Fall 2015

BitFed, A Centralized Cryptocurrency with Distributed Miners

Shruti Sharma
San Jose State University

BitFed, A Centralized Cryptocurrency with Distributed Miners

A Project

Presented to

The Faculty of the Department of Computer Science

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Shruti A Sharma

December 2015

The Designated Project Committee Approves the Project Titled

BitFed, A Centralized Cryptocurrency with Distributed Miners

by

Shruti A Sharma

APPROVED FOR THE DEPARTMENTS OF COMPUTER SCIENCE

SAN JOSE STATE UNIVERSITY

December 2015

Dr. Thomas Austin	Department of Computer Science
Dr. Chris Pollett	Department of Computer Science
Mr. Ronald Mak	Department of Computer Science

ABSTRACT

BitFed, A Centralized Cryptocurrency with Distributed Miners

by Shruti A Sharma

Bitcoin is a decentralized peer-to-peer electronic currency wherein all the payments are sent from one transactor to another directly [1]. Financial institutions are not present in the protocol, hence, there are lower processing fees. The distributed nature provides resilience to Bitcoin transactions, and it operates on mathematical principles and cryptographic proofs.

As per Bitcoin generation algorithm, the number of bitcoins in existence will never surpass 21 million, which will lead to deflation and encourage hoarding. In this project, we have implemented a Bitcoin-like currency in order to mitigate the issue of deflation [7]. The idea for our protocol is based on the US Federal reserve, which is the gatekeeper of the US economy and promotes the stability of prices by preserving the purchasing power of dollar over a long period of time. Our approach to fix the Bitcoin deflation issue is by creating BitFed, a centralized authority that decides the rate of issuance of bitcoins each day while otherwise maintaining the decentralized nature of protocol. Specifically, the BitFed plays no role in verifying transactions. In order to prove that this approach is viable, we have developed a simulator in Java that emulates the Bitcoin protocol with the BitFed and tested it by introducing cheaters in the network. The system has no forks when majority of the miners are good (more than 51 percent). After this we see several forks in the system, since the network is controlled by cheaters.

ACKNOWLEDGMENTS

This thesis owes its existence to the support and inspiration of several people. Firstly, I am greatly indebted to Dr. Thomas Austin for his expert guidance and suggestions. The valuable insights provided by him throughout the course of this project have been precious for the development of this thesis. His idea of conducting group meetings with other students helped me in knowing the perceptions of different people about my project.

I would also like to express my gratitude to Mr. Ronald Mak and Dr. Chris Pollett for graciously consenting to be on my committee. They have really been a source of encouragement and enthusiasm, not only during this thesis project but also during the two years of my Master's program.

Thanks are also due to all my friends for their help and companionship which made graduate school a memorable experience!

And to my husband Niranjan Manjunath, thank you for putting up with my tough schedules and making my life pleasurable when I was busy! Your unflagging love and unconditional support has made this dream a reality.

Thank you all!

TABLE OF CONTENTS

CHAPTER

APPENDIX

LIST OF FIGURES

CHAPTER 1

Introduction

In an economy where there is a central reserve, currency is issued by the reserve bank at a rate that is supposed to match the growth of the amount of goods that are bartered so that these goods can be exchanged with stable prices. The monetary base is controlled by a central bank.

In a completely decentralized monetary system, there is no single authority with paramount power that regulates the monetary base. Instead, currency creation is by the nodes in a peer-to-peer network that solve a tough cryptographic problem. The Bitcoin generation algorithm has determined beforehand the method of currency creation and also the rate at which currency is created. The bitcoins are created each time a miner mines a new block. The rate at which blocks are mined is 6 per hour and remains constant over time. The rate at which new bitcoins are generated per block is stipulated to reduce geometrically, with a 50 percent reduction in each span of four years[15]. The result is that the number of bitcoins in the Bitcoin network will never exceed 21 million. Because the monetary base of bitcoins cannot be expanded, there will be limited number of bitcoins that can be traded. This will lead to a general decline in personal spending and also to a reduction in the supply of money which would lead to severe deflation.

According to Keynesian economists, deflation is bad for an economy because it encourages people and corporations to save bitcoins rather than spend them. Price deflation encourages hoarding rather than spending since people anticipate that limited number of bitcoins will lead to a steep increase in price and that they can trade them later for a higher price. There will come a point where, as demand rises, the value of the currency will only go up and that could lead to hoarding on an even larger scale.

As a part of this thesis, we propose a change to the Bitcoin protocol to handle deflation. We suggest BitFed : a central authority that decides the rate at which new coins are 'mined'. With this approach, the supply of coins can be controlled to benefit the economy while retaining the decentralized and distributed verification of transactions.

1.1 History of currency

Currency can be defined as any form of monetary unit that is in public circulation. It refers to the unit that is legally accepted for transactions and designated by a central governing authority, such as the Federal reserve in the US. In some economies, currency can refer to an object that has value and can be used as an entity of exchange for other objects.

The various currencies in use today have mainly evolved from two innovations of 2000 BC. People stored grains in temple granaries in Mesopotamia and received a receipt for storing the grains [8]. Hence money was a form of receipt, representing grain stored and was called representative currency. Representative currency was followed by commodity money. In 600 - 700 BC, metals were used as tokens to represent the value stored in the form of commodities. This was the origin of commodity money wherein the value of coins was tied to content of the metal [8].

In 900 AD in China, the large number of copper coins were becoming really cumbersome to be exchanged physically and hence a new form of money was introduced. This was paper money, i.e. banknotes. A banknote also known as a bill in countries such as Canada and the United States, is a type of currency, and commonly used as a means of transaction. Coins and banknotes together comprise the cash forms of money [12].

With the recent advancements in the field of computers and the Internet, a new form of currency is now in demand : digital currency. It is different from physical currencies such as bank notes and coins since it is used over the internet but it does exhibit properties similar to physical currencies. It allows for immediate money exchange and border less transfer of ownership. Virtual currencies and cryptocurrencies are types of digital currencies.

Bitcoin is the first decentralized digital currency [9] (no central point of control over the money supply) that can be exchanged for goods and services. It was introduced in 2008 by pseudonymous developer Satoshi Nakamoto [1].
Some of the unique are :

- **Digital** : bitcoins are mined they cannot be issued. They must be generated through computerized methods alone.
- **Decentralized** : bitcoins are not monitored by any government agency or banks. This makes transactions cheaper as there are lower transaction fees that need to be paid to a third party. Also the people involved in the transactions is not known to anyone.
- **Anonymity** : transactions allow for anonymity and are almost instantaneous.
- **Global** : bitcoins are border less currencies and can be used anywhere.

1.2 Motivation

Currently there is no fool proof mechanism to control deflation in bitcoins. Within a few years it will suffer from galloping deflation [16]. The user base of Bitcoin is currently small but as the number of users expands, the demand will force the value higher and higher. Both inflation and deflation are problematic for economies, but deflation is generally considered the more problematic amongst the two [11]. It can eventually lead to an economic crash. Bitcoin is becoming widely used for transactions, and hence governments may want to use a similar currency for transactions, but with some ability to manage inflation and deflation. This has motivated us to improve the protocol by creating BitFed. Our BitFed protocol regulates the number of bitcoins that can be created and hence we need a central authority to determine the creation rate of bitcoins to control deflation while maintaining the verification decentralized.

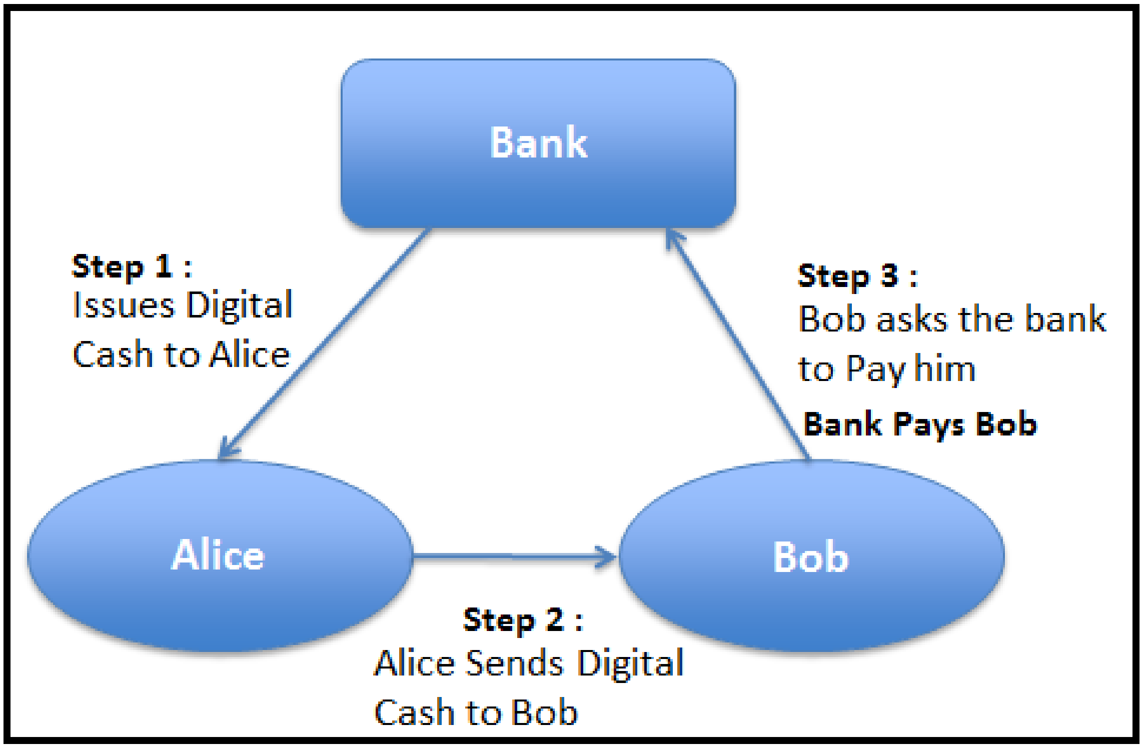
1.3 Other approaches and Flaws

Several cryptocurrencies gained popularity before Bitcoin came into picture. They had several flaws and could not garner sufficient support.

1.3.1 Digital Cash

It is extremely annoying to take cash everywhere and use it for transactions. Credit cards and cheques have helped the cause but they still do not eliminate the need to physically use them for exchanges. Digital cash tries to eliminate the physical use of cash by allowing authenticated untraceable transactions thereby providing anonymity. Digital cash has value since there is a trusted third party such as a government involved.

Figure 1: Digital Cash protocol



The way Digital cash works is depicted in the Figure 1. Alice cannot use the same cash for some other transaction otherwise she will be caught. In the same way, Bob cannot deposit the same amount into two different accounts otherwise he will be caught. In the digital cash system, identity of an individual is anonymous unless he is caught for double spending the same amount.

There are a few issues attached with using Digital Cash extensively:

- **Many situations do not need so much privacy** : In most cases it is not required that the transactions are private.

- **Communications with the bank are an overhead** : It is cumbersome to approach the bank and obtain signatures.

- **Digital cash has one way anonymity** : In the above example Alice is anonymous but Bob is not since he is the one receiving money from the bank. The government can there by know how much money you have but it does not know how and where you spent it.

