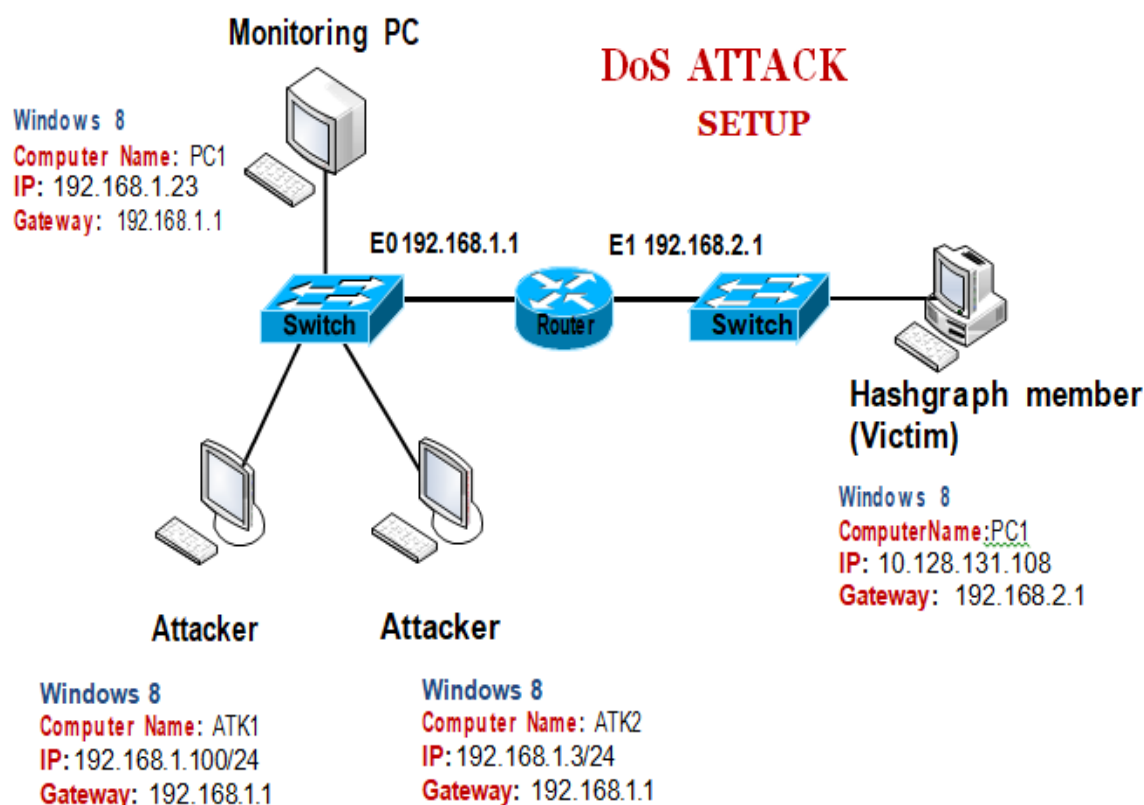


Figure 7.2. DoS/DDoS attack setup [78-80]

7.2 Hardware and Software Specification

7.2.1 Hardware

So as to be reliable and create precise information from this investigation, the equipment utilized in the majority of the examinations was kept indistinguishable. The equipment benchmark was contained an Intel® Core™ i5 2.80 GHz processor with 8.00 GB RAM for the effective activity of working frameworks; The assault was finished utilizing on the web apparatuses and Windows CMD. Table 5.1 blueprints the sort and details of the equipment included.

Table 7.1: Hardware Specifications

Hardware		Specifications
PC	CPU	Intel® Core™ i5 2.80 GHz
	RAM	8.00 GB
	Hard Disk	Western Digital Caviar SE 160 GB
	LAN Card	Intel® PRO/1000 GT Adapter
	Motherboard	Lenovo
	Motherboard Chipset	Intel Q965 Rev. 1

7.2.2 Software

In terms of software specification, Microsoft Windows 8.1 operating system was used. Table 5.2 describes the operating systems, roles, and software installed on the system. In addition, the details of software are also explained in Section 5.3.

Table 7.2: Software Specifications

Operating System	Role	Software installed/used
Windows 8	Victim	Nemesy for DDoS attack on Network, Monitor Network activities on Windows task Manager
Windows 8	Monitoring machine	Hashgraphy system developed using Eclipse IDE and Java SDK, Monitor Network activities on Windows Task manager
Windows 8	Attacker	Nemesy for DDoS Attack on Network, Windows CMD for ping of death attack on IP address
Windows 8	Attacker	Nemesy for DDoS Attack on Network, Windows CMD for ping of death attack on IP address

7.3 Defence Mechanisms for DDoS Attacks

DDoS attacks mainly fall under these three primary attacks namely: volumetric (Gbps), convention (pps) and application layer (rps) attacks as shown in Table 7.3. Every one of the three have the aim to upset a few or the majority of its unfortunate casualty's administrations, yet each performs it an alternate way.

Table 7.3 Type of DDoS attack and its various other genres along with mitigation technique.

DDOS ATTACK TYPE	METRIC	FAST FACT	CATEGORY	CHARACTERISTICS	EXAMPLES	MITIGATION
Volumetric Attack	Bits per second (bps), Giga bits per second (Gbps), flood	Was the first famous DDoS attack	Connectionless	High volume, using bots	Dyn, UDP flood	Volumetric attacks are absorbed in a global network of scrubbing centers that scale on demand to counter multi-gigabyte DDoS attacks.
Protocol attack	Packets per second (pps)	Traces its origins back to 1996	Connection-based	Attacks the network layer	Syn flood, ping of death	This type of attack is mitigated by blocking "bad" traffic before it reaches the site. Uses visitor identification technology to differentiate legitimate website visitors (humans, search engines) and automated or malicious clients.
Application layer attack	Requests per second (rps), low-rate	Made famous by Mirai malware	Connection-based	Difficult to detect	SQL injection, XSS	Application layer attacks are blocked by monitoring visitor behavior, blocking known bad bots, and challenging suspicious or unrecognized entities with JS test, cookie challenge, and even CAPTCHAs.

A definite investigation on DDoS assaults and their guard components have been completed. Subsequent to utilizing the resistances, as expressed over, the outcomes demonstrated that the execution of Windows was expanded. The Hybrid Method and Threshold Limit were the best protections against a DDoS assault in the vast majority of the examinations, though the Software Firewall and Network Load Balancing were the least viable resistances. The Hybrid Method and Threshold Limit could build the TCP throughput and UDP throughput. The Hybrid Method and Threshold Limit could altogether lessen the CPU use from 25% (amid the assault) to 2%, while IP Verify, ACLs, Network Load Balancing, and the Software Firewall just decently decreased the CPU use somewhere in the range of 8% and 21%. [59]

7.4 Implementation of Ping of Death Attack on Hashgraphy system

One type of Denial of Service (DoS) attack happens when an aggressor can surge a legitimate hub on a system with good for nothing messages, keeping that hub from executing substantial obligations & jobs. Distributed Denial of Service (DDoS) utilizes open administrations for gadgets to accidentally enhance the DoS attack - making them a considerably more prominent risk.

In a DLT, a DDoS attack could concentrate on the centre points that add to the significance of accord and, possibly, shield that understanding from being set up.

The hashgraph is DDoS immune as it empowers no single centre or unassuming number of centres with phenomenal rights or obligations in developing an understanding. Both Bitcoin and hashgraph are passed on with the end goal that restricts DDoS attacks. An aggressor may flood one section or miner with bundles, to unexpectedly isolate them from the web. In any case, the system all things considered will continue working normally. A assault on the system in general would require flooding an immense piece of the people with packs, which is logically troublesome. There have been different proposed choices to blockchain reliant on pioneers or round robin. These have been proposed to avoid the affirmation-of-work costs of Bitcoin. Be that as it may, they have the disadvantage of being delicate to DDoS assaults. In the event that the assailant assaults the present head, and changes to assaulting the new pioneer when one is picked, at that point the aggressor can solidify the whole framework while as yet assaulting just a single PC at any given moment. Hashgraph maintains a strategic distance from this issue, while still not requiring verification-of-work.

As we have just found in the hashgraphy usage that every part has an IP address and port related with it.

- Setting up IP addresses for all individuals:

Since, it is a mimicked situation execution, of course every one of the hubs will have a similar IP of the framework. To do the ping assault I will arrange a particular IP address for each hub. For which reason, I have pursued the underneath steps:

1. Snap Start Menu > Control Panel > Network and Sharing Centre. (For Windows 8 and higher, scan for and open Control Panel and select Network and Internet).
2. Snap Change connector settings.
3. Right-tap on LAN and tap on Properties.
4. Select Internet Protocol Version 4 (TCP/IPv4) and tap on Properties.

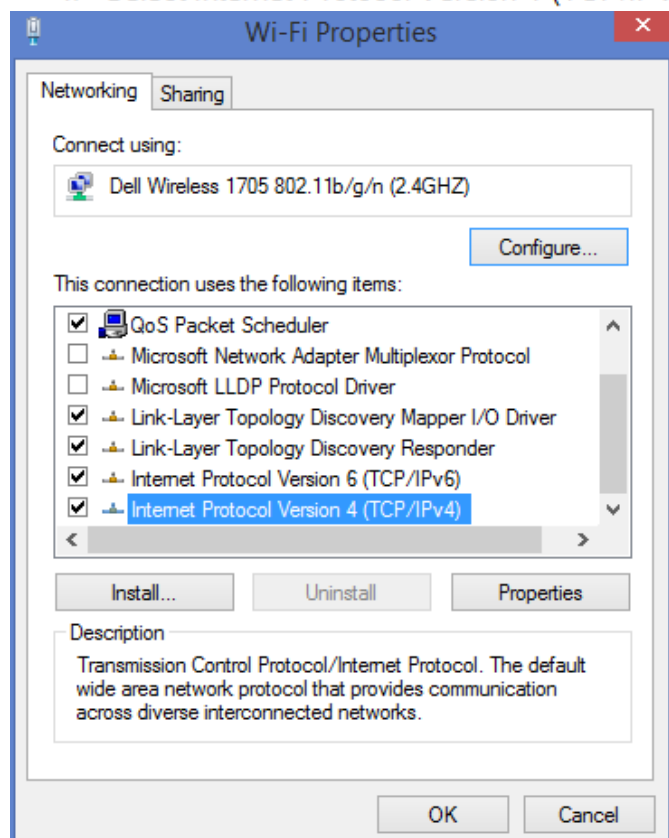


Figure 7.3 Wi-Fi Properties screen

5. Select "Utilize the accompanying IP address" and enter the IP address, Subnet Mask, Default Gateway. what's more, DNS server. Snap OK and cancel out the LAN Connection properties casement.