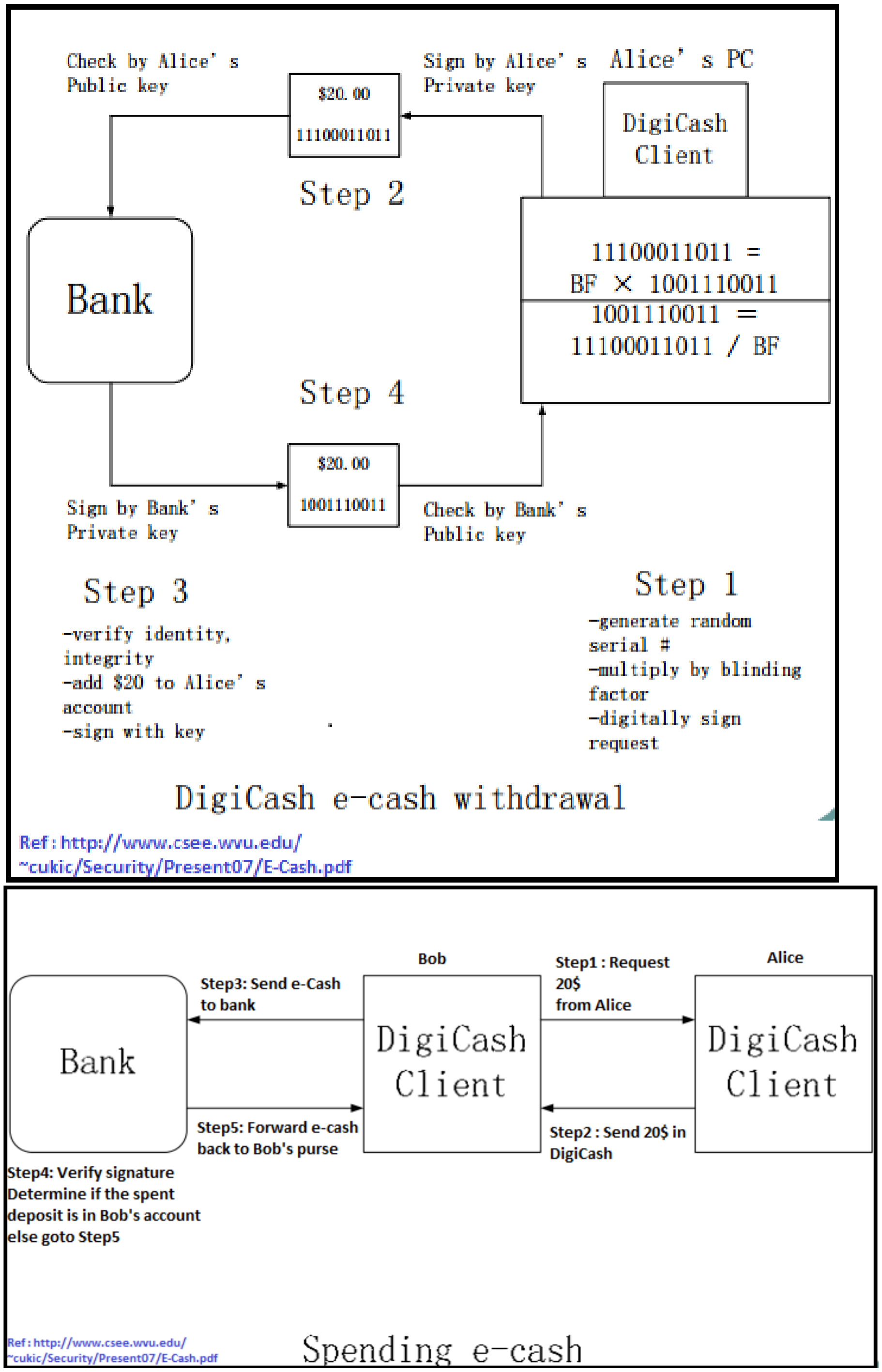
- **Digital Cash can be easily used for illegal activities** : It is very easy to use digital cash for money laundering without the ability to trace the actual culprit.

1.3.2 DigiCash

This is a Digital payment system which was proposed by David Chaum in 1983 in his paper [13]. It is a stored-value cryptographic coin system that facilitates internet-based transactions using software that runs on personal computers. Cryptographic tokens are used to represent the value of DigiCash. These tokens can be withdrawn from bank accounts, deposited in bank accounts, or transferred to other people. DigiCash is unique in its implementation of electronic cash because it has tried to keep the transactions anonymous and untraceable very similar to cash transactions. DigiCash uses blind signatures for untraceable payments [13].

The protocol transactions can be diagrammatically represented as shown in Figure 2:

Figure 2: DigiCash protocol


Check by Alice' s Public key
$20. 00
11100011011
Sign by Alice' s Private key
Alice' s PC
DigiCash Client
Step 2
Bank
11100011011 = BF × 1001110011
1001110011 = 11100011011 / BF
Step 4
$20. 00
1001110011
Sign by Bank' s Private key
Check by Bank' s Public key
Step 3
-verify identity, integrity
-add $20 to Alice' s account
-sign with key
Step 1
-generate random serial #
-multiply by blinding factor
-digitally sign request
DigiCash e-cash withdrawal
Bob
Alice
Bank
DigiCash Client
DigiCash Client
Step3: Send e-Cash to bank
Step1 : Request 20$ from Alice
Step5: Forward e-cash back to Bob's purse
Step2 : Send 20$ in DigiCash
Step4: Verify signature Determine if the spent deposit is in Bob's account else goto Step5
Spending e-cash

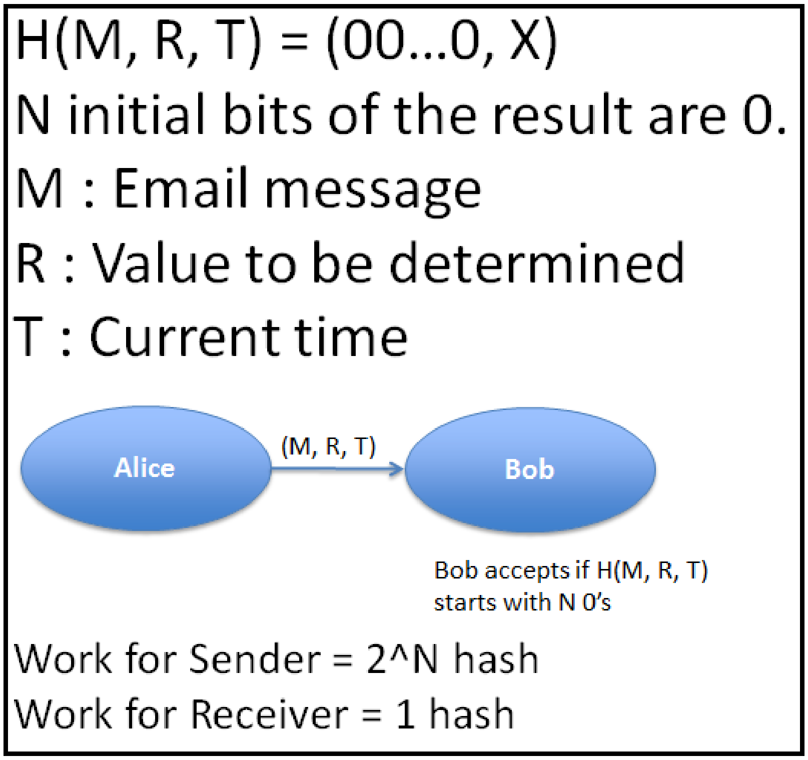
Ref : http://www.csee.wvu.edu/~cukic/Security/Present07/E-Cash.pdf

Ref : http://www.csee.wvu.edu/~cukic/Security/Present07/E-Cash.pdf

1.3.3 Hash Cash

Hash Cash is a proof of work system which is mainly used to reduce the number of denial of service attacks as well as spam mails. The mining portion of Bitcoin uses Hash Cash [5].

Figure 3: Hash Cash Algorithm



H(M, R, T) = (00...0, X)

N initial bits of the result are 0.

M : Email message

R : Value to be determined

T : Current time

Bob accepts if H(M, R, T) starts with N 0's

Work for Sender = 2^N hash

Work for Receiver = 1 hash

The way Hash Cash works is depicted in Figure 3. The system assumes that since the sender has spent effort in generating the hashes, the sender is not a spammer. There are a few issues attached with using Hash Cash extensively

- **Legit servers may be stuck calculating hashes**

- **Botnets can be used by spammers**

1.3.4 e-Cash

This is another cryptocurrency introduced in 1990's as per which, the bank acts as a central authority and maintains all the transactions as a part of a public ledger. Maintaining a public ledger helps in detecting double spending and ensuring the validity of coins. The entire set of valid coins are not published since it seems impractical. [5]

1.3.5 B-money

B-money is a cryptocurrency that was introduced in 1998 and was the first system that publicly broadcasts all the transactions [5].

1.3.6 Smart Contracts

It is a system that was suggested in 1990's to enable transacting parties to decide on an agreement that could be enforced by using cryptographic principles [5].

1.3.7 Our Approach with BitFed

There are a few approaches to handle the issue of deflation in Bitcoin. One of the approaches suggests controlled supply algorithm mitigate the issue of deflation. In an electronic cash system that is controlled by the government, currency is issued by a central bank at a fixed rate which depends on the amount of goods traded at a stable price. The controlled supply approach suggests controlling deflation by creating new bitcoins when required. In a decentralized Bitcoin generation, the algorithm determines in advance how and at what rate the currency will be created.

Hence controlling supply is not a viable approach.

1.4 My approach and contributions

My main contribution to this project is implementing a BitFed: a cryptocurrency patterned after Bitcoin. I have implemented this project in Java and designed the user interface using Swing. After implementing the simulator, I have integrated a central authority to determine the rate of bitcoins on daily basis. The complete steps of the protocol are listed as a part of Section 2. Please refer for further understanding of the approach.

1.5 Summary of results

We have tested the protocol by generating transactions every 15 seconds and with BitFed sending the rate change messages every 30 seconds. This testing was run for more than 100 transactions and we did not observe forks. The block chains were synchronized in the right way and hence we were able to prove that the BitFed approach will work well in practice.

CHAPTER 2

Bitcoin Protocol

This chapter has more information on the inner workings of the Bitcoin protocol.

2.1 What is Bitcoin?

Bitcoin uses a collection of concepts that form the basis of a new payment system in the digital money ecosystem. There is no central authority in Bitcoin and neither does it have any middlemen involved. Hence, it can be called the first decentralized peer-to-peer payment system that is essentially controlled by its users. Since it is a digital currency, it is entirely virtual and its users follow the Bitcoin protocol and communicate with each other via the internet. Its advantages are that it is fast, secure, and not limited to specific countries [1].

2.2 Who created Bitcoin?

Satoshi Nakamoto published the proof of work and specification as a part of a cryptography mailing list in 2008 [5]. It is a new form of money that uses cryptography concepts such as digital signature and hashing to control its creation and transactions, rather than a central authority. It is open source software with a growing developer community.

Bitcoin is based on the assumption that people's greedy behavior cannot bring the system down since most of the miners are good.

2.3 Chronology of events

- The proof of work was published for the first time in 2008 [5].
- On 3rd January, 2009, the first block in Bitcoin also known as the Genesis block was mined [5].
- On 20th of May, 2009, for the first time Bitcoin was used to place a pizza delivery order [5].
- As per date, the highest price that Bitcoin has garnered has been 1200 $ in 2013.

2.4 Basis of Bitcoin?

Bitcoin is based on having a transaction ledger that is public to all the people on the network. This ledger is maintained in a distributed manner by the participants in the network who work to solve cryptographic puzzles. When these 'miners' find the solution (known as a proof-of-work) they are rewarded with newly minted coins, as well as any transaction fees offered. On an average, there are 30 Bitcoin transactions per minute and new bitcoins are created every 10 minutes.

2.5 Main properties of cryptocurrencies

All the communication happens on a secure communication network and Bitcoin strongly adheres to the main ideas of Cryptography.

The main components have been suggested by Tatsuaki Okamoto and Kazuo Ohta [14]:

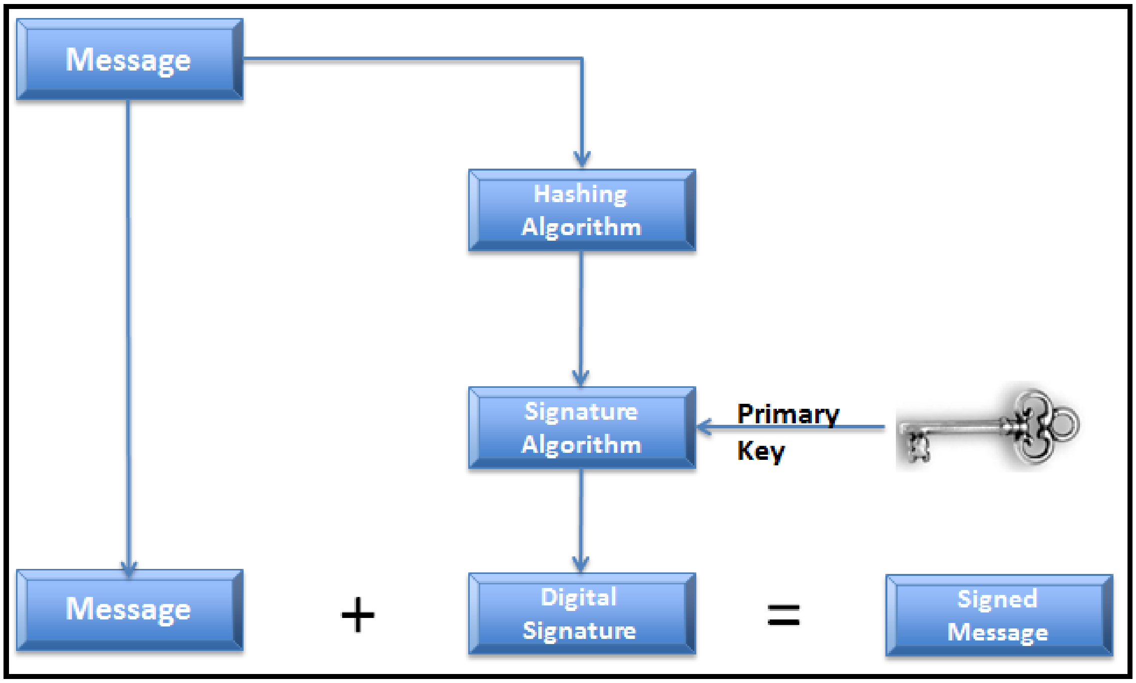
- **Security : Am I paying the right person?**

  Authentication is achieved in Bitcoin protocol using public key and digital signatures.

- **Transferability: Can I make the transaction any time I want?**
- **Integrity: Can a single coin be spent twice? (This is known as double spending) Can a transaction be reversed?**

  Integrity is achieved using cryptographic hash functions. Double spending is another major problem that has plagued several cryptocurrencies that preceded Bitcoin. The issue of double spending arises when same token of money is spent more than once. It is easy to handle the problem when there is a central authority that authorizes all the transactions. In the case of Bitcoin where there is no central agency, the issue of double spending is prevented by broadcasting transaction details to all the nodes on the network. This enables all the nodes on the network to determine whether the transaction is permissible or not.

Figure 4: Generation of Signed transaction message



The Figure 4 shows us how the message and signature together form the signed message.

- **Anonymity: Does any miner know who the sender/receiver is?**
- **Divisibility: Can you divide the cash into smaller units.**
- **Independence: Security is independent of physical location. Transfers are done through the network.**

2.6 How does Bitcoin work?

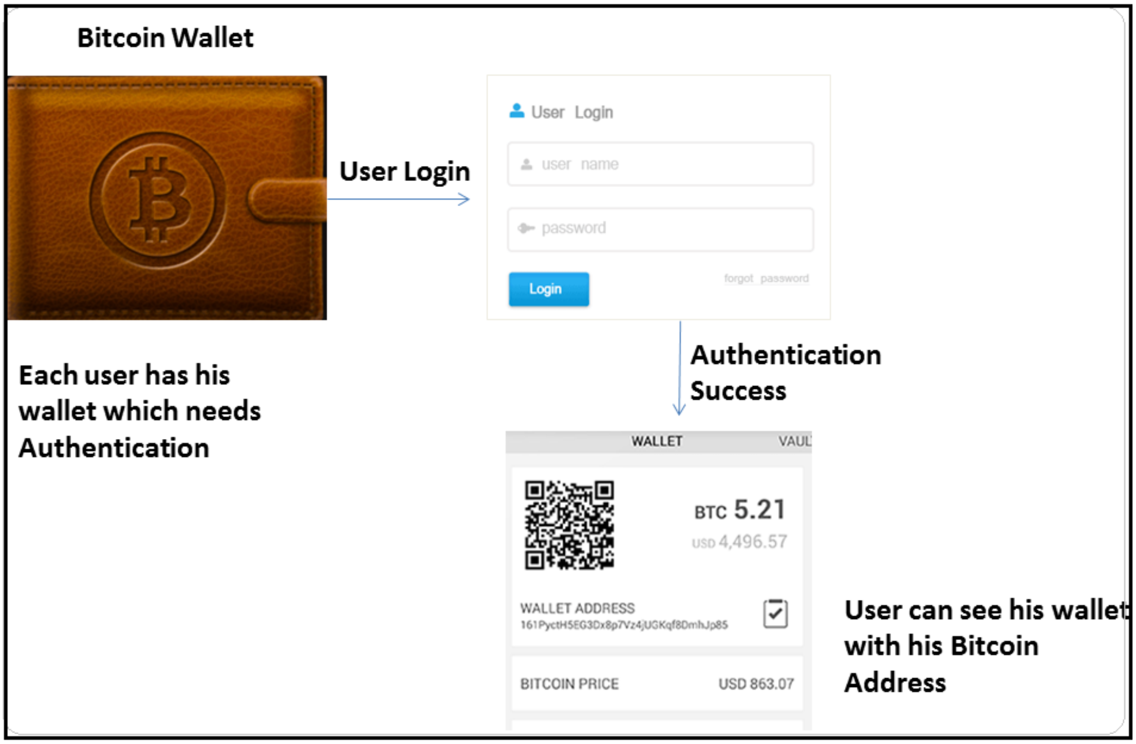
- **Step 1 : Wallet**

Every Bitcoin user has his or her own Bitcoin wallet used to send and receive bitcoins. The ownership of bitcoins is determined by the wallet, which has private and public keys as well as the bitcoin address of the wallet holder. These keys are not stored anywhere on the network but are stored locally by the user's wallet in a file system or database. These keys are not dependent on the Bitcoin protocol but are managed by each user's wallet. The sender will only be able to select the receiver of money, amount to be sent and, the transaction fees. All other validations that happen behind the scene are not visible to the actual users of Bitcoin. This helps to maintain the complex protocol of sharing ledger behind the scene while providing ease of use to every user.

Senders and receivers have unique addresses and digital signatures, which help to maintain authenticity of all the transactions that take place. Every transaction requires a valid signature to be added in the blockchain, and this signature can only be generated with a valid private key.

A bitcoin wallet must be safely protected by a password. There are many bad entities who can attempt to break passwords, hence it is a user's responsibility to keep his bitcoins safe by using a strong password.

Figure 5: Bitcoin addresses are used for transactions



In the Figure 5 we see a wallet address, which is a user's bitcoin address. This is a unique address which differentiates one user from another. Just like an email address, the bitcoin address can be shared and anyone can use it to transfer money to the current users wallet.

- **Step 2 : Private and Public Keys, Bitcoin address**

As soon as a user logs in, the wallet generates a public key, a private key, and a bitcoin address for the user. The keys usually come in pairs, as a private and public key and are generated based on the principles of public key cryptography. The private key is a number that is randomly picked . Elliptic curve multiplication, which is a one-way cryptographic function, is used to get the public key from the private key. In a similar manner, a one-way cryptographic function is used to get the bitcoin address from the public key.

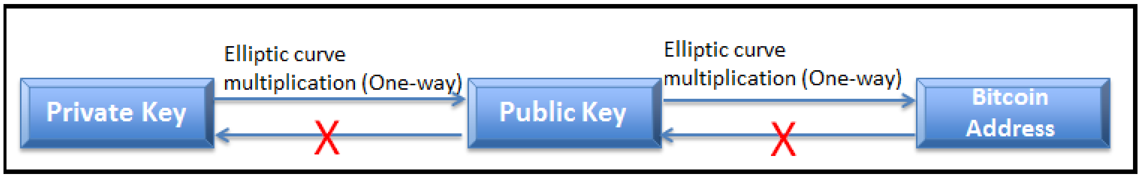
The public key is derived from the private key by using elliptic curve multiplication.

This is an irreversible operation and the formula for calculation is :

$PublicKey = PrivateKey \centerdot G$

where G is a constant point called the generator point. The reverse calculation is extremely difficult since it involves calculating discrete logarithms.

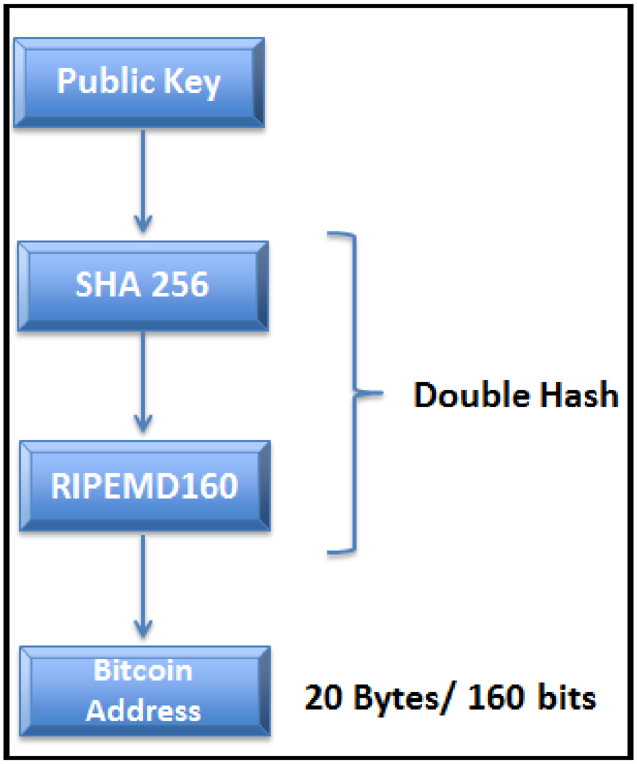
Figure 6: Bitcoin address generation algorithm



The Figure 6 shows how the bitcoin address generation works. A bitcoin address looks like $1BRrfJdB2AnWaFNgSbv3MZC2m74997JafV$. It is mainly an alphanumeric string and starts with a number. The algorithms used to generate the bitcoin address from public key are the Secure Hash Algorithm (SHA) and RACE Integrity primitives evaluation message digest(RIPEMD). The versions used are SHA256 and RIPE160. The formula used for bitcoin address calculation is :

$BitcoinAddress = RIPEMD160(SHA256(PublicKey))$

Figure 7: Bitcoin address generation involves Double Hashing

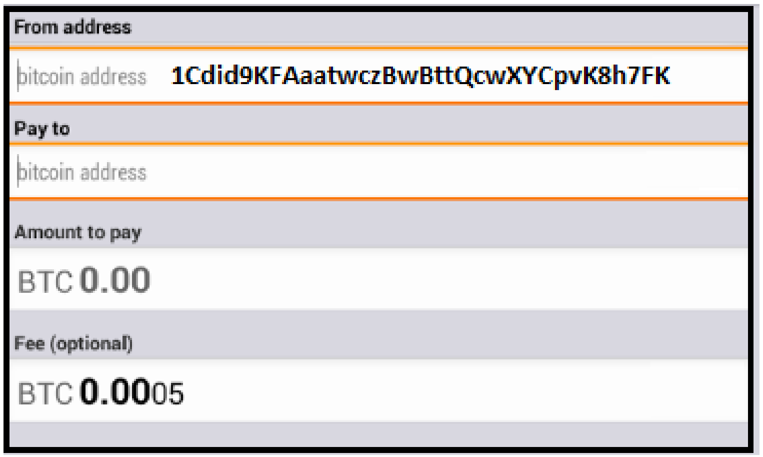


The Figure 7 depicts how the bitcoin address is generated. At this point, the bitcoin address is not known to the network. The association between the bitcoin address and a user's account starts only when the address is referenced as the recipient of a bitcoin transaction and added to the public ledger that is maintained.

- **Step 3: Send Bitcoin Functionality**

Once the user logs in, he can send bitcoins to another user's wallet. It has the sender's bitcoin address, the receiver's bitcoin address, the amount to be sent, and an optional transaction fee that will be charged to validate the transaction and get it added to the ledger. User's willingly pay the transaction fees since it ensures that their transactions are verified and if valid, added to the transaction ledger.