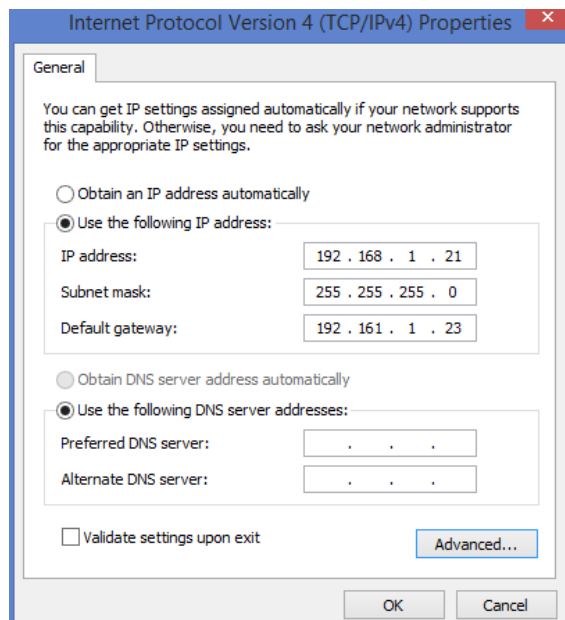


Figure 7.4 IPv4 Properties screen

6. Similarly, I have included various IP address for each particular hub.

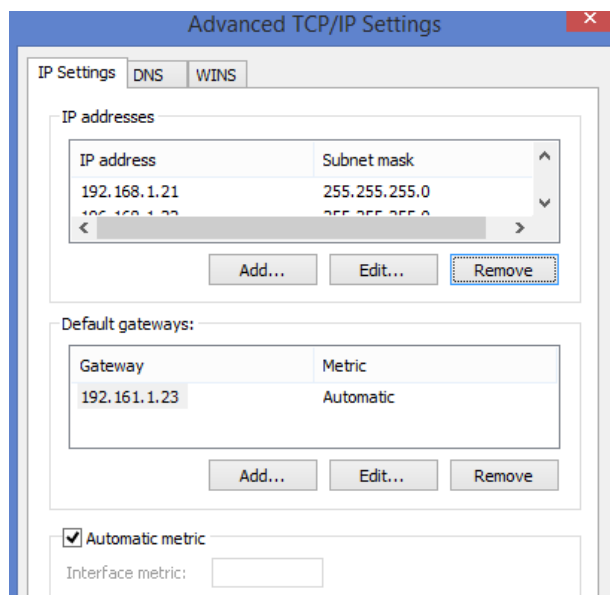


Figure 7.5 Advanced TCP/IP settings box

Aek: 10.128.131.108

Ben: 192.168.1.22

Cate: 10.128.131.107

Dev: 192.168.1.21

Targeting Aek's IP:

- To demonstrate this attack, 4 PCs have been set-up that are on a similar system. What's more, I utilized them to assault the unfortunate casualties IP.DOS attacks are unlawful on systems that you are not approved to do as such. This is the reason we should setup our very own system for this.
- In this case, Aek is the injured individual for the ping attack. Aek's IP is 10.128.131.108.
- We will ping our injured individual PC with unbounded information parcels of more than 65000 approximately (Say 65500)

Enter the accompanying direction

Ping 10.128.131.108 -t |65500

Here,

- "Ping" sends the information parcels to the person in question.
- "10.128.131.108" is the IP address of the person in question.
- "- t" signifies the information bundles ought to be sent until the point that the program is ceased.
- "- l" determines the information load to be sent to the person in question.

You will get results like the ones appeared as follows

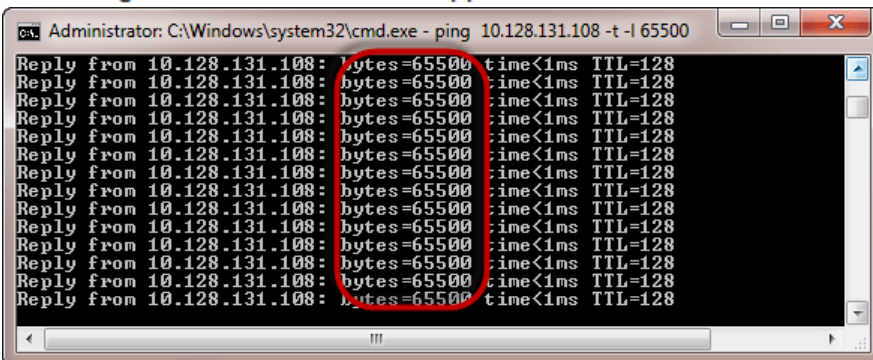


Figure 7.6 Ping of death result window

As we see flooding the objective PC's IP with information parcels doesn't have significant impact on the person in question. Furthermore, henceforth, it doesn't have any significant effect on the Hashgraph all in all. Thusly, as per the Hashgraphy white paper we can say that the Hashgraph framework is strong to ping attack, which is a kind of DDoS attack.

On the off chance that you need to see the impacts of the assault on the objective PC, you can open the assignment director and view the system exercises.

- Right click on the taskbar
- Select start task manager
- Click on the network tab
- You will get results similar to the following

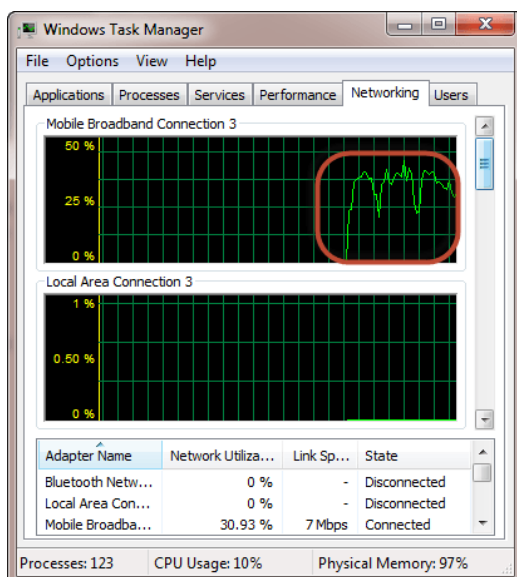


Figure 7.7 System Task management

In the event that the attacks were effective, you would have the capacity to see a lot of activity happening on the system. Nonetheless, for this situation the system activities appear to be typically normal and Hashgraph framework proves to keep the system secure.

Not all PCs can deal with information bigger than a certain size. Along these lines, when a ping of death parcel is sent from a source PC to an objective machine, the ping bundle gets divided into littler parts.

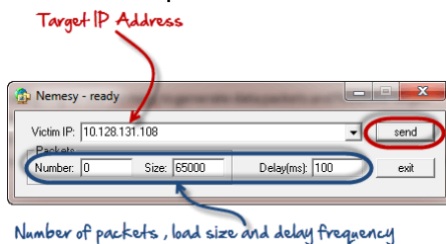
One part is of 8 octets estimate. At the point when these bundles achieve the objective PC, they touch base in parts. Along these lines, the objective PC reassembles the parcels which are acquired in pieces. Be that as it may, the entire assembled bundle causes support flood at the objective PC.

This support issue regularly causes the framework crash making the framework increasingly helpless against attacks.

When the framework turns out to be progressively powerless against attacks, it permits more assaults like the infusion of a trojan horse on the objective machine.

7.5 Implementation of DDoS Attack on Hashgraphy System

- To dispatch a DDoS attack ,Nemesy which is an online DDoS attack instrument has been utilized to produce information parcels and surge the objective PC. Nemesy will be identified as an unlawful program by your enemy of infection henceforth against infection must be incapacitated for this attack. [60]
- As per figure 7.7, Enter the objective IP address of Aek.
- 0 as the quantity of bundles implies interminability. You can set it to the ideal number on the off chance that you would prefer not to send, vastness information bundles.
- The size field determines the information bytes to be sent and the postponement indicates the time interim in milliseconds.
- Tap send catch



- Following results should be evident.

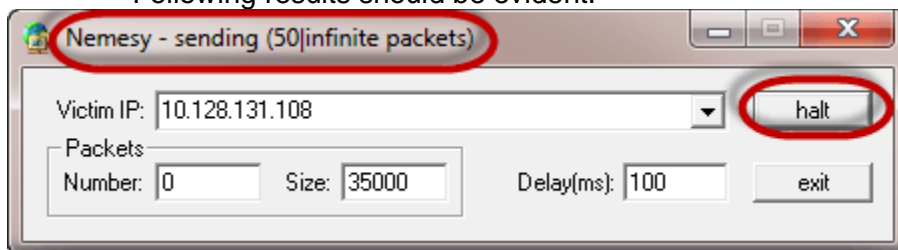


Figure 7.7 DDoS attack demonstration

- Head bar will demonstrate the amount of parcels sent.
- End catch can be utilized to prevent the program from sending information bundles.
- Screen the errand supervisor of the objective PC to see the system exercises.
- Because of the idea of the DDoS attack, there has been a slight back off in the execution of system exercises of Aek. Anyway this has not influenced the general agreement of the Hashgraph and it is as yet running consistent as delineated beneath.

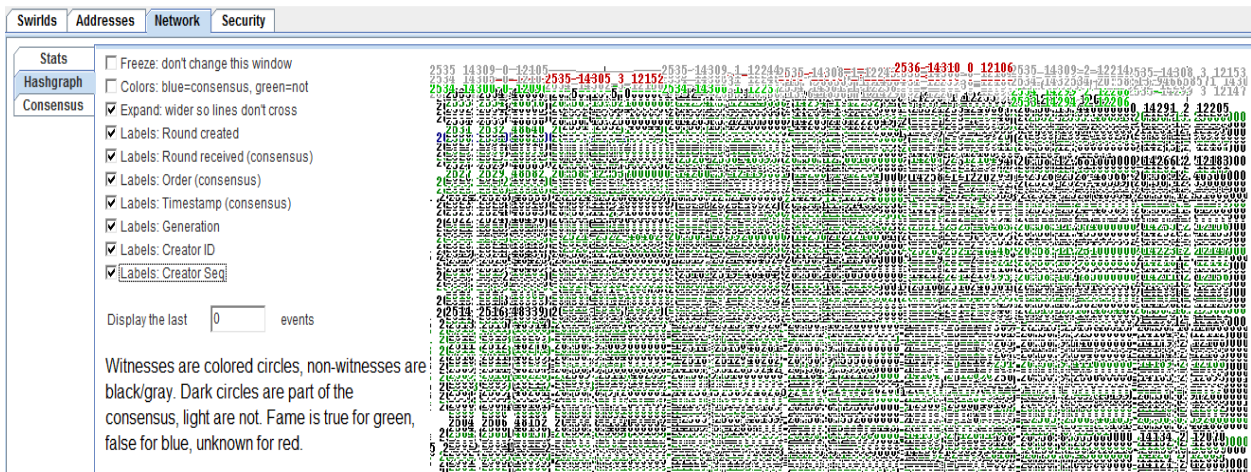


Figure 7.8 DDoS attack effect on Aek's performance

7.6 Theoretical Evaluation of Hashgraph resilient to Sybil Attack

How does the Swirlds stage maintain a strategic distance from Sybil attacks, where groups of attacks on dummy accounts from one assailant can control the framework? The short answer is that it utilizes verification of-stake inside, however is adaptable remotely. So it can work as verification of-stake or evidence of-work. It can work as permissioned or not permissioned. It can work in numerous different modes, too. In any case, inside, it is evidence of-stake. A Sybil attack in straightforward terms is, "A vicious mechanism misguidedly goes up against numerous personalities nodes. The extra characters are called Sybil nodes."; Sybil attack causes excessively undermine to steering calculation, information accumulation, reasonable asset portion, casting a ballot framework, rowdiness location. Thus, distinguishing and keeping this type of attack is significant for security of the remote sensor arrange. In spite of the fact that, the Swirlds whitepaper demonstrates that Hashgraph is sheltered from Sybil Attacks. There are a few elective approaches to recognize and avert Sybil attacks as appeared in these whitepapers. [68-70]

7.6 Chapter Summary

This chapter presented an Overview of DDoS attacks. There is a practical demonstration of DDoS, Ping of death attacks. And a theoretical proof of Sybil attacks defence. Also Byzantine fault Tolerance(BFT) theorem as explained in detail in the previous chapter makes Hashgraph resilient to attacks. And stands true because in this demonstration Hashgraph has been resilient to DDoS and ping of death attacks. Although the attack has been setup to attack members of the community, it has in no way been successful to cease the community from outreaching consensus. The system continued to work as normal.

The results showed that the performance of Hashgraph was not reduced during a ping of death and DDoS attack. However, this thesis references to practical demonstrations for DDoS/DoS attack for Windows 8 Operating System's computer's performance that can be evaluated with six defences, i.e. Control (Access) Lists, verge limit, Hybrid Defence, IP Verification, Load Balancing for network, and DDoS Software Firewall. [80]

The next chapter demonstrates us with the Real-time tracking application on Hashgraphy.

CHAPTER 8

Application of Hashgraph to Real time tracking application

In this chapter a real-time tracking has been demonstrated. The below section gives its flowchart and also explains it step-wise on how the application works.

A real-time tracking application has been developed and Hashgraphy algorithm has been applied to it to measure speed, efficiency, safety of the application. A vehicles of this tracking application are the main members of the Hashgraph. The hashgraphy algorithm will be used in tracking multiple vehicles and monitoring their precise location using Global Positioning System (GPS) device and gossip some real time information like weather, traffic, delivery details and so on.

A detailed description of the tracking application and how it works has been explained below

8.1 Flow-chart and architecture of the tracking application

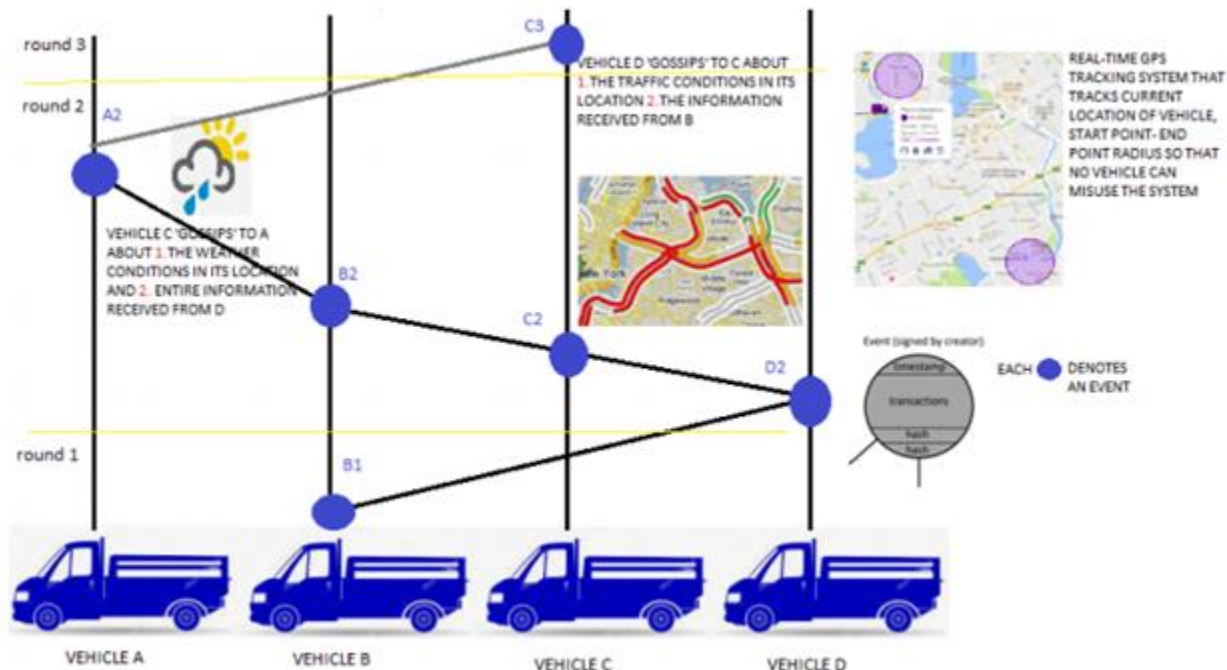


Figure 8.1 Flowchart of tracking application.

According to the gossip convention of Hashgraphy, a vehicle 'B1' as appeared in Figure 8.1 circulates data haphazardly to vehicle 'D2' pretty much all the data it knows and furthermore all data it acquired from alternate individuals in more established occasions, D2 now chats with C2 and reveals to it about the traffic conditions in its location(self-data) and furthermore all data it acquired from B1. This occasion contains a timestamp, data, and two hashes - one self-created and one from the other hub. Occasions along these lines build a Hashgraph, and this is added to history. In this way, every vehicle approaches each data that has been shared inside this application arrange. There are distinctive rounds made for a lot of occasions. In this precedent we have cycle 1, all around 3. In cycle 2, the vehicles get data from round 1(vehicle B1), since history is kept up in hashgraph every one of the individuals can perceive how the individuals beneath have imparted. With the deep rooted system of virtually casting a ballot, the vehicle can foresee on how another vehicle will get cast a ballot. There are rounds made for each arrangement of occasions here. There is no chance to get of any vehicle to pass false data in light of the fact that everything is put away ever, consequently producing a proof-of-stake with minimal effort. After some time, vehicles constantly

refresh the chart with ongoing data they will get. The Hashgraph runs the following application on the PCs/gadgets of each part who is a piece of that mutual world (a "swirl"). Also, as we probably aware that the network of individuals is an "organize" of "full hubs" (or of "vehicles"). The hashgraph consensus calculation guarantees that this following application sees similar exchanges in a similar request. The application is then in charge of refreshing the state as per the guidelines of the application. For instance, in this following application, an "exchange" is an explanation that X reports on ongoing data ought to be exchanged from vehicle Y to wallet vehicle. Application inspects if vehicle Y have got some constant updates. In the event that it does, the application plays out the exchange, by refreshing its nearby record of what amount is in Y and what amount is in Z. On the off chance that Y hasn't obtained numerous updates, application can't do anything, since they realise that exchange is incapacitated. Because everybody is operating equivalent application, & because everybody goes up with similar exchanges in a similar request, at that point everybody will finish up with a similar state. They will all concur precisely what numbers of updates presented in Y after initial 100 exchanges. It will also concur about which exchanges were substantial and are incapacitated. Thus, everyone will acknowledge that state. Also, check if marked state is duplicated, permanent record. Hashgraph assists with quicker conveyance, cost the executives, higher security of this following application. The system secures assets and increment their productivity. A standout amongst the most generally appropriate parts of hashgraphy is that it empowers increasingly secure, straightforward observing of exchanges. In this manner, exchanges should be reported for lasting decentralized history — diminishing time obstruct, included expenses, & manual blunders.

8.2 Progress Reports during tracking application build

This part of the chapter explains overall tracking application progress reports on Jira (a Project management tool). Since a combination of Agile + Quantitative Research methodology have been used to build this thesis, Jira has been helpful to track the progress of the tracking application build specifically on the Agile aspect.

a. The Cumulative Flow Diagram - Shows the completed work(done) and the remaining number of issues(bottlenecks) during the overall project duration. This encourages to distinguish possible bottlenecks that should be examined, that are causing hinderance to complete the project. [61]

Cumulative Flow Diagram

24/Jul/18 to 16/Jan/19 (All Time) Refine report

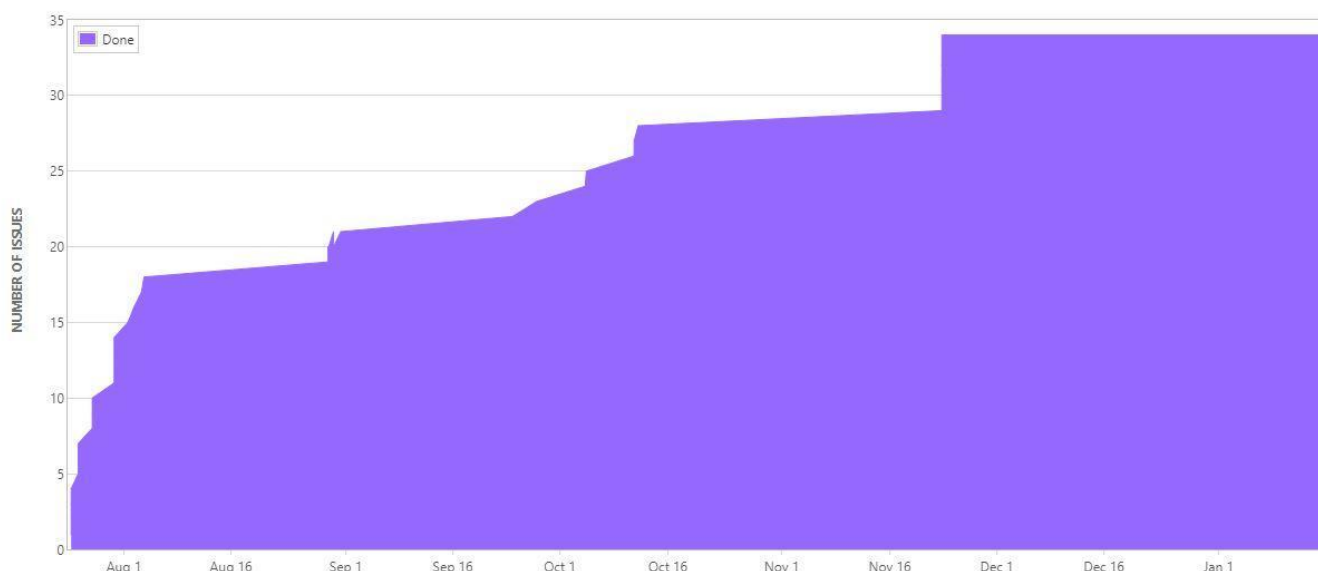


Figure 8.2. Cumulative flow diagram.

b. Epic Report – Helps to understand and track advancement towards finishing an epic over an estimated period of time. It also tracks the issues we encounter in a planned/estimated work and unestimated work. An epic constitutes main features of the application that are targeted to be completed as per estimation-known as story points. Story points are usually in terms of hours. Each epic further contains subtasks. And each subtask or a group of subtasks are targeted to get completed in a certain time period- known as sprint.

Epic Report

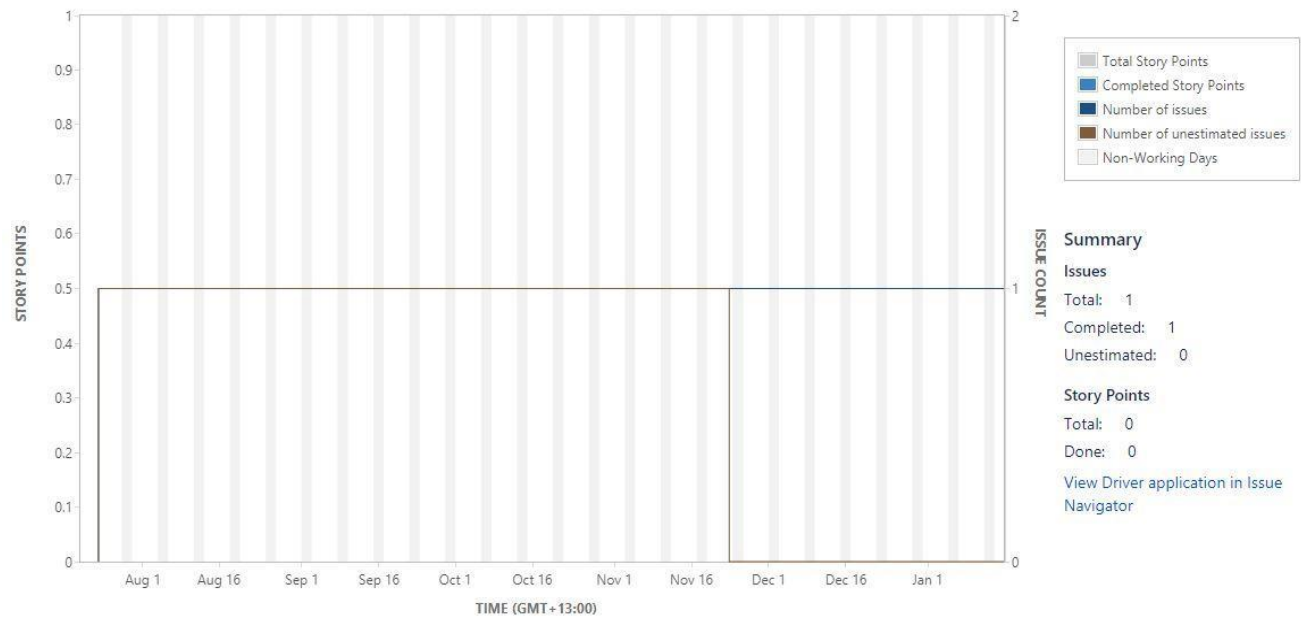


Figure 8.3 Epic report.

c. Sprint Report - Understand the work finished or pushed back to the accumulation in each run. This causes you decide whether your work is overcommitted or completed on time.