Figure 8: Amount transfer user interface



From address

bitcoin address 1Cdid9KFAaatwczBwBttQcwXYCpvK8h7FK

Pay to

bitcoin address

Amount to pay

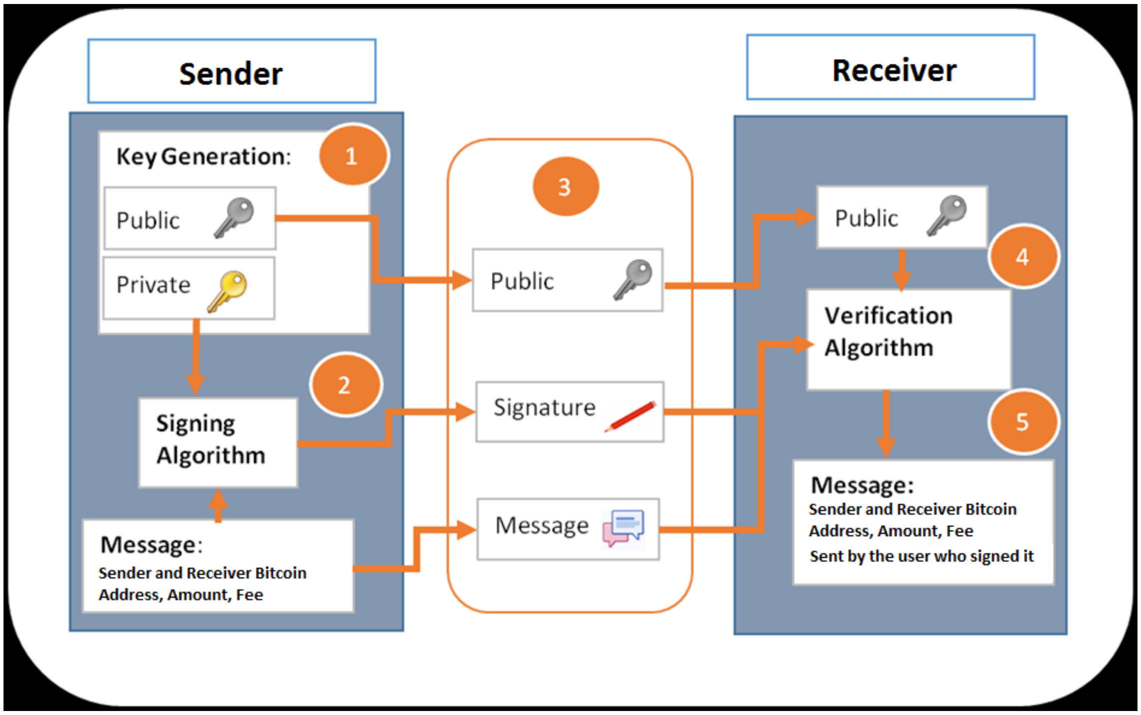
BTC 0.00

Fee (optional)

BTC 0.0005

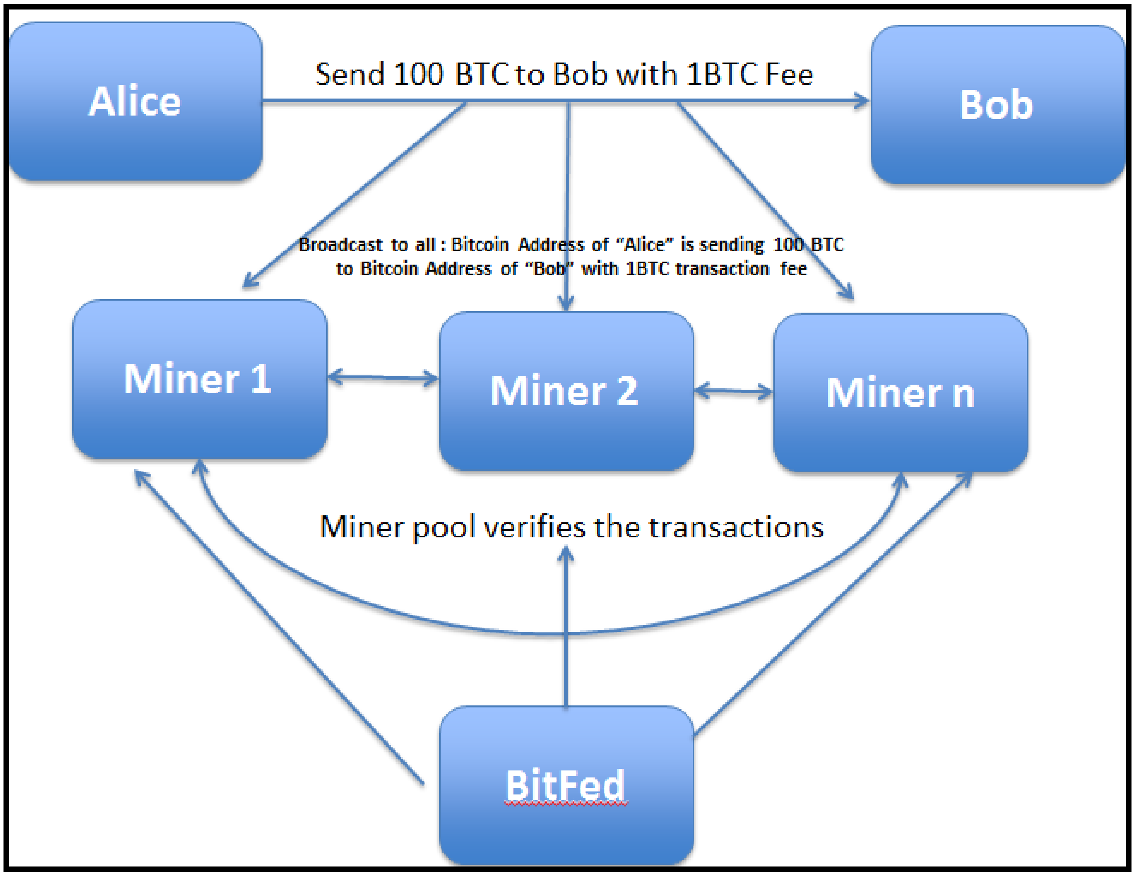
The Figure 8 depicts how a user can use the interface to transfer bitcoins. A user enters the amount to be sent in terms of bitcoins and the transaction fee that he is willing to pay. Once the user has carefully checked and made sure that the transaction amount and the receiver of the amount are correct, he can transfer the amount to the receiver. Once the user has sent the money, the message is constructed as below:

Figure 9: Message construction methodology



The message construction is shown in Figure 9. The send message tells the bitcoin network that the user, has authorized a transfer of value from his bitcoin addresses to the receiver's address. The message is broadcast to all the miners on the network announcing the transaction that Alice intends to pay Bob. As the transaction is transmitted via the peer-to-peer protocol, it quickly propagates across the network and most of the nodes in the network receive the transaction. Every 10 minutes, the miners try to gather all the transactions that have occurred in the network and try to add them to the global ledger that is maintained. The main idea behind preventing the addition of false transactions is to make the proof-of-work problem complex. The problem should be complex enough to consume a large portion of hardware resources and time, whereas it should be really easy to verify the solution to the proof-of-work problem.

Figure 10: Process of Transaction message generation


Alice
Send 100 BTC to Bob with 1BTC Fee
Bob
Broadcast to all : Bitcoin Address of "Alice" is sending 100 BTC to Bitcoin Address of "Bob" with 1BTC transaction fee
Miner 1
Miner 2
Miner n
Miner pool verifies the transactions
BitFed

- **Step 4: Building blocks of Bitcoin**

- **Transaction**

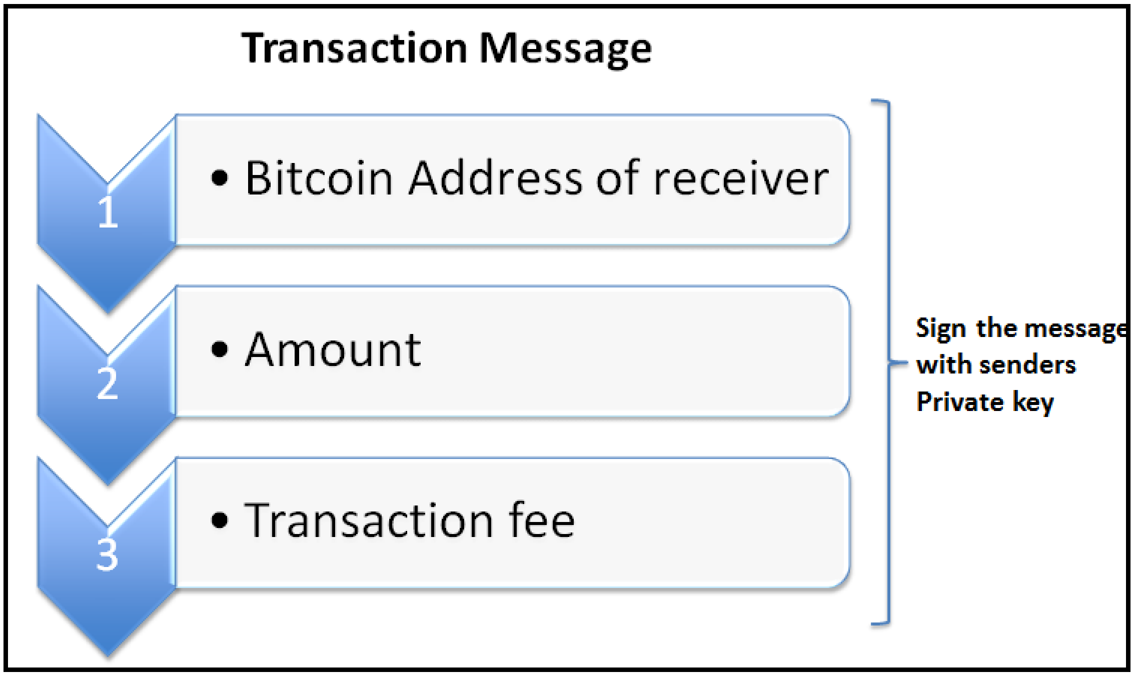
The transaction functionality of Bitcoin enables the successful transfer of bitcoins from source account to destination account. As previously described, a bitcoin address is used to identify a user's account. In order to transfer bitcoins, a transaction is created with the receiver's bitcoin address, amount to be sent and a small transaction fee for the miners who get the transaction included in the global ledger. In order to send the amount, the transaction is signed by the sender using his private key so that he cannot deny the transaction later. Instead of tabulating the balance of each account, the outputs that transferred amount to the user's account are tracked. The output is the main thing that is tracked as a part of the ledger and it has to be consistent across all the replicas maintained by the users. Each output can be claimed only once by any user. New outputs can only be created if there has been a transaction.

Output Tuple : Value in Bitcoins, Condition to spend the amount

Balance of Account = Sum of values of all unspent outputs for the account

Sum of value of Output claimed >= Sum of values of newly allocated inputs

Figure 11: BitFed transaction


Transaction Message
1
• Bitcoin Address of receiver
2
• Amount
3
• Transaction fee
Sign the message with senders Private key

Transactions are broadcast to all nodes on the network and the nodes start verifying the transaction that has taken place. This is depicted in Figure 11.

Double Spending : When a transaction takes place:

- A user may try to transfer bitcoins that he does not have
- Two transactions may be transferring the same bitcoin

The second scenario can cause an issue known as double spending. It is extremely difficult to detect double spending in a distributed setting since there is no central authority through which all the transactions happen. Transactions may be received in a different order by the network nodes, so some nodes may try to add transaction A to a ledger while others may try to add transaction B. This is a conflict that needs to be resolved in the Bitcoin protocol and it is done based on the longest chain.

- **Block**

All nodes in the network have to agree on a common order in which the transactions take place. This is difficult in a distributed setting. This issue is handled in Bitcoin by committing tentative transactions and then synchronizing them later at regular intervals by broadcasting blocks. [3] Block "b" has all the transactions "T_b" that have been committed. All other nodes on the network will spend resources to verify if the transactions are valid. If not, the committed transactions are rolled back from their local replicas until the last block that was successfully verified [3].

If there are conflicted transactions, they are discarded as orphan blocks. The successfully committed transactions cannot be rolled back. It is very difficult for anyone to forge the saved transactions since they would require more mining power than that owned by the good miners. The block creators may only decide the order in which the transactions have arrived and whether or not they are valid.

- **Block Chain**

All the blocks are arranged in a directed tree and each of them contains a reference to the previous block found. The root block is known as the "Genesis block" and it is the ancestor of all the blocks.

Distance between block "b" and genesis block is known as : Block Height

The "Blockchain Head" is the one which is furthermost from the genesis block. Transactions which are at a height less than the blockchain head are verified before the blockchain head. Miners receive bitcoins as rewards for finding blocks that can be added to the blockchain. Only miners whose blocks which are part of the blockchain are rewarded and the other miners work on adding blocks to the new blockchain head.

- **Proof-of-work problem**

Proof-of-work problem is a cryptographic puzzle that is difficult to solve but easy to verify.
Let the Blockchain be = X
Let the Block to be added be = Y
Let the additional number be = N
Find N such that the Hash (X, Y, N) = begins with α zeroes
We need to try different values for N so that the above condition is satisfied.

Adding a new block the global ledger, gives the miners the ability to create "n" new bitcoins. The way this works is : the first miner to find a solution to the proof-of-work problem, broadcasts their solution to the network of miners. The rest of the miners verify the solution and once it is accepted as the correct solution by majority of miners, the new block is added to the blockchain. The first miner thereby possesses "n" new bitcoins that are rewarded to him for the effort he has spent in solving the proof-of-work problem.

The difficulty of the problem α is adjusted periodically so that the rate of adding blocks to the global ledger is a constant that is once every ten minutes. As the number of miners in the network increases, the problem difficulty increases as well. The first miner in the Bitcoin network was assigned a total of "n = 50" bitcoins for the effort in finding the first block. The number of bitcoins rewarded is halved for every 210,000 blocks that are added. It approximates to once every four years. Due to this constraint, the number of bitcoins in the Bitcoin system can never exceed 21 million.

With time, mining is not highly profitable and hence users can decide to pay a small transaction fees to the miners as an additional incentive to make sure that the transaction is included as a part of the block chain.