Figure 8.4 Sprint report

8.3 Backend Database structure of the tracking application

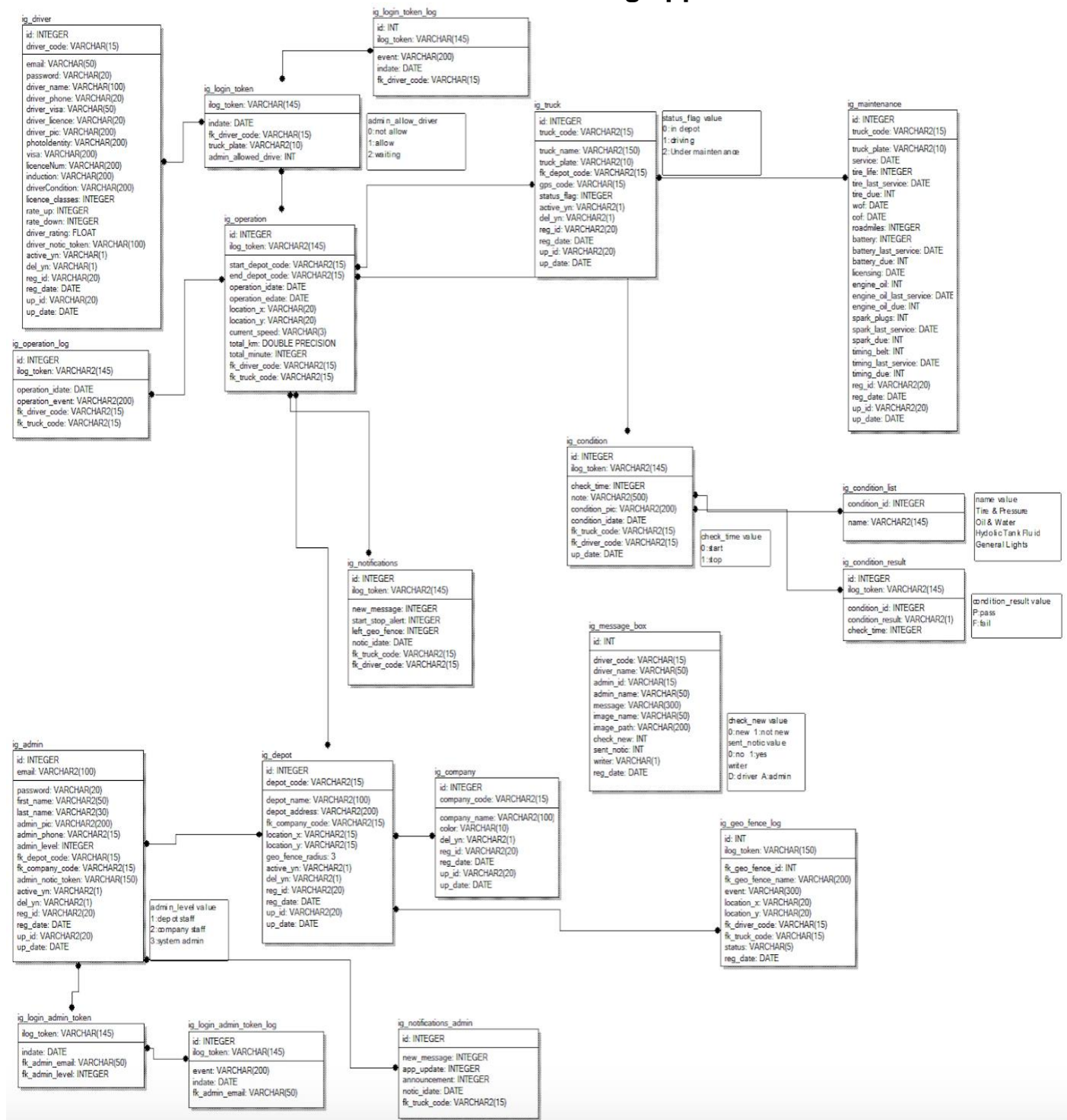


Figure 8.5. Entity Relationship Diagram

List of main Tables in the Database:

Login table: Contains all login information for Truck Driver, Truck company Owner, Admin.

Truck table: Contains all truck details and relates to other tables such as Truck maintenance, Truck conditions, truck model, etc.

Driver table: Contains Driver details like name, phone number, license number, photoID etc.

Geo-tracking: Contains details of truck location, truck speed etc.

Message box: Contains details of communication between driver and company owner.

8.4 Working of Tracking Application

1. List of All Vehicles

The screenshot shows the 'List All Vehicles' page of a tracking application. The sidebar on the left contains navigation links: TRACKING (highlighted), GEO-FENCE, INBOX (5), TRIPS, EVENTS LOG, and MANAGE. The main content area has tabs for 'List All Vehicles', 'Map View', and 'Leaderboard'. Below the tabs are buttons for '+ Add Vehicle' and '+ Add Driver', and a search bar. The table below lists 20 vehicles with their details and action icons.

NAME	VEHICLE #	DRIVEN	LOCATION	SPEED	ACTION
Owen Valdez	HHF211	0km	428 Dickens Road	21	[Icons]
Jay Hodges	ODG834	0km	5331 Dayna Mountain Apt. 535	0	[Icons]
Clarence Tate	LOGCR	10km	762 Dickinson Villages Apt. 048	31	[Icons]
Hunter Garner	SHF231	25km	957 Emmie Burg Suite 886	98	[Icons]
Juan Boyd	ODF321	50km	092 Considine Row	53	[Icons]
Norman Woods	MKN34	200km	568 Susan Locks	58	[Icons]
Nicholas Townsend	KHG363	40km	038 Beer Hollow Apt. 096	93	[Icons]
Miguel Dunn	ODG834	0km	9450 Terry Springs Apt. 515	21	[Icons]
Cole Nunez	LOGCR	10km	091 Ferry Gardens Apt. 853	95	[Icons]
Earl Ferguson	SHF231	25km	008 Alessandra Corners Apt. 586	46	[Icons]
Mary Griffith	ODF321	50km	85 West Stream	78	[Icons]
Warren Phillips	MKN34	200km	808 Bridget Falls	77	[Icons]
Teresa Morales	KHG363	40km	29 McDermott Lakes	46	[Icons]
Miguel Craig	ODG834	0km	484 Jewess Villages	70	[Icons]
Barbara Castro	LOGCR	10km	640 Briana Shore	17	[Icons]
Georgie Mack	SHF231	25km	330 Zieme Wall Suite 144	36	[Icons]
Eric Clark	ODF321	50km	506 Hattie Ramp	57	[Icons]
Andrew Johnson	MKN34	200km	320 Violet Islands	59	[Icons]

At the bottom right, there are pagination controls: First, 10, 11, 25, 26, Last.

Figure 8.6 List of all vehicles

- Full list of vehicles and drivers can be viewed.
- We can also add a new vehicle/driver to the application.
- Only xAdmin can perform following Actions. [61]



Figure 8.7 Actions List

As shown in Figure 8.7, implies as follows:

1. View/monitor all Drivers Information.
2. View/ monitor all Truck information.
3. Chat with driver or company owner/ view communication.
4. Event logging.
5. Live Tracking of truck movement, speed, current location, and start-end locations.

2. Map View:

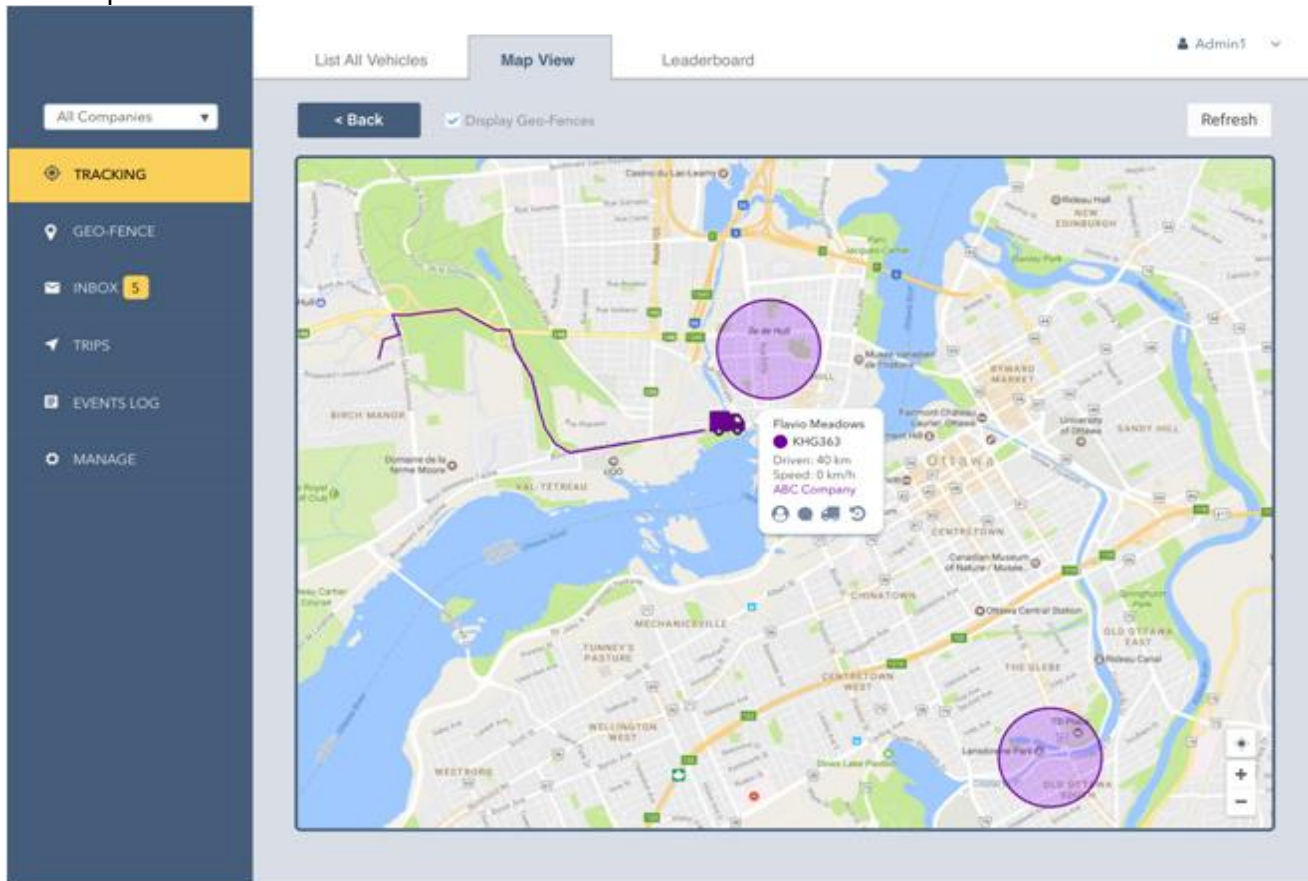


Figure 8.8 Map View

- It displays all the truck's location in the map.
- The two circles indicate the start and end location radius of a trip.
- Clicking on truck's icon gives detailed information about the Truck.

3. Leader board:

The screenshot shows the 'Leaderboard' tab selected in the top navigation bar. The left sidebar is identical to the Map View. The main area displays a table of driver performance metrics. The table has columns for NAME, VEHICLE #, DRIVEN, LOCATION, SCORE, and ACTION. The data is sorted by score in descending order. At the bottom, there are pagination controls showing 'First', '10', '11', '25', '26', and 'Last'.

NAME	VEHICLE #	DRIVEN	LOCATION	SCORE	ACTION
Owen Valdez	HHF211	0km	428 Dickens Road	21	[Icons]
Jay Hodges	ODG834	0km	5331 Dayna Mountain Apt. 535	0	[Icons]
Clarence Tate	ILOGCR	10km	762 Dickinson Villages Apt. 048	31	[Icons]
Hunter Garner	SHF231	25km	957 Emmie Burg Suite 886	98	[Icons]
Juan Boyd	COF321	50km	092 Considine Row	53	[Icons]
Norman Woods	MKN34	200km	568 Susan Locks	58	[Icons]
Nicholas Townsend	KHG363	40km	038 Beer Hollow Apt. 096	93	[Icons]
Miguel Dunn	ODG834	0km	9450 Terry Springs Apt. 515	21	[Icons]
Cole Nunez	ILOGCR	10km	091 Ferry Gardens Apt. 853	95	[Icons]
Earl Ferguson	SHF231	25km	008 Alessandra Corners Apt. 586	46	[Icons]
Mary Griffith	COF321	50km	85 West Stream	78	[Icons]
Warren Phillips	MKN34	200km	808 Bridget Falls	77	[Icons]
Teresa Morales	KHG363	40km	29 McDermott Lakes	46	[Icons]
Miguel Craig	ODG834	0km	484 Jewess Villages	70	[Icons]
Barbara Castro	ILOGCR	10km	640 Briana Shore	17	[Icons]
Georgie Mack	SHF231	25km	330 Zieme Wall Suite 144	36	[Icons]
Eric Clark	COF321	50km	506 Hattie Ramp	57	[Icons]
Andrew Johnson	MKN34	200km	320 Violet Islands	59	[Icons]

Figure 8.9 Leader Board

- Displays ratings for each driver- Performance indicator

- Basically acts like a score card/points system to motivate them to lead the rating board.

4. GPS Location Radius:

The screenshot shows the 'GEO-FENCE' interface with a sidebar on the left containing 'All Companies', 'TRACKING', 'GEO-FENCE' (highlighted), 'INBOX 5', 'TRIPS', 'EVENTS LOG', and 'MANAGE'. The main area displays three groups of vehicles based on their location radius:

Alvaton Coal Mine (2)

NAME	VEHICLE #	DRIVEN	LOCATION	SPEED	ACTION
Owen Valdez	HHF211	0km	428 Dickens Road	21	[Icons]
Jay Hodges	OOG834	0km	5331 Dayna Mountain Apt. 535	0	[Icons]

Rosannaborough Warehouse (1)

NAME	VEHICLE #	DRIVEN	LOCATION	SPEED	ACTION
Owen Valdez	HHF211	0km	428 Dickens Road	21	[Icons]
Jay Hodges	OOG834	0km	5331 Dayna Mountain Apt. 535	0	[Icons]

Other (20)

NAME	VEHICLE #	DRIVEN	LOCATION	SPEED	ACTION
Owen Valdez	HHF211	0km	428 Dickens Road	21	[Icons]
Jay Hodges	OOG834	0km	5331 Dayna Mountain Apt. 535	0	[Icons]
Clarence Tate	ILOGCR	10km	762 Dickinson Villages Apt. 048	31	[Icons]
Hunter Garner	SHF231	25km	957 Emmie Burg Suite 886	98	[Icons]
Juan Boyd	OOF321	50km	092 Considine Row	53	[Icons]
Norman Woods	MKN34	200km	568 Susan Locks	58	[Icons]
Nicholas Townsend	KHG363	40km	038 Beer Hollow Apt. 096	93	[Icons]
Miguel Dunn	OOG834	0km	9450 Terry Springs Apt. 515	21	[Icons]
Cole Nunez	ILOGCR	10km	091 Ferry Gardens Apt. 853	95	[Icons]
Earl Ferguson	SHF231	25km	008 Alessandra Corners Apt. 586	46	[Icons]
Mary Griffith	OOF321	50km	85 West Stream	78	[Icons]
Warren Phillips	MKN34	200km	808 Bridget Falls	77	[Icons]
Teresa Morales	KHG363	40km	29 McDermott Lakes	46	[Icons]
Miguel Craig	OOG834	0km	484 Jewess Villages	70	[Icons]
Barbara Castro	ILOGCR	10km	640 Briana Shore	17	[Icons]
Georgie Mack	SHF231	25km	330 Zieme Wall Suite 144	36	[Icons]
Eric Clark	OOF321	50km	506 Hattie Ramp	57	[Icons]
Andrew Johnson	MKN34	200km	320 Violet Islands	59	[Icons]

At the bottom right, there is a pagination control showing 'First', '10', '11', '25', '26', and 'Last'.

Figure 8.10 GPS Location Radius

- List up all the information grouped by Geo-location radius. For example, two vehicles are in the geo-location radius of Rosanna borough Warehouse.

5. Real-time chatting feature:

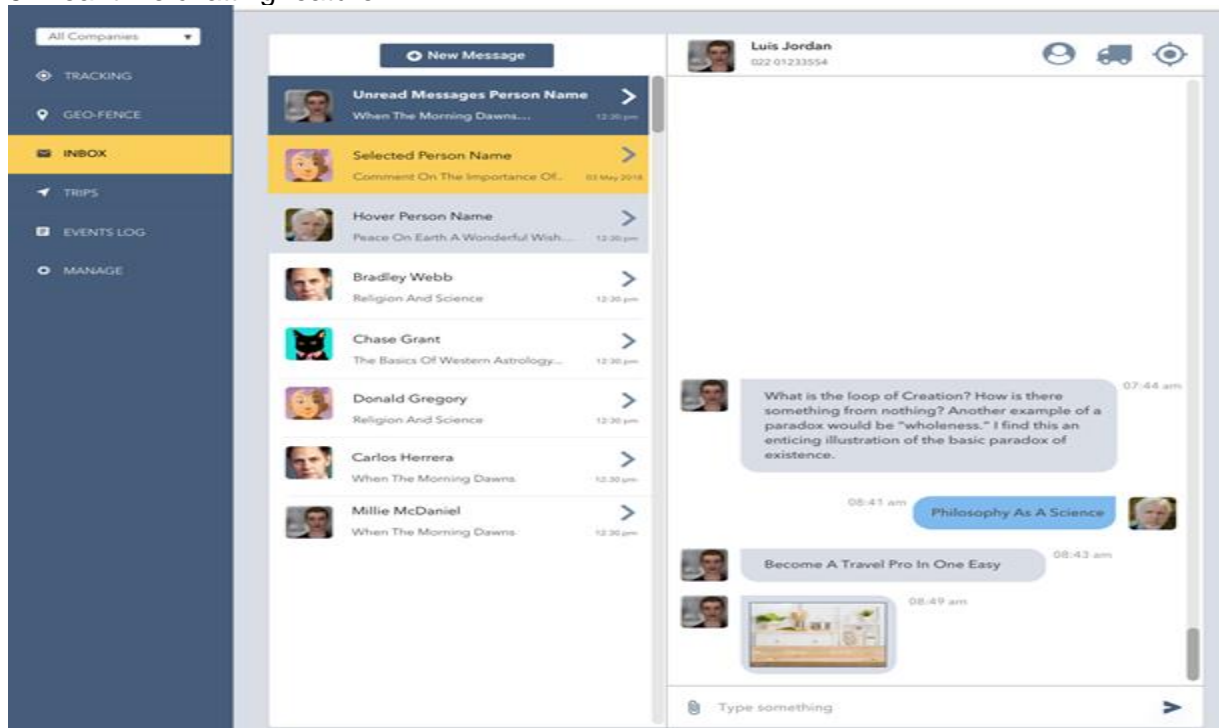


Figure 8.11 Real-time chatting Feature

- Instant messaging between drivers. To communicate about latest traffic updates, weather conditions etc.

- System monitors speed of the vehicles and sends message to driver if they are over-speeding.

6. Trips information:

TRIPS						
<div> <div>All Companies</div> <div> <div>TRACKING</div> <div>GEO-FENCE</div> <div>INBOX 5</div> <div>TRIPS</div> <div>EVENTS LOG</div> <div>MANAGE</div> </div> </div>						
<div> <div> <div>+</div> <div>Add a Trip</div> </div> </div>						
<div> <div>Alvaton Coal Mine to Rosannaborough Warehouse (2)</div> <div>Edit Trip</div> <div>Add a Vehicle to Trip</div> </div>						
NAME	VEHICLE #	DRIVEN	LOCATION	SPEED	ACTION	
Owen Valdez	HHF211	0km	428 Dickens Road	21		
Jay Hodges	ODG834	0km	5331 Dayna Mountain Apt. 535	0		
<div> <div>Rosannaborough Warehouse to Alvaton Coal Mine (2)</div> <div>Edit Trip</div> <div>Add a Vehicle to Trip</div> </div>						
NAME	VEHICLE #	DRIVEN	LOCATION	SPEED	ACTION	
Owen Valdez	HHF211	0km	428 Dickens Road	21		
Jay Hodges	ODG834	0km	5331 Dayna Mountain Apt. 535	0		
<div> <div>Warehouse1 permanent (4)</div> <div>Edit Trip</div> <div>Add a Vehicle to Trip</div> </div>						
NAME	VEHICLE #	DRIVEN	LOCATION	SPEED	ACTION	
Owen Valdez	HHF211	0km	428 Dickens Road	21		
Jay Hodges	ODG834	0km	5331 Dayna Mountain Apt. 535	0		
Clarence Tate	ILOGCR	10km	762 Dickinson Villages Apt. 048	31		
Hunter Garner	SHF231	25km	957 Emmie Burg Suite 886	98		

Figure 8.12 Trips information

- Display all trips information , including vehicle number, driver details, geo-location radius, speed etc.

7. Event Log:

EVENTS LOG

All Companies

TRACKING

GEO-FENCE

INBOX 5































































































































TRIPS

EVENTS LOG

MANAGE

Admin1

Search

NAME	VEHICLE #	TIME	EVENT	ACTION
 Owen Valdez	 HHF211	30/08/1881 11:52	Has left Coal Mine.	    
 Jay Hodges	 OOG834	13/06/1839 15:47	Arrived to Warehouse 1	    
 Clarence Tate	 ILOGCR	22/08/1839 04:08	Arrived to Warehouse 1	    
 Hunter Garner	 SHF231	06/08/1949 01:19	Has left Coal Mine.	    
 Juan Boyd	 COF321	25/02/1978 06:36	Arrived to Coal Mine	    
 Norman Woods	 MOKN34	03/04/1885 12:50	Alert: Failed start up check: oil	    
 Nicholas Townsend	 KHG363	28/12/1934 07:29	Arrived to Warehouse 1	    
 Miguel Dunn	 OOG834	16/06/1985 21:33	Has left Coal Mine.	    
 Cole Nunez	 ILOGCR	25/06/1899 18:12	Arrived to Coal Mine	    
 Earl Ferguson	 SHF231	15/11/2017 21:08	Alert: Left Warehouse 2 (permanent geo-fence)	    
 Mary Griffith	 COF321	16/04/1960 02:11	Arrived to Warehouse 1	    
 Warren Phillips	 MOKN34	26/11/1962 16:07	Left Coal Mine.	    
 Teresa Morales	 KHG363	04/07/1834 21:52	Arrived to Coal Mine	    
 Miguel Craig	 OOG834	23/11/1960 03:02	Alert: Failed start up check: engine	    
 Barbara Castro	 ILOGCR	17/05/1849 21:59	Has left Coal Mine.	    
 Georgie Mack	 SHF231	17/05/1995 09:14	Arrived to Warehouse 1	    
 Eric Clark	 COF321	14/08/1872 03:57	Arrived to Warehouse 1	    
 Andrew Johnson	 MOKN34	01/11/1958 06:20	Alert: Left Warehouse 2 (permanent geo-fence) Second line Third Fourth	    

First

◀

10

11

...

25

26


▶

Last

Figure 8.13 Event Log

- Logs all events such as Driver starting a trip, start and end point of the trip, Vehicle failure, Driver reaching end point of trip.

8. Vehicle Profile:

Vehicle: KGH638 






Log Book	<input type="button" value="View Log Book"/>
Last Driver	Alexa Smith  <input type="button" value="View Profile"/>
Service due	<input type="button" value="Edit"/> 31 Dec 2018
WoF due	<input type="button" value="Edit"/> 12 Apr 2017
COF due	<input type="button" value="Edit"/> 01 Dec 2018
Roadmiles	<input type="button" value="Edit"/> 342,303km
Number Plate	<input type="button" value="Edit"/> KGH638
Tyre Life/Rotation <small>Last Service 01 Jan 2017</small>	<input type="button" value="Edit"/> 7% <small>Due: 3212km</small>
Battery <small>Last Service 01 Jan 2017</small>	<input type="button" value="Edit"/> 50% <small>Due: 3212km</small>
Engine Oil <small>Last Service 01 Jan 2017</small>	<input type="button" value="Edit"/> 35% <small>Due: 3212km</small>
Spark Plugs <small>Last Service 01 Jan 2017</small>	<input type="button" value="Edit"/> 54% <small>Due: 3212km</small>
Timing Belt <small>Last Service 01 Jan 2017</small>	<input type="button" value="Edit"/> 98% <small>Due: 3212km</small>

Figure 8.14 Vehicle Profile

- Displays WOF, Number plate information, Distance travelled etc.