- **Step 5: Process of Mining**

A transaction is created when a user tries to send an amount of bitcoins to another user. The transaction has to be verified before it can be added to a public ledger. A transaction is mainly an array of inputs and an array of outputs which are hashed using SHA256 to create a unique transaction ID. Every transaction is considered as valid only if the sum of all transaction outputs is less than the sum of all inputs.

Transactions would not be secure if they were sent directly from one user to another. Users could deny the transactions and there can be issues of double spending that arise. This issue of double spending is handled in Bitcoin by using a global, permanent ledger. Verification of transactions are done using the global ledger and new transactions can be published to the ledger. The ledger is implemented as blocks of transactions each having a hash of the previous block which thereby results in the creation of a block chain. Most miners on the network have to agree before a blockchain can be added as a part of the global ledger.

All miners can try their best to add their blockchain to the global ledger for which they first need to solve the proof-of-work problem. The first block chain that is broadcast with a valid solution to the cryptographically tough proof-of-work problem is considered as a candidate to be added to the ledger. On receiving the broadcast, all the miners will then verify if the proof-of-work is valid (just need to calculate one hash). If it is valid, the block chain can be added to the global ledger and the rest of the miners will work on finding a block that will follow the current block, and if invalid it is rejected and miners continue working to find the right solution to the problem.

2.7 Advantages of Bitcoin

- **User Anonymity :** A user's transactions are never associated with their true identity. The only thing that becomes public is his bitcoin address, which does not reveal any information about a user.

- **No 3rd party involved :** The main benefit of Bitcoin is that governments, banks, and financial institutions cannot interrupt any of the transactions since it is a decentralized system. This gives users a lot of flexibility and freedom as opposed to using any other currency.

- **No taxes on purchases :** Sales tax can be avoided since there is no third party involved in validating transactions. No Bitcoin taxation system can be put into effect since no one can map a transaction to a particular individual. Similar to cash, "under the table" transactions are possible.

- **Low transaction fees :** Wire transfers have a lot of transaction fees and exchange costs involved since there is government and third party involvement. Bitcoin is free from these exchange rates and hence the transaction fee is low.

2.8 Disadvantages of Bitcoin

- For every transaction to be included in the global ledger, one has to wait for a minimum of 10 minutes and this time increases for large amounts.

- There can be forking issues. Forks can be created accidentally or maliciously by the miner network.

- Incentives to hijack the currency has grown just as the bitcoin network has grown.

2.9 Frauds conducted using Bitcoin

- There was a Black market website called Silk Road that operated between February 2011 - October 2013 that used bitcoins for transactions.

- For Botnets, it is a source of income.

- During 2014, a virus known as "Cryptolocker" affected several victims. It extorted several million dollars from the victims. It worked by encrypting their files and then asked them for ransom in terms of bitcoins to provide the decryption key.

- Several bitcoins have been lost due to thefts.

CHAPTER 3

The BitFed Protocol

This chapter has more information on the BitFed protocol that I have implemented. A new authority called as BitFed will dictate the mining rate of new coins, thereby controlling inflation and deflation. This is a centralized agency that serves the role of :

- Determining the rate of crypto coins

Our main goal is to control deflation, which will be caused in the Bitcoin protocol due to the way the it is designed. As per the Bitcoin protocol, the maximum number of bitcoins that will be in circulation cannot exceed 21 million. [6]

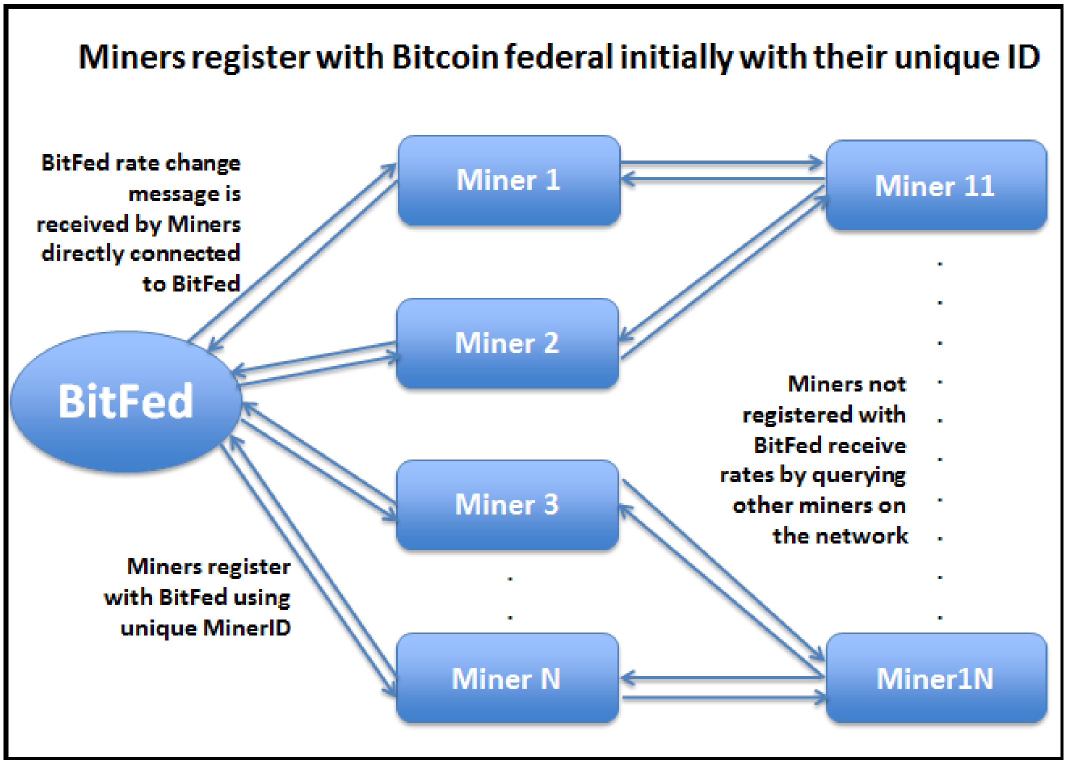
Due to this constraint, there is a high possibility that people will start hoarding bitcoins to trade them later at a higher rate. This will result in the currency becoming deflationary since bitcoins are being excessively used currently.

Our approach is to create the BitFed to manage the money supply and curb the deflation issues. The BitFed is similar in design to the US Federal reserve. It is an independent agency that acts as a gatekeeper of the Bitcoin economy and helps to promote the stability of prices to make sure people transact bitcoins and do not hoard them.

3.1 Most Miners register with BitFed

The first step in this cryptocurrency is that miners register with the BitFed.

Figure 12: Miners register with BitFed


Miners register with Bitcoin federal initially with their unique ID
BitFed rate change message is received by Miners directly connected to BitFed
BitFed
Miner 1
Miner 2
Miner 3
Miner N
Miner 11
Miner1N
Miners not registered with BitFed receive rates by querying other miners on the network
Miners register with BitFed using unique MinerID

The registration process is as shown in Figure 12. This ensures that most of the miners receive the current currency rate applicable. Registering with BitFed is optional; miners who distrust the central authority could learn the exchange rate from other miners instead.

3.2 BitFed broadcasts rate change messages on regular basis

The BitFed broadcasts the rate to be used and the time from when the new rate should be effective.

Figure 13: Rate change broadcast message by BitFed

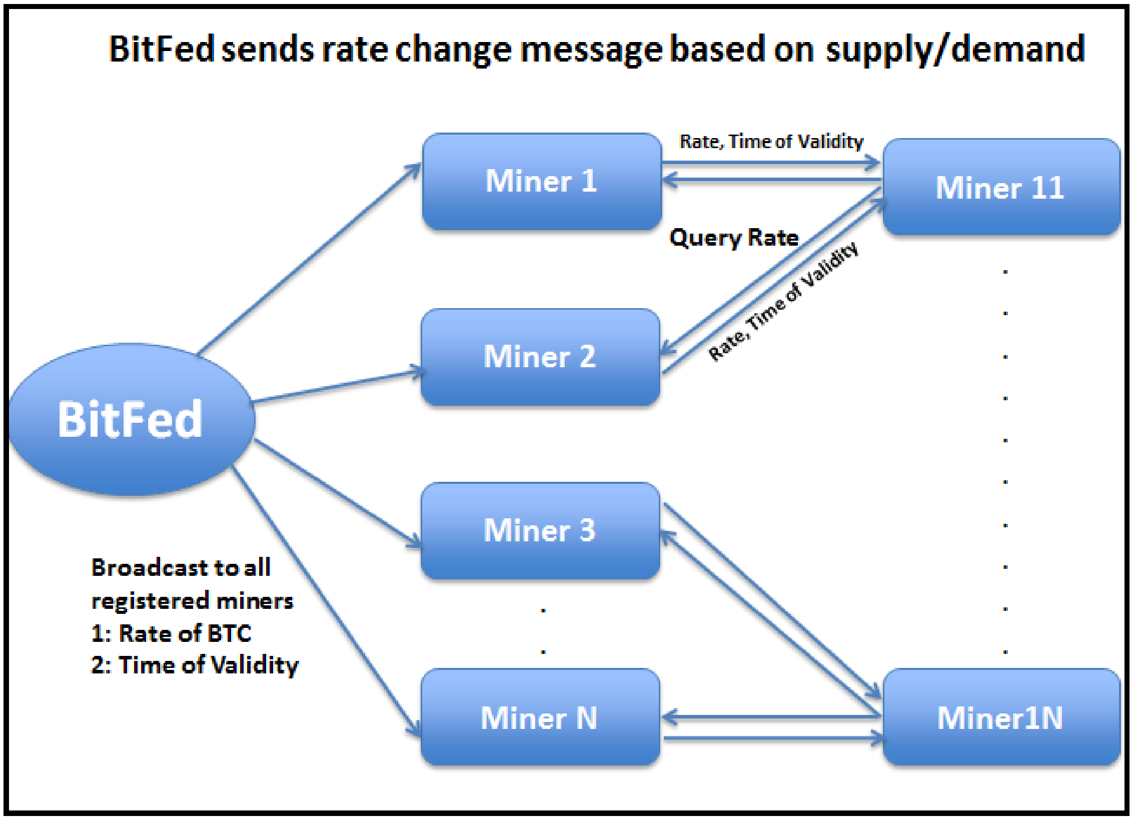


Figure 13 depicts the handling of rate change message. Most miners will start using the new rate as soon as possible while some may try to cheat and use the older rate till the duration of time that it is effective. The rate is manually determined by the BitFed based on currency supply of bitcoins needed.

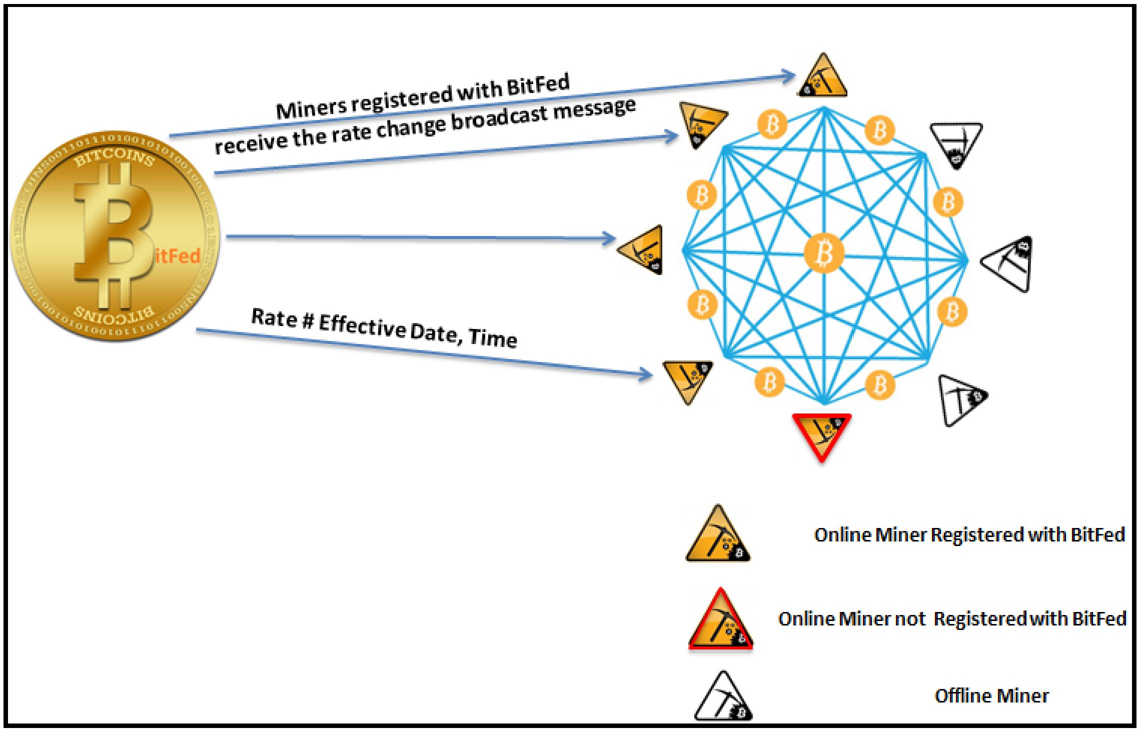
3.3 BitFed rate decision factors

BitFed decides the rate at which bitcoins should be transacted. The rate depends on several factors such as number of bitcoins currently in the system, average time being taken to solve the proof of work problem, supply and demand. Based on these criteria the federal decides the rate of transaction and gives miners a 24 hour window to supply the new rate. BitFed broadcasts the transaction rate to all miners effective immediately from the time the message is received.

3.4 Scenario 1: Miner is online and registered with BitFed when the BitFed broadcasts a new rate

- Most miners in the network register with the BitFed using their bitcoin addresses.
- If registered, the miners can receive broadcast messages from the BitFed immediately once it is sent without any delays.

Figure 14: Miner online and registered with BitFed when BitFed sends new rate


BITCOINS
itFed
Miners registered with BitFed
receive the rate change broadcast message
Rate # Effective Date, Time
Online Miner Registered with BitFed
Online Miner not Registered with BitFed
Offline Miner

3.5 Scenario 2: Miner is online but not registered with BitFed when the BitFed broadcasts a new rate

- Some miners in the network may not register with the BitFed for different reasons such as not trusting a central authority.

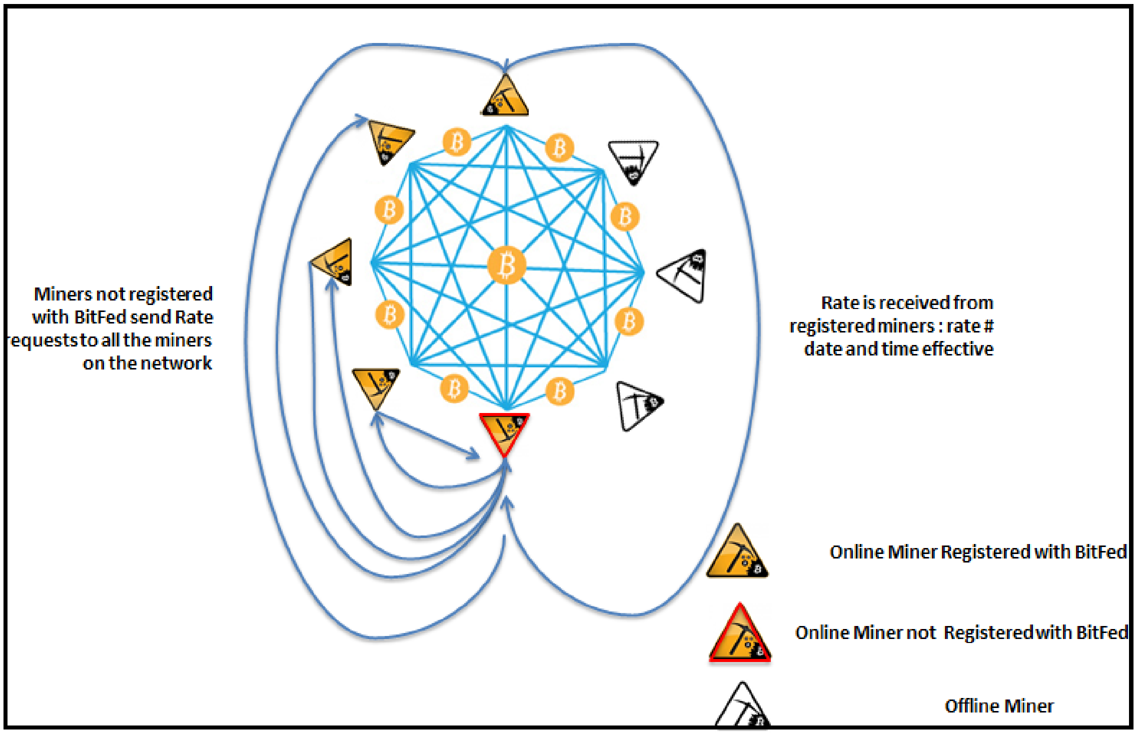
- If not registered the miners can query the other miners in the network to receive the current rate of transaction.

- A miner who needs the rate broadcasts a request for current rate and receives a reply from most of the online nodes on the network.

- There may be a conflict with the rate received from the miner network. He uses the rate that is being used by most of the miners in the network assuming that more than 51 percent of miners are good and will use the current right rate.

- Since the non-registered miners receive broadcast messages from the miners on the network there can be delays in receiving the rate as opposed to the registered miners receiving it directly from the BitFed.

Figure 15: Miner online but not registered with BitFed and BitFed sends new rate

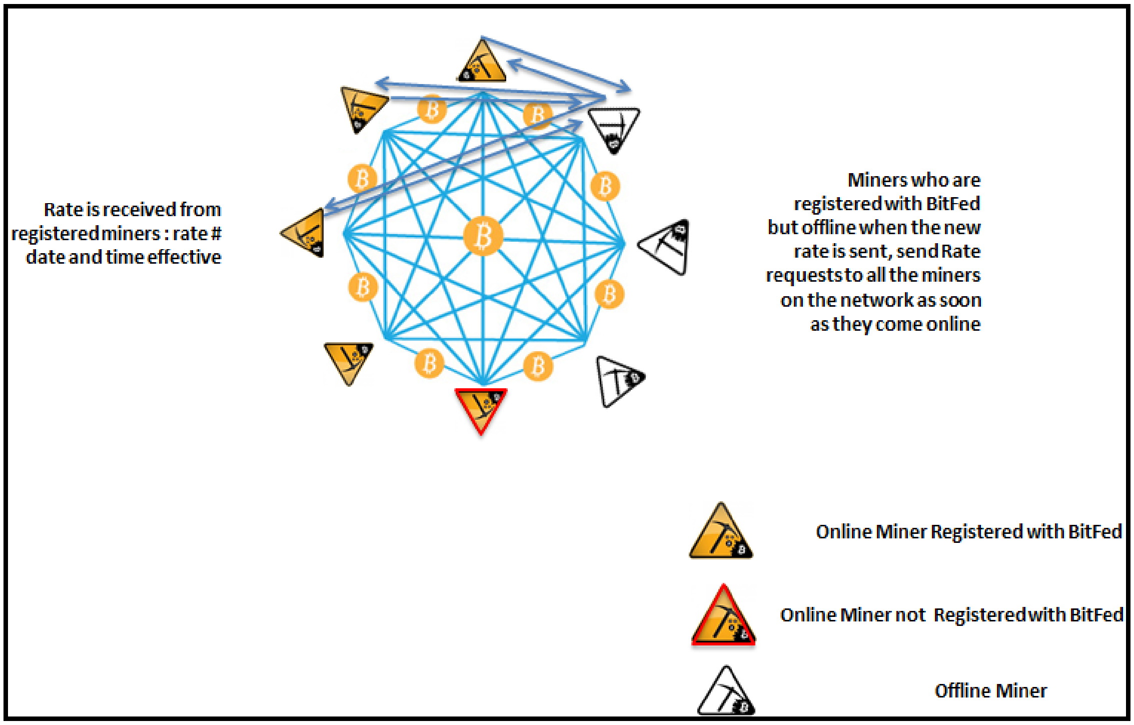

Miners not registered
with BitFed send Rate
requests to all the miners
on the network
Rate is received from
registered miners : rate #
date and time effective
Online Miner Registered with BitFed
Online Miner not Registered with BitFed
Offline Miner

3.6 Scenario 3: Miner is offline and misses rate broadcast sent by BitFed

- It is possible that a miner is connected to the BitFed but is offline at the time the BitFed sends the broadcast message.
- The miner thereby misses the rate change message from the BitFed.
- The miner who has missed the rate change message, broadcasts a request for the current rate and receives a reply from most of the online nodes on the network.
- There may be a conflict with the rate received from the miner network. He uses the rate that is being used by most of the miners in the network assuming that more than 51 percent of miners are good and will use the current right rate.

- Since the miners who have missed the rate change message receive broadcast messages from the miners on the network, there can be delays in receiving the rate as opposed to the registered miners receiving it directly from the BitFed.

Figure 16: Miner offline and misses BitFed rate message


Rate is received from registered miners : rate # date and time effective
Miners who are registered with BitFed but offline when the new rate is sent, send Rate requests to all the miners on the network as soon as they come online
Online Miner Registered with BitFed
Online Miner not Registered with BitFed
Offline Miner

3.7 Scenario 4: Miner is online and registered with BitFed but ignores the rate change message from the BitFed

- It is possible that the miner is a cheater and does not want to use the rate change message immediately.

- On receiving the rate change message, he saves the rate and time it will be effective and continues using the older rate while it is still valid. This way he is able to make more profit than what he could using the new rate.

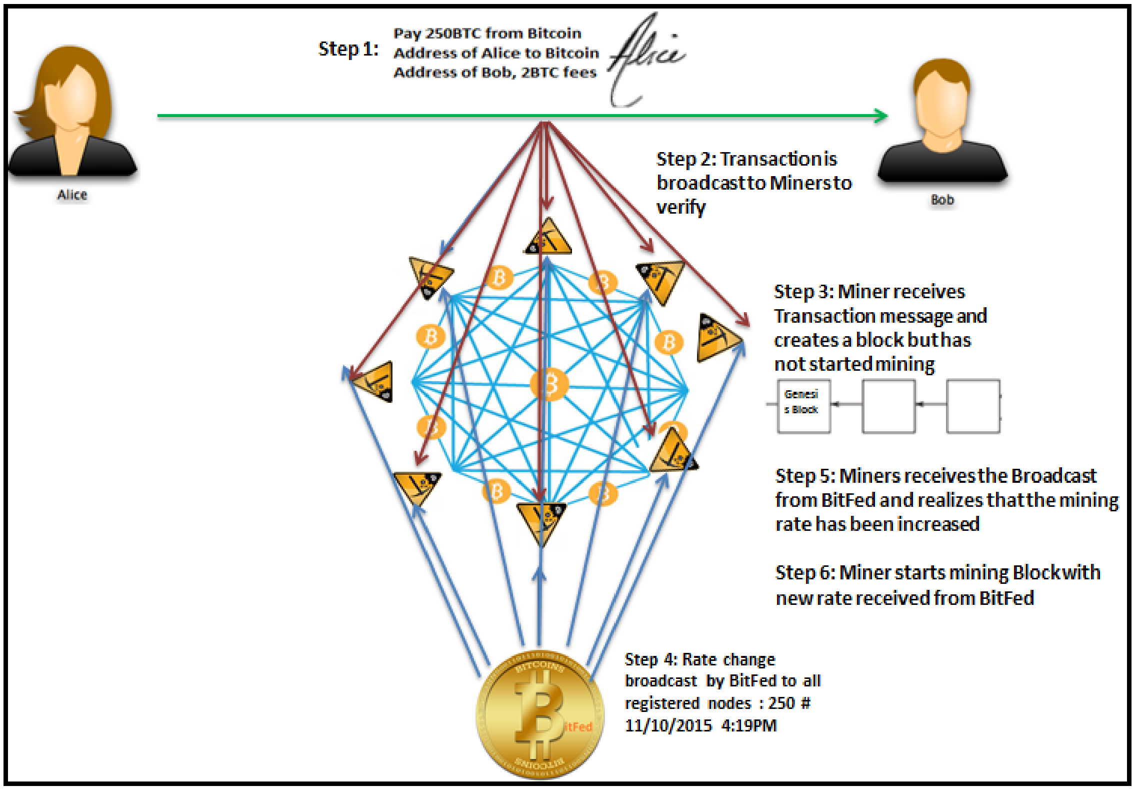
3.8 Scenario 5: BitFed sends rate change message several times within the same day

- It is possible for the BitFed to send the rate change message more than once on the same day.
- Rate change message has Rate, Time attached.
- This will help the miners in determining the latest rate and the effective time.
- The BitFed gives users a small window to ensure that all the broadcast messages are propagated to miners who are not connected to the BitFed directly.
- Good miners tend to use the new rate immediately whereas cheaters try their best to postpone using the new rate and use the new rate at the end.