9. Driver Profile:

Profile: Alex Smith 

 Ph: 022 01233554
  

Total Driven
242,342km
Total Hours
2,422hrs

Last Vehicle	KGH234	<input type="button" value="View Info"/>
Log Book	<input type="button" value="View Log Book"/>	
Photo Identity	<input type="button" value="Replace"/>	<input type="button" value="View"/>
Unique Code	<input type="button" value="Edit"/>	39546
License	<input type="button" value="Replace"/>	<input type="button" value="View"/>
Induction	<input type="button" value="Replace"/>	<input type="button" value="View"/>
Driver Conditions Licensing	<input type="button" value="Replace"/>	<input type="button" value="View"/>
Name	<input type="button" value="Edit"/>	Alex Smith
Phone	<input type="button" value="Edit"/>	022 01233554
E-mail	<input type="button" value="Edit"/>	alex@google.com

Figure 8.15 Driver profile

- Total Driven distance and hours. Log book. License information. Photo ID etc

8.5 Security of the application

1. Realness: The framework permits undeniable validness of the messages conveyed between the hubs. Each occasion is marked, which goes about as a mark on the exchanges inside it. Everything is marked, and all correspondence channels are SSL encoded. it utilizes norms for marks, hash & encoded (ECDSA, SSL/TLS etc..)

2. Information encryption: All correspondence during a gossip adjusts are SSL/TLS scrambled, utilizing session key arranged using keys of 2 of its members. In the event that an application needs further encryption, for example, encoding information inside an exchange with the goal that just a subset of the individuals can peruse it, at that point the application is allowed to do as such, and a portion of the API capacities in the stage help to make such an application less demanding to compose.

3. Security: The hashgraph stage enables every part to characterize their very own key combine, and use that as their personality. The following application is based over this stage to build up a organize, choose how individuals can be allowed to blend, for example, after arranging up a CA for the keys, or on the other hand by having votes in favour of every part, or by utilizing evidence of-stake dependent on a cryptographic money, and so on. The application just oversees protection agreement dependent on a key match for every hub.

8.6 RESEARCH QUESTIONS ANSWERED

- How Hashgraphy can be applied for real time tracking of an object?

GPS tracking and Hashgraph technology have been applied to a real-time tracking system to facilitate a fast-paced communication network for vehicles.

- How Hashgraphy can be used to deliver real-time data from weather to traffic conditions by the tracking system?

Using the consensus mechanism of Hashgraph and GPS tracking.

- How can Hashgraphy make the tracking system resilient?

Since Hashgraph is resilient to attacks like DoS, DDoS and sybil etc. it makes the tracking system resilient.

- How can Hashgraphy make the tracking system achieve faster tracking speed?

The communication of members of the hashgraph is completely dependent on the bandwidth of the network. As implemented in this chapter, the members (vehicles) of the tracking system can communicate information at the rate of 45000 transactions per second for a information size of 100 bytes per transaction at bandwidth of 100 Mbps. Thus making it an efficient system overall.

The Research questions as listed above have been answered in this thesis in the current chapter, and also in the Hashgraphy implementation chapter.

8.7 EXPERIMENTAL METRICS USED

1. Speed: Speed of the hashgraphy system is calculated on the basis of the no. of transactions per second (tps) that can be handled by system. It is said by Leemon Baird and also proven in the Hashgraph whitepaper that number of tps is directly proportional to bandwidth of the system. This thesis has proved in the implementation section that the hashgraphy system can handle upto 45,000 tps for 100 bytes/sec bandwidth. Therefore same assumption has been made for this tracking application that it can communicate upto 45,000 tps for the same bandwidth.

2. Efficiency:

Productivity of the tracking system determines its efficiency. This is explained on the below assumption. According to the tracking application, for a trip Albany Depot to Hamilton Depot (one-way), let's say,

Time = 4hrs

Speed = 50 km/hr

Driver cost =(Payrate of driver * Time)= (20 NZD per hour * 4 hours = 80 NZD)

Fuel cost = 20 NZD apprx.

Vehicle cost = 60 NZD per day

Cost per km = $\frac{\text{Driver cost} + \text{Fuel cost} + \text{Vehicle cost}}{\text{Total Distance (Km)}}$

Total cost = $\frac{80+20+60}{200} = 0.8$ cents per km

Average Time (Services) = $\frac{\text{Total (Service) time}}{\text{Total no. of trips}} = 4=4\text{hrs}$

Total Distance (km) = Speed x Average Service Time = 50 * 4 = 200 km.

So, for cost of 0.8 cents per kilometre, the distance travelled is 200km within a time of 4 hrs and vehicle speed of 50 km/hr. Hence, this is the efficiency of the system.

3. Safety: The security of hashgraphy is verified by its resilience to DDoS, Sybil and Identity loss attacks. It has been proved in the whitepaper that implementation of Keyless Signature Infrastructure (KSI) has been done to ensure identity loss attacks are avoided.

Also according to Hashgraphy white paper it is proved that, the algorithm is resilient towards DDoS and Sybil attacks. This thesis has also done an implementation of DDoS attacks in Chapter 7 on a hashgraphy prototype system and observed that it is indeed resilient to such type of attacks. Also, the concept of Crypto-hashes are integrated in the Hashgraph concept itself and are used for signing events in a hashgraph to keep it secure and immutable.

4.Tracking: Tracking current location is done using Global Positioning System (GPS) tracking device. There are various tracking devices available in local stores and we also use them in our daily lives to get directions for commuting from one place to another. I have purchased a tracking device and integrated with my tracking application's code to determine the location coordinates of vehicles.

8.8 Chapter Summary

This chapter showcased the implementation of the real-time vehicle tracking system. And how the entire system works in accordance to the Hashgraphy algorithm. The experimental metrics calculations prove the efficiency of the system in terms of speed, tracking, efficiency and safety. Hashgraph helps with faster delivery, cost management and also higher security of this tracking application. The network helps to protect assets and increase their efficiency. A standout amongst the most generally material parts of hashgraphy is that it empowers increasingly secure, straightforward data communication. In this manner, data can be reported in a perpetual decentralized record — decreasing time delays, expenses and human mistakes.

Chapter 9

SUMMARY, CONCLUSIONS, AND FUTURE WORKS

9.1 Summary & Conclusions

The most recent discussion in the tech world is with respect to blockchain, which is an established and popular member to the DLT framework and hashgraph, which is a moderately new addition to the universe of DLT.

So far we have all being utilizing the cloud to team up for storage of organizational documents, or protect data from illicit access. However, it was a noteworthy worry that "cloud" implied a focal server, with every one of the expenses and security issues that suggests. It ought to be feasible for anybody to make a common world on the web, and welcome the same number of members as they need, to team up, or purchase and move, or caper. Server shouldn't be that expensive. It should be fast & reasonable also Byzantine. Also, guidelines of its network should be implemented, regardless of whether no individual is relying on everyone. This should be what the web looks like. It has been a dream to see how the internet should run. This is the thing that we require. Be that as it may, no such framework existed, until the news of another Distributed Ledger "Hashgraph" came as a much needed refresher. On the off chance that there is hashgraph, along the gossip about gossip & virtual voting techniques, we can get correctness, speed and confirmation about Byzantine adaptation. Hashgraph has been based on an agreement framework which did not utilize more calculation, did not use more data transfer or bandwidth capacity, & did not use more of data storage, although it would be totally reasonable, quick.

The exploration discoveries after the definite investigation on Hashgraph in this thesis, says about the new highlights it compasses contrasted with the current DLTs and hypotheses.

- The Directed Acyclic Graph (DAG) alongside hashes is old, which has been widely used. Utilizing to stock up the historical backdrop of talk is latest.
- Accord calculation seems to be like casting a ballot based Byzantine calculations that have been around for a considerable length of time. Be that as it may, utilizing "virtual casting a ballot" is new.
- Disseminated database along agreement is not new. However, the stage for applications who can react for both the non-agreement and accord arrange is newest.

This gives the idea about hashgraph and the Swirlds stage can perform all the things which are as of now being finished with blockchain. Also the hashgraph has more prominent effectiveness. Be that as it may, hashgraph likewise gives recent new property types, which enables recent sorts of usage be assembled. [62]

Further to the above research findings, we have the following summary:

- Underlying technique for utilized accord.
- The Swirlds hashgraph agreement framework is utilized to accomplish accord on the reasonable request of exchanges. It additionally provides agreement timestamps when every exchange has reached at the network. Additionally it also provides agreement on authorization of principles, for example, in smart contracts.
- Consensus is accomplished when more than 2/3 of the network is on the web and taking part. Very nearly 33% of the network could be aggressors, and they would be not able stop accord, or to unjustifiably inclination what arrange turns into the agreement for the exchanges.
- Over 2/3 of the hubs should be online for accord. In the event that less are on the web, the exchanges are still conveyed to everybody online rapidly, and everybody will promptly know for sure that those exchanges are destined to be a piece of the changeless record. They just won't know the Appendix 1 accord arrange until more than 2/3 come on the web.
- Ownership of nodes.
- The stage has an option of being utilized to make a system is accessed or not.

- Latest stages of mechanism.
- Each transactions are kept into "occasions", that resemble squares, where every digger can mine numerous squares every second. There will never be a need to back off mining to avoid forking the chain. The occasions are spread by a talk convention. At the point when Aek talks with Ben, she discloses to Ben the majority of the occasions that she realizes that he doesn't, and the other way around. After Ben gets those, he makes another occasion celebrating that talks report match up, which contains the hash of the last occasion he made and the hash of the last occasion Aek made before synchronizing with him. He can likewise incorporate into the occasion any new exchanges he needs to make right then and there. Also, he signs the occasion. That is it. There is no requirement for some other correspondence, for example, casting a ballot. There is no requirement for verification of work to back off mining, since anybody can make occasions whenever.
- When a transaction is considered "safe" or "live". [63]
- On the assumption that Aek knows about an exchange, she promptly checks it and knows for sure that it will be a piece of the official history. Thus does anybody she talks with after that, following a short pause (seconds to a moment or two), she will know its EXACT area ever, and have a scientific assurance this is the consensus arrangement. That learning isn't probabilistic (as in, after 6 affirmations, you're almost certain). It's a scientific certification.
- Fault Tolerance.
- This is Byzantine fault tolerant as long as under 1/3 of the hubs are broken/traded off/assaulting. The math confirmation expect the standard suppositions: assaulting hubs can conspire, and are permitted to for the most part control the web. Their solitary limit on control of the web is that if Aek over and over sends Ben messages, they should in the long run enable Ben to get one.
- Forking vulnerability.
- The agreement can't fork as long as under 1/3 are flawed/assaulting.
- Cryptography/strength of the algorithm.
- Each part (hub) produces its very own open private key match when it joins.
- There is no pioneer.
- If a hub makes an invalid occasion (terrible hashes or awful signature) at that point that invalid occasion is overlooked by legitimate hubs amid adjusts. Mistakes in a hub can't hurt the framework as long as under 1/3 of the hubs have blunders.
- Governance enforcement.
- If an association utilizes the stage to fabricate a system, at that point that association can structure administration in the manner in which they want.
- Security standards.
- It uses standards for signatures, hashes, and encryption (ECDSA, SHA-256, AES, SSL/TLS) [64]
- Data encryption.
- All correspondences amid a chat matchup is SSL/TLS incoherent, utilizing a session key arranged utilizing the keys of the two members. On the off chance that an application needs further encryption, for example, encoding information inside an exchange with the goal that just a subset of the individuals can peruse it, at that point the application is allowed to do as such, and a portion of the API capacities in the stage help to make such an application less demanding to compose.

This thesis gives a deliberate audit of critical DL calculations - Blockchain, Tangle, and Hashgraph relevant to grasp and structure the Distributed Ledger Technology (DLT) field. Besides, in perspective of this it depicts Hashgraph thoughts, and how this latest development can enhance a continuous following and observing system. It is evident that every one of the three stages plan to accomplish comparative 'consensus' agreement objectives however through various roads as far as mechanical progressions and use of specialized designs.

This thesis plans to state that Hashgraphy is in general superior to another Blockchain, for being all the more reasonable, progressively effective and increasingly secure dependent on the outcomes from Hashgraphy and Blockchain frameworks structure usage.