3.9 Scenario 6: Block computation not started and rate increase message received

- A miner has not started computing the proof-of-work for a block and receives rate change broadcast where the rate has been increased.
- If the miner is a good miner : He will try to use the new rate as soon as possible.
- If the miner is a bad miner : He will try to use the new rate immediately as it is highly profitable for the miner.

Figure 17: Rate increase message received before miner starts computing block


Step 1:
Pay 250BTC from Bitcoin Address of Alice to Bitcoin Address of Bob, 2BTC fees
Alice
Alice
Bob
Step 2: Transaction is broadcast to Miners to verify
Step 3: Miner receives Transaction message and creates a block but has not started mining
Genesis Block
Step 5: Miners receives the Broadcast from BitFed and realizes that the mining rate has been increased
Step 6: Miner starts mining Block with new rate received from BitFed
Step 4: Rate change broadcast by BitFed to all registered nodes : 250 # 11/10/2015 4:19PM
BitFed

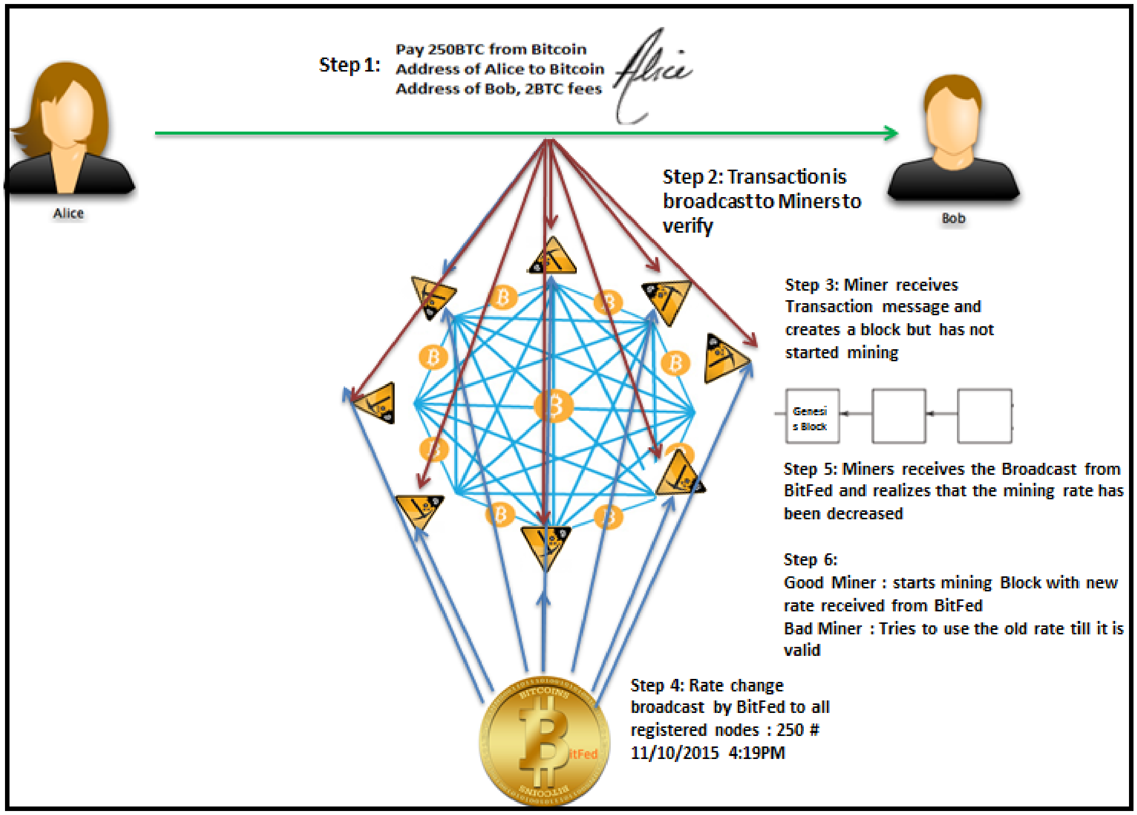
3.10 Scenario 7: Block computation not started and rate decrease message received

- A miner has not started computing the proof-of-work for a block and receives rate change broadcast where the rate has been decreased.

- If the miner is a good miner : He will try to use the new rate as soon as possible.

- If the miner is a bad miner : He will continue to use the old rate since it is higher and he will get more profit out of it. The miner has a small window to use the old rate and he tries to make maximum use of this window. However, he runs the risk that his transactions could be rejected.

Figure 18: Rate decrease message received before miner starts computing block


Step 1:
Pay 250BTC from Bitcoin Address of Alice to Bitcoin Address of Bob, 2BTC fees
Alice
Bob
Step 2: Transactionis broadcastto Minersto verify
Step 3: Miner receives Transaction message and creates a block but has not started mining
Genesi s Block
Step 5: Miners receives the Broadcast from BitFed and realizes that the mining rate has been decreased
Step 6:
Good Miner : starts mining Block with new rate received from BitFed
Bad Miner : Tries to use the old rate till it is valid
Step 4: Rate change broadcast by BitFed to all registered nodes : 250 # 11/10/2015 4:19PM
itFed

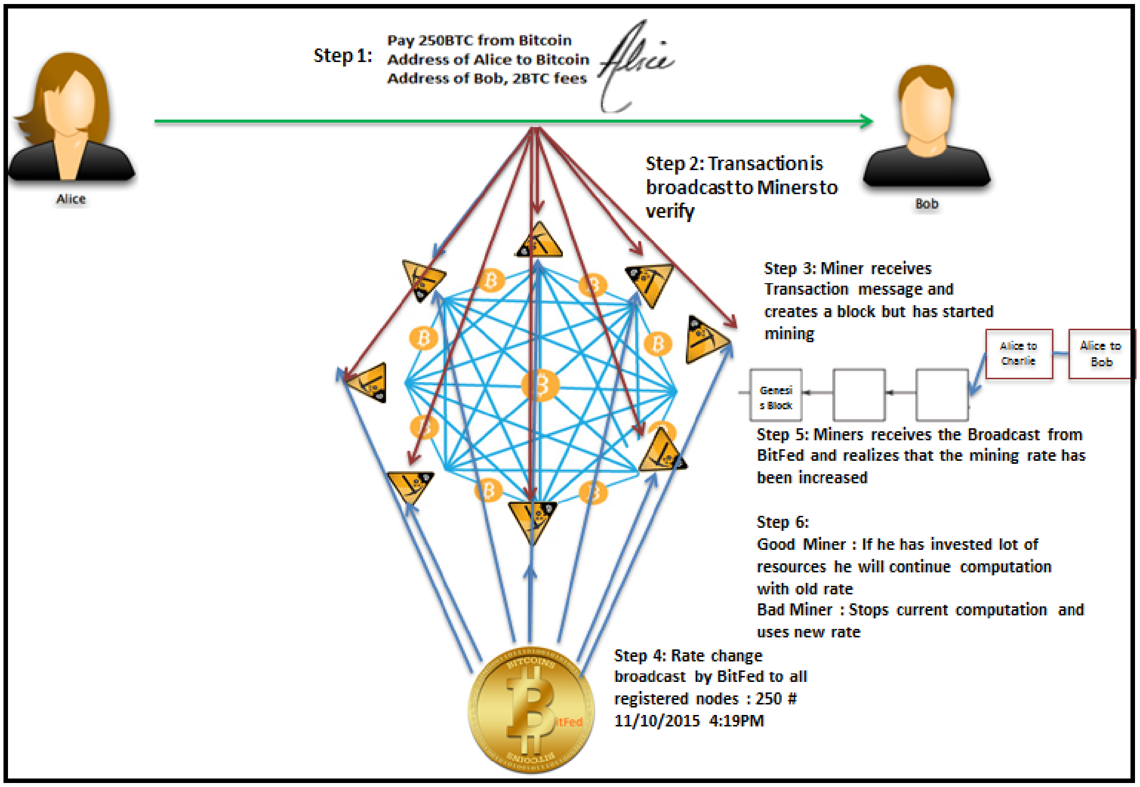
3.11 Scenario 8: In the middle of block computation the rate increase message received

- A miner has started computing the proof-of-work for a block and receives rate change broadcast where the rate has been increased.

- If the miner is a good miner : If the amount of computational resources he has used towards mining is less, he will stop the computation in the middle and start using the new rate. If he has invested a lot of time on mining the block he will complete the current block and use the new rate.

- If the miner is a bad miner : A bad miner stops the computation in middle and starts computing the block again with a new rate.

Figure 19: Rate increase message received when miner is in the middle of computing a block


Step 1:
Pay 250BTC from Bitcoin Address of Alice to Bitcoin Address of Bob, 2BTC fees
Alice
Alice
Bob
Step 2: Transactionis broadcastto Minersto verify
Step 3: Miner receives Transaction message and creates a block but has started mining
Alice to Charlie
Alice to Bob
Genesis Block
Step 5: Miners receives the Broadcast from BitFed and realizes that the mining rate has been increased
Step 6:
Good Miner : If he has invested lot of resources he will continue computation with old rate
Bad Miner : Stops current computation and uses new rate
Step 4: Rate change broadcast by BitFed to all registered nodes : 250 # 11/10/2015 4:19PM
BITCOINS
BitFed

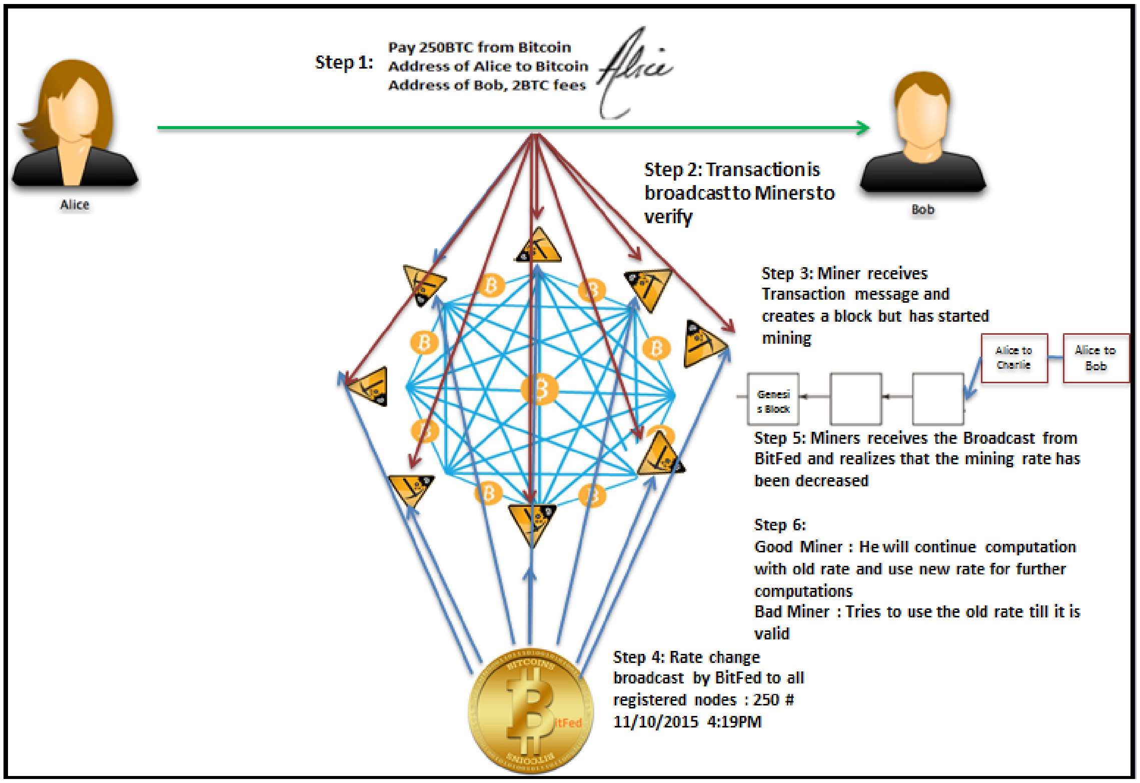
3.12 Scenario 9: In the middle of block computation the rate decrease message received

- A miner has started computing the proof-of-work for a block and receives rate change broadcast where the rate has been decreased.

- If the miner is a good miner : He may complete the current block and start using the decreased bitcoin rate from the next block. This will ensure him the profit he makes on the current block.

- If the miner is a bad miner : He will continue to use the old rate since it is higher and he will get more profit out of it. The miner has a 24 hour window to use the new rate and he tries to make the maximum use of this window and uses the rate till the end time of validity.