9.2 Future works

The essential intention in picking Hashgraph is the freshness of innovation and its stunning properties. It claims to be exponentially brisk with a speed of >250,000 exchanges every second (tps), increasingly secure and progressively capable appeared differently in relation to other DLT computations Blockchain (3-4 tps) and Tangle (500-800 tps). [65]

This examination gave an execution investigation of the working of a Hashgraph framework on a neighbourhood PC with Windows 8 64-bit Operating framework and 64GB RAM. This investigation likewise looked at the execution of Blockchain and Hashgraph instruments. The aftereffect of Hashgraph was seen to be more effective than Blockchain. Measurements used to think about the distinction being, Speed, Efficiency, Safety and Tracking.

As per the examination and structure usage we have the accompanying figuring's.

- Speed has been determined dependent on the quantity of exchanges every second that are conveyed over the individuals from the system for both Hashgraph and Blockchain. It has been seen that Hashgraph can do upto 45000 exchanges for every second on a normal for a standard home broadband of 100 Mbps for a bundle size of 100 bytes for each exchange. In actuality, for a similar transfer speed of 100 Mbps and parcel estimate 100 bytes for each exchange, Blockchain can do 3-4 exchanges every second on a normal.
- Efficiency has been determined dependent on the speed and data transfer capacity. In the event that a framework can convey data at a rate of 45000 exchanges for each second like Hashgraph, it is certainly an effective one. Obviously, the exchange rate will go higher for a higher data transfer capacity. Lamentably, on account of Blockchain, regardless of the redesign in data transmission it can't reach up-to the limit of Hashgraph.
- Security has been estimated by making an assault situation and propelling it to the plan execution and watching the framework's weakness to the attack. There are numerous whitepapers as of now demonstrating Blockchains security, yet anyway as indicated by hypothesis in spite of the inventive changes of blockchain; the innovation itself still has some intrinsic security dangers. In addition, the progressive idea of decentralization and self-association in blockchain has just activated insignificant security issues. A portion of the many are recorded here.
- Potential Risk of Cryptography Application: The issue of private key administration isn't understood in blockchain. Existing blockchain applications more often than not utilize private key to affirm a client's personality and complete an instalment exchange. Along these lines, data cannot be distorted for private key security [32 - 34]. Unlike conventional cryptography which is open key, clients associated with blockchain are in charge of their personal private keys, it implies that a private key is created & dealt with client rather than an outsider. On the off chance that a client misplaces his private key, it becomes difficult to obtain admittance at their computerized resources of blockchain.
- One key issue is to naturally bunch the personal conduct standards of all the blockchain hubs into classes. Sybil attack[s] can be one sort of the problem. What's more, there are numerous other problems that may demonstrate irregular standards of conduct. For instance, blockchain hubs that are caught by a programmer may intermittently lead counterfeit exchanges, a dangerous hub may perform visit exchanges with little adds up to back off the network. [67]
- The agreement component of blockchain depends on a supposition that most of hubs is straightforward to run and keep up the framework. When at least one hubs control over 51% processing intensity of the entire system [66], they can combine to dispatch an attack to mess with the substance in squares and direct problematic attacks, for example, DDoS

As an answer for the above specialized constraints and security issues of blockchain, we can anticipate Hashgraph as a improvisation. This thesis has directed attacks on the Hashgraph framework and it is seen that as indicated by the Swirlds whitepaper, Hashgraph is to be sure flexible to DDoS, Sybil attacks. The hashgraph framework actualizes the Keyless Signature Infrastructure (KSI) to address issues-of key misfortune in DLTs.

- Tracking is a component of the application made to which the Hashgraph calculation has been coordinated. We utilize this application for continuous following of vehicles and traffic and climate conditions which help in effective and enhanced administration of the present following framework models. A GPS device has been incorporated to the following application to accomplish this metric computation. The following is future studies that can be done on Hashgraph based systems.

9.3 Future user cases for Consensus Mechanism

Notwithstanding customary use cases (cryptographic money, open record, smart contracts), the consensus system likewise gives reasonableness in the information exchange and requesting. This can empower use situations where the request must be reasonable, for example, a securities exchange, or a closeout, or a challenge, or a patent office, or an enormously multiplayer on the web (MMO) diversion.

9.4 Work with Industry

The outcome of a concept constructed on Swirlds policy has been reported by Ping existence for Distributed Session Management. Swirlds is as of now subsidized by a blend of financial specialists including investment, key allies, and a dedicated intermediary financing Hashgraph. In addition, Hashgraph inferred conventions can go about as a crypto-economy framework whereupon organizations can fabricate computerized resources. Like how every organization could make a site in the late 90's utilizing HTML for the framework of the page, each organization will have the capacity to make advanced savings for their administrations and items utilizing Hashgraphy that can accelerate the general execution of the organizations frameworks with its exceptionally quick abilities which will be available by a more extensive system.

Moving ahead, we should think about where we see monetary plans of action advancing in creating this vital system. It is evident that phases produced by Hashgraph have benefits through genuine integration onto a computerized economy. As various blockchain and DLT platforms are iterated on the foundation of historical competencies existing in our current technical trends, the platforms constructed thereafter are based on the regulation of utilization cases in our environment, and I see diverse sorts of utilization cases layered upon one another.

So as to really accomplish the dimension of communication and adaptability that is needed by our current frameworks, a convention must be constructed and structured in view of all things considered, just like how the web was first planned. Hashgraphy can go about as the primary innovation that benefits an evolutionary system to incorporate the essential factors in trade domains. However, the stage is at present advancing and could likewise profit by the inherent capacities in the DLT partners.

At some point in the future we will incorporate advances that have not yet been achieved, conventions ought to be inspected on how useful will they be in the aspiring age of the web and sometimes the most obvious arrangement isn't to concentrate on just a single innovation.

APPENDIX

Appendix A: Software Specifications

Name of Software	Specification
Eclipse IDE	Oxygen 4.7
Java SE	Version 8
Swirls SDK	
Java Security	Version 8
Windows 8 Command prompt	Ping command (to launch ping of death attack)
Nemesy tool	Online tool to launch DDoS attack
GPS device	Tracking the location of vehicles in the tracking app
Microsoft SQL Server	2016
Database	MySQL
Programming Languages	Java, Go, PHP, Javascript, HTML, CSS, Angular JS
Features	Firewall protection, VPN support, MPLS support, hardware encryption, and Quality of Service (QoS)

Table 1: Software Specifications

REFERENCES

- [1]"What is cryptography? - Definition from WhatIs.com", *SearchSecurity*, 2019. [Online]. Available: <https://searchsecurity.techtarget.com/definition/cryptography>.
- [2]2019. [Online]. Available: <https://www.thegeekstuff.com/2012/07/cryptography-basics/>.
- [3]"Basic cryptology concepts", *ibm.com*, 2019. [Online]. Available: <https://www.ibm.com/developerworks/tivoli/tutorials/s-crypto/s-crypto.html>.
- [4]"What is cryptology? - Definition from WhatIs.com", *SearchSecurity*, 2019. [Online]. Available: <https://searchsecurity.techtarget.com/definition/cryptology>.
- [5]"What is Cryptanalysis? - Definition from Techopedia", *Techopedia.com*, 2019. [Online]. Available: <https://www.techopedia.com/definition/1769/cryptanalysis>.
- [6]"What is Cryptanalysis (All Types of Attacks and Tools) - Kifanga", *Kifanga*, 2019. [Online]. Available: <https://kifanga.com/what-is-cryptanalysis/>.
- [7]"Cryptanalysis Tools", *InfoSec Resources*, 2019. [Online]. Available: <https://resources.infosecinstitute.com/cryptanalysis-tools/#gref>.
- [8] "Hashgraph wants to give you the benefits of blockchain without the limitations", *TechCrunch*, 2018. [Online]. Available: <https://techcrunch.com/2018/03/13/hashgraph-wants-to-give-you-the-benefits-of-blockchain-without-the-limitations/>
- [9] What is Cryptoeconomics? The Ultimate Beginners Guide - Blockgeeks. (2018). Retrieved from <https://blockgeeks.com/guides/what-is-cryptoeconomics>
- [10]"Making Sense of Cryptoeconomics - CoinDesk", *CoinDesk*, 2019. [Online]. Available: <https://www.coindesk.com/making-sense-cryptoeconomics>.
- [11] Tosh, K. Deepak, "Security implications of blockchain cloud with analysis of block withholding attack," Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. IEEE Press, 2017, pp: 458-467.
- [12] Huang, B., Liu, Z., Chen, J., Liu, A., Liu, Q. and He, Q. (2017). Behavior pattern clustering in blockchain networks. *Multimedia Tools and Applications*, 76(19), pp.20099-20110.
- [13] " Mitigating attacks on blockchain | Blockchain | Fintech |", *Allerin.com*, 2017. [Online]. Available: <https://www.allerin.com/blog/mitigating-attacks-blockchain>.
- [14] L. Baird, "Swirls and Sybil Attacks", 2016.
- [15]A. Ommani, "Strengths, weaknesses, opportunities and threats (SWOT) analysis for farming system businesses management: Case of wheat farmers of Shadervan District, Shoushtar Township, Iran", *Academicjournals.org*, 2011. [Online]. Available: https://www.academicjournals.org/article/article1380639652_Ommani.pdf
- [16] "Twitter", *Twitter.com*, 2018. [Online]. Available: <https://twitter.com/ahier/status/904296280356118528>
- [17]*Documents.worldbank.org*, 2019. [Online]. Available: <http://documents.worldbank.org/curated/en/177911513714062215/pdf/122140-WP-PUBLIC-Distributed-Ledger-Technology-and-Blockchain-Fintech-Notes.pdf>.

[18]"Different Types of DLT | SoftServe", Softserveinc.com, 2019. [Online]. Available: <https://www.softserveinc.com/en-us/blogs/types-of-dlt/>.

[19]"Blockchains & Distributed Ledger Technologies", BlockchainHub, 2019. [Online]. Available: <https://blockchainhub.net/blockchains-and-distributed-ledger-technologies-in-general/>.

[20] 2018. [Online]. Available: https://www.researchgate.net/figure/Block-structure_fig2_318131748.

[21] M. Sharples and J. Domingue, "The blockchain and kudos: A distributed system for educational record, reputation and reward," in Proceedings of 11th European Conference on Technology Enhanced Learning (EC-TEL 2015), Lyon, France, 2015, pp. 490–496.

[22] "Blockchain and Distributed Ledger Technology | SAP", SAP, 2018. [Online]. Available: <https://www.sap.com/products/leonardo/blockchain.html>.

[23] V. Buterin, "On public and private blockchains," 2015. [Online]. Available: <https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/>

[24] "Consortium chain development." [Online]. Available: <https://github.com/ethereum/wiki/wiki/Consortium-Chain-Development>

[25] V. Buterin, "On public and private blockchains," 2015. [Online]. Available: <https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/>

[26] M. Vukolić, "The quest for scalable blockchain fabric: Proof-of-work vs. bft replication," in International Workshop on Open Problems in Network Security, Zurich, Switzerland, 2015, pp. 112–125.

[27] "Crypto-currency market capitalizations," 2017. [Online]. Available: <https://coinmarketcap.com>

[28] T. Crain, V. Gramoli, M. Larrea, and M. Raynal, "(leader/randomization/signature)-free byzantine consensus for consortium blockchains," arXiv, Tech. Rep. 1702.03068, 2017.

[29] NRI, "Survey on blockchain technologies and related services," Tech. Rep., 2015. [Online]. Available: http://www.meti.go.jp/english/press/2016/pdf/0531_01f.pdf

[30] D. Lee Kuo Chuen, Ed., Handbook of Digital Currency, 1st ed. Elsevier, 2015. [Online]. Available: <http://EconPapers.repec.org/RePEc:eee:monogr:9780128021170>

[31] V. Buterin, "A next-generation smart contract and decentralized application platform," white paper, 2014.

[32] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ecdsa)," International Journal of Information Security, vol. 1, no. 1, pp. 36–63, 2001.

[33] Z. Ye, L. Wei, N. Meng, Y. Shi and F. "From Bitcoin to Cybersecurity: a Comparative Study of Blockchain Application and Security Issues", the 2017 4th International Conference on Systems and Informatics (ICSAI 2017), 2017.

[34] "The biggest mining pools." [Online]. Available: <https://bitcoinworldwide.com/mining/pools/>

[35] Huang, B., Liu, Z., Chen, J., Liu, A., Liu, Q. and He, Q. (2017). Behavior pattern clustering in blockchain networks. Multimedia Tools and Applications, 76(19), pp.20099-20110.

- [36] L. Baird, "THE SWIRLDS HASHGRAPH CONSENSUS ALGORITHM: FAIR, FAST, BYZANTINE FAULT TOLERANCE", 2016.
- [37] P. Vasin, "Blackcoins proof-of-stake protocol v2," 2014. [Online]. P. Vasin, "Blackcoins proof-of-stake protocol v2," 2014. [Online]. Available: <https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf>
- [39] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum ProjectYellow Paper, 2014.
- [40] V. Zamfir, "Introducing casper the friendly ghost," Ethereum Blog URL: <https://blog.ethereum.org/2015/08/01/introducing-casperfriendly-ghost>, 2015.
- [41] "Tangle (IOTA) – public IOTA", Publiciota.com, 2018. [Online]. Available:<http://publiciota.com/wikiota/tangle-iota>
- [42] "iota statistics - Google Search", Google.co.nz, 2018. [Online]. Available: https://www.google.co.nz/search?rlz=1C1CHZL_enUS748US748&biw=1366&bih=613&tbm=isch&sa=1&ei=f2aUW6SfJ8Xc8QWqqp7IAg&q=iota+statistics&o=iota+statistics&gs_l=img.3..0i24k1.61978.67179.0.67787.29.18.0.4.4.0.398.3594.29j4.13.0.....1c.1.64.img..18.10.1810...0j35i39k1j0i67k1j0i5i30k1j0i8i30k1.0.BluQ0M7OMQo#imgsrc=d7fm_kssLIL35M:
- [43] Baird, L. (2016): The SWIRLDS Hashgraph Consensus Algorithm: Fair,fast, Byzantine Fault Tolerance.
- [44]"hashgraph blockchain tangle - Google Search", Google.co.nz, 2018. [Online]. Available:https://www.google.co.nz/search?rlz=1C1CHZL_enUS748US748&biw=1366&bih=613&tbm=isch&sa=1&ei=xBiVW5rfGYPZQbu8Lu4AQ&q=hashgraph+blockchain+tangle&oq=hashgraph+blockchain+tangle&gs_l=img.3...11861.21385.0.21709.50.31.0.0.0.0.250.4271.0j11j10.21.0....0...1c.1.64.img..30.14.2835.0..0j0i5i30k1j0i24k1j0i8i30k1.0.KLvqPxlgYY4#imgsrc=nT2uHXUr9WGyaM:
- [45]M. Conti, S. E. C. Lal and S. Ruj, "A Survey on Security and Privacy Issues of Bitcoin", IEEE Communications Surveys & Tutorials, pp. 1-1, 2018.
- [46]C. Announcements, G. Author, L. Harding and J. Sanchez, "How many transactions can bitcoin pass per second? What are the long term plans? ", Coinnounce, 2018. [Online]. Available: <https://coinnounce.com/number-of-transactions-bitcoin-per-second/>
- [47]G. Baker-Whitelaw and B. Vincent, "The world's fastest recorded internet speeds will blow your mind", The Daily Dot, 2018. [Online]. Available: <https://www.dailydot.com/debug/fastest-internet-speed-ever-recorded/>
- [48]2019. [Online]. Available: <http://www.ambyssoft.com/essays/agileLifecycle.html>].
- [49]"2013 Agile Project Initiation Survey Results", Ambyssoft.com, 2019. [Online]. Available: <http://www.ambyssoft.com/surveys/projectInitiation2013.html>.
- [50]"impact of ddos attacks - Google Search", Google.com, 2019. [Online]. Available: https://www.google.com/search?q=impact+of+ddos+attacks&tbm=isch&source=lnms&sa=X&ved=0ahUKEwi2hJaj1frfAhVCSX0KHZkPCEUQ_AUICygC&biw=1366&bih=626&dpr=1#imgsrc=tF2kLPw9U0e6M:
- [51] "What is Proof of Stake? (PoS) | Lisk Academy", Lisk, 2018. [Online]. Available: <https://lisk.io/academy/blockchain-basics/how-does-blockchain-work/proof-of-stake>
- [52]"Whitepaper:Nxt-NxtWiki", Nxtwiki.org,2019.[Online].Available:

<https://nxtwiki.org/wiki/Whitepaper:Nxt>.

[53]W. stake?, "What are the downsides of proof of stake?", Bitcoin Stack Exchange, 2019. [Online]. Available: <https://bitcoin.stackexchange.com/questions/25743/what-are-the-drawbacks-of-verification-of-stake>.

[54]Swirlds.com, 2019. [Online]. Available: <http://www.swirlds.com/wp-content/uploads/2016/06/2016-05-31-Swirlds-Consensus-Algorithm-TR-2016-01.pdf>

[55] "Code your own Proof of Stake blockchain in Go! – Coral Health – Medium", Medium, 2019. [Online]. Available:<https://medium.com/@mycoralhealth/code-your-own-proof-of-stake-blockchain-in-go-610cd99aa658>.

[56]Miguel Miguel Correia, Giuliana Santos Veronese, Nuno Ferreira Neves, and Paulo Verissimo. Byzantine consensus in asynchronous message-passing systems: a survey. International Journal of Critical Computer-Based Systems, 2(2):141–161, 2011.

[57]Leemon.com, 2019. [Online]. Available: <http://leemon.com/papers/2016b.pdf>

[58]"Swirlds and Sybil Attacks – Zane Witherspoon – Medium", Medium, 2019. [Online]. Available: <https://medium.com/@zanewitherspoon/swirlds-and-sybil-attacks-90357eb1e29>.

[59]C. Douligieris and A. Mitrokotsa, "DDoS attacks and defense mechanisms: a classification - IEEE Conference Publication", ieeexplore.ieee.org, 2019. [Online]. Available: <https://ieeexplore.ieee.org/document/1341092>

[60]D.Bekerman, Incapsula.com,2019.[Online].Available:<https://www.incapsula.com/blog/gbps-pps-rps-ddos-attacks.html>.

[61]"Jira | Issue & Project Tracking Software | Atlassian", Atlassian, 2019. [Online]. Available: <https://www.atlassian.com/software/jira>

[62] G. Samman and G. Samman, "A New Approach to Consensus: Swirlds HashGraph", SAMMANTICS,2019.[Online].Available:<http://sammantics.com/blog/2016/7/27/hashgraph-consensus>

[63] Zyskind, Guy, O. Nathan, "Decentralizing privacy: Using blockchain to protect personal data," Security and Privacy Workshops (SPW), IEEE, 2015, pp: 180-184.

[64] Kosba, Ahmed, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," Security and Privacy (SP), 2016 IEEE Symposium on. IEEE, 2016, pp:839-858.

[65] Meiklejohn, Sarah, Claudio Orlandi, "Privacy-enhancing overlays in bitcoin," International Conference on Financial Cryptography and Data Security, Springer Berlin Heidelberg, 2015, pp:127 141.

[66]"The biggest mining pools." [Online]. Available:<https://bitcoinworldwide.com/mining/pools/>

[67] Huang, B., Liu, Z., Chen, J., Liu, A., Liu, Q. and He, Q. (2017). Behavior pattern clustering in blockchain networks. Multimedia Tools and Applications, 76(19), pp.20099-20110.

[68] Demirbas, M., & Song, Y. (2006). An RSSI-based scheme for Sybil attack detection in wireless sensor networks . IEEE Computer Society International Symposium on World of Wireless, Mobile and Multimedia Networks, (pp. 570-574).

[69] Dhamodharan, U. S., & Vayanaperumal, R. (2015). Detecting and preventing Sybil attacks in wireless sensor networks using message authentication and passing

method. Scientific World Journal, 1(1), 13-17

[70] Amuthavalli, R., & Bhuvaneswaran, R. S. (2014). Detection and prevention of Sybil attack in wireless sensor network employing random password comparison method. Journal of Theoretical and Applied Information Technology, 67(1), 236-246

[71] Allen, D. (2017). Discovering and Developing the Blockchain Crypto-Economy. *SSRN Electronic Journal*. doi: 10.2139/ssrn.2815255

[72] L. Baird, "Overview of Swirlds Hashgraph ", 2016

[73] Batsaikhan, U. (2018). Cryptoeconomics – the opportunities and challenges of blockchain | Bruegel. Retrieved from <http://bruegel.org/2017/07/cryptoeconomics-the-opportunities-and-challenges-of-blockchain/>

[74] Denial-of-service Attacks Rip the Internet. IEEE Computer Society 33(4), 12-17. doi: 10.1109/MC.2000.839316

[75] Agent Based Preventive Measure for UDP Flood Attack in DDoS Attacks. International Journal of Engineering Science and Technology, 2(8), 3405- 3411. Retrieved from <http://www.ijest.info>

[76] Computer Security Handbook (4 ed.). New York: John Wiley & Sons, Inc., 2002

[77] Arbor Special Report: Worldwide Infrastructure Security Report. Retrieved from http://pages.arbornetworks.com/rs/arbor/images/wisr2012_en.pdf

[78] Kolahi, S.S., Li, P. (2011). Evaluating IPv6 in Peer-to-Peer 802.11n Wireless LANs. IEEE Internet Computing, 15(4), 70-74. doi: 10.1109/MIC.2011.89

[79] Bandwidth-IPSec Security Trade-off in IPv4 and IPv6 in Windows 7 Environment. Paper presented at the Second International Conference on Future Generation Communication Technology, London. doi: 10.1109/FGCT.2013.6767214

[80] Study of DDoS Attacks Using DETER Testbed. Journal of Computing and Business Research, 3(2), 1-13.

[81] 2018, N. (2018). Number of Blockchain wallets 2018 | Statista. Retrieved from <https://www.statista.com/statistics/647374/worldwide-blockchain-wallet-users/>



Declaration

Name of candidate: Sushmitha Kommushetty

This Thesis/Dissertation/Research Project entitled: “**A crypto-economy based distributed & asynchronous Hashgraphy algorithm for a Tracking system**”.

is submitted in partial fulfillment for the requirements for the Unitec degree of Master of Computing (Level-9)

Principal Supervisor: Mr Bahman Sassani (Sarrafpour)

Associate Supervisor/s: Dr Iman Ardekani

CANDIDATE’S DECLARATION

I confirm that:

- This Thesis/Dissertation/Research Project represents my own work;
- The contribution of supervisors and others to this work was consistent with the Unitec Regulations and Policies.
- Research for this work has been conducted in accordance with the Unitec Research Ethics Committee Policy and Procedures, and has fulfilled any requirements set for this project by the Unitec Research Ethics Committee.

Research Ethics Committee Approval Number: N/A

Candidate Signature:

Date: 8 February 2019

Student number: 1467178

Full name of author: ...Sushmitha Kommushetty.....

ORCID number (Optional):

Full title of thesis/dissertation/research project ('the work'):

... A crypto-economy based distributed & asynchronous Hashgraphy algorithm for a Tracking system.....

Practice Pathway:..... Cyber Security & Information Technology

Degree:...Master of Computing.(Level-9) ..

Year of presentation:2018.....

Principal Supervisor: ...Mr Bahman Sassani (Sarrafpour)

Associate Supervisor: ...Dr Iman Ardekani.....

Permission to make open access

I agree to a digital copy of my final thesis/work being uploaded to the Unitec institutional repository and being made viewable worldwide.

Copyright Rights:

Unless otherwise stated this work is protected by copyright with all rights reserved.

I provide this copy in the expectation that due acknowledgement of its use is made.

AND

Copyright Compliance:

I confirm that I either used no substantial portions of third party copyright material, including charts, diagrams, graphs, photographs or maps in my thesis/work or I have obtained permission for such material to be made accessible worldwide via the Internet.

Signature of author



(Sushmitha Kommushetty)

Date: ...08 / Feb / 2019