Figure 20: Rate decrease message received when miner is in the middle of computing a block


Step 1:
Pay 250BTC from Bitcoin
Address of Alice to Bitcoin
Address of Bob, 2BTC fees
Alice
Alice
Bob
Step 2: Transaction is broadcast to Miners to verify
Step 3: Miner receives Transaction message and creates a block but has started mining
Alice to Charlie
Alice to Bob
Genesis Block
Step 5: Miners receives the Broadcast from BitFed and realizes that the mining rate has been decreased
Step 6:
Good Miner : He will continue computation with old rate and use new rate for further computations
Bad Miner : Tries to use the old rate till it is valid
Step 4: Rate change broadcast by BitFed to all registered nodes : 250 # 11/10/2015 4:19PM
BITCOINS
BitFed

3.13 Scenario 10: Two miners broadcast proof-of-work calculation simultaneously with same difficulty but at different rates

- Consider a situation wherein two miners simultaneously broadcast blocks that they have computed. The bitcoin rates used for both the computations are different.

- The effort spent in computing the two blocks and the difficulty in computing the blocks is almost the same.

- The existence of two blocks, creates a fork in the block chain as shown below.

- The last block before there is a fork is Block Q. Due to forking, some of the miners add Block R.1 to their ledger and others add R.2 to the ledger.

- This is a major issue that needs to be addressed since a common ledger is one of the main goals of Bitcoin protocol.

- First step in this process would be to see if the rate at which the blocks have been computed is still valid or not i.e if it is in the 24 hour window or not.

- If the rate is not valid, then one of the blocks is immediately rejected.

- If the rate is valid, the process waits for the next block to be added to either R.1 or R.2 with two different parts of the network working on each.

- Suppose that the miners using the block R.2 for their ledger successfully encode a new block S before the miners working on block R.1

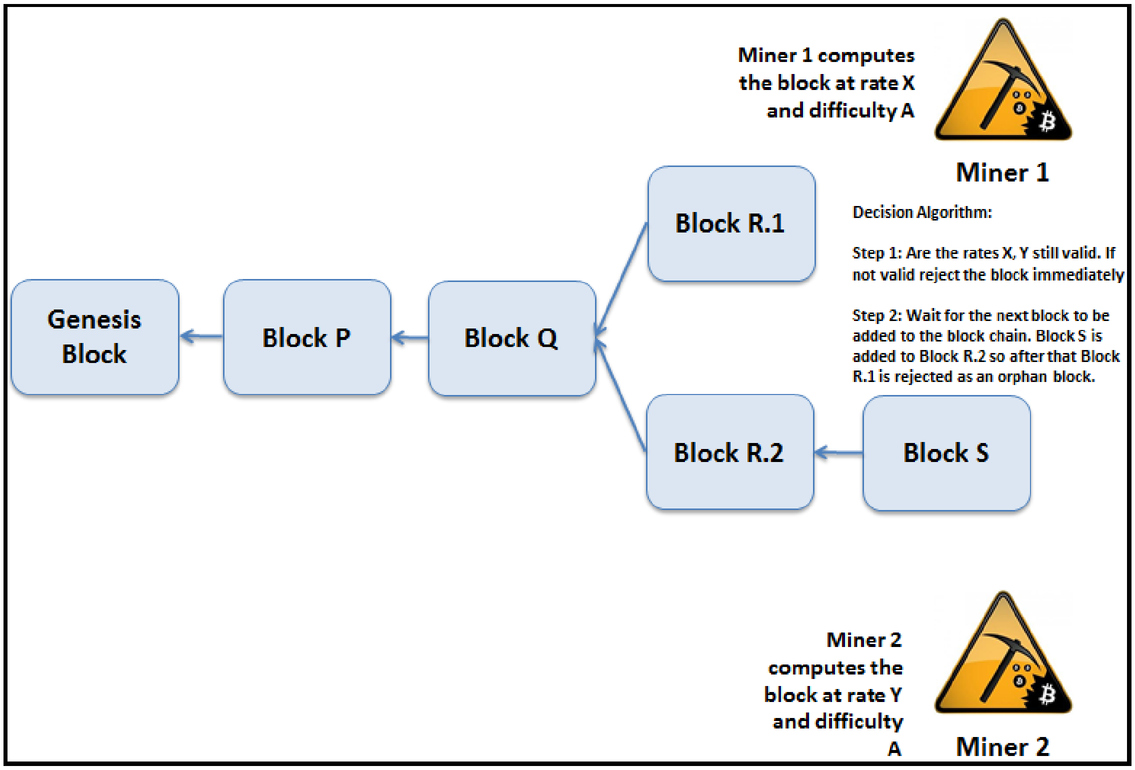
- In that case, the network will accept the chain with blocks O, P, Q, R.2, S as the main ledger because it it required higher computational effort and has higher difficulty. The block R.1 is marked as an orphan.

- With time, more blocks seal a transaction on the block chain so that there is more certainty that the transactions are correct and that the transactions will remain encoded in the main ledger.

- An attacker who desires to modify the previous encoded blocks in the block chain will have to recompute all the targeted blocks in the block chain as well as subsequent blocks.

- The cumulative difficulty involved in modifying blocks in the ledger are statistically impossible to achieve.

Figure 21: Two miners broadcast block chain at the same time with same difficulty in computing the blocks


Miner 1 computes the block at rate X and difficulty A
Miner 1
Genesis Block
Block P
Block Q
Block R.1
Block R.2
Block S
Miner 2 computes the block at rate Y and difficulty A
Miner 2

Decision Algorithm:

Step 1: Are the rates X, Y still valid. If not valid reject the block immediately

Step 2: Wait for the next block to be added to the block chain. Block S is added to Block R.2 so after that Block R.1 is rejected as an orphan block.

3.14 Scenario 11: Two miners broadcast proof-of-work calculation simultaneously with different difficulty and at different rates

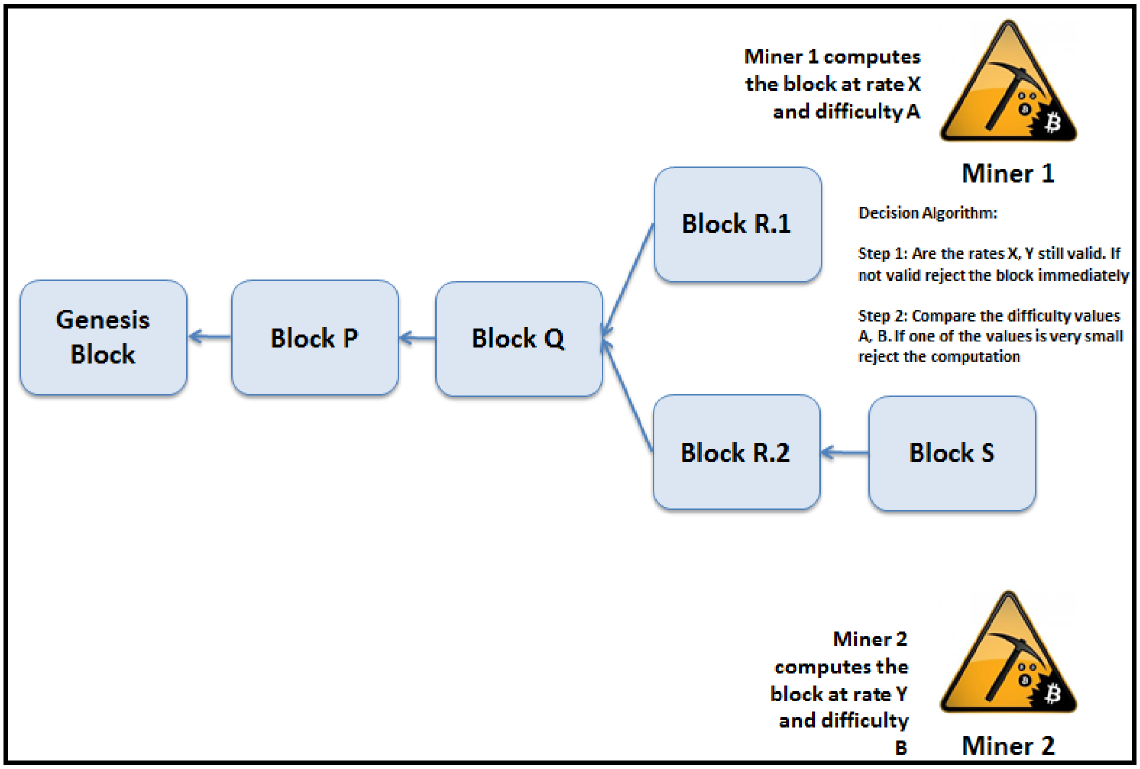
- Consider a situation wherein two miners simultaneously broadcast blocks that they have computed. The bitcoin rates used for both the computations are different.

- The effort spent in computing the two blocks and the difficulty in computing block A is easier than computing block B.

- The existence of two blocks creates a fork in the block chain as shown below. The miners choose block B as the main block in the ledger since the computational effort is more for Block B so it is coming from the majority miner pool.

- This is like a voting mechanism wherein miners can vote as to which block should be included in the block chain. Based on majority votes the blocks are included in the block chain.

Figure 22: Two miners broadcast block chain at the same time with different difficulty in computing the blocks


Miner 1 computes the block at rate X and difficulty A
Miner 1
Decision Algorithm:
Step 1: Are the rates X, Y still valid. If not valid reject the block immediately
Step 2: Compare the difficulty values A, B. If one of the values is very small reject the computation
Genesis Block
Block P
Block Q
Block R.1
Block R.2
Block S
Miner 2 computes the block at rate Y and difficulty B
Miner 2

3.15 Scenario 12: Miner network receives a block with rate X at 12:01 AM where as the last valid time for rate X was 11:59 PM

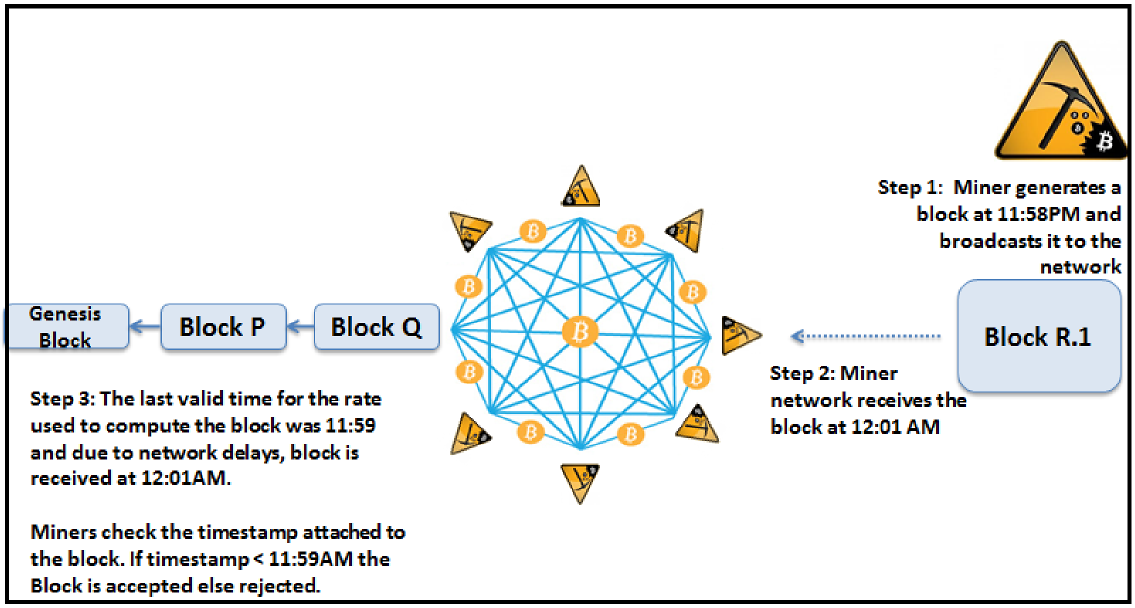
- Consider a situation wherein the miners in the network receive a block with proof-of-work computed with rate X at 12:01AM. The last valid time for rate X was 11:59PM.

- It is possible that the miner had broadcast the block at 11:59PM when the rate X was still valid but due to network delays, the block was received at 12:01AM.

- The proof of work has time stamp attached to it. This is the time at which the block was computed. If the timestamp is less than 11:59PM and the block is received before 12:05AM, the block is accepted by the miner network for processing.

- We assume that even with network delays, at least one of the miners can receive the transactions within the time frame of five mins.

Figure 23: Miner network receives the block < five minutes after the rate expires



Step 1: Miner generates a block at 11:58PM and broadcasts it to the network

Step 2: Miner network receives the block at 12:01 AM

Step 3: The last valid time for the rate used to compute the block was 11:59 and due to network delays, block is received at 12:01AM.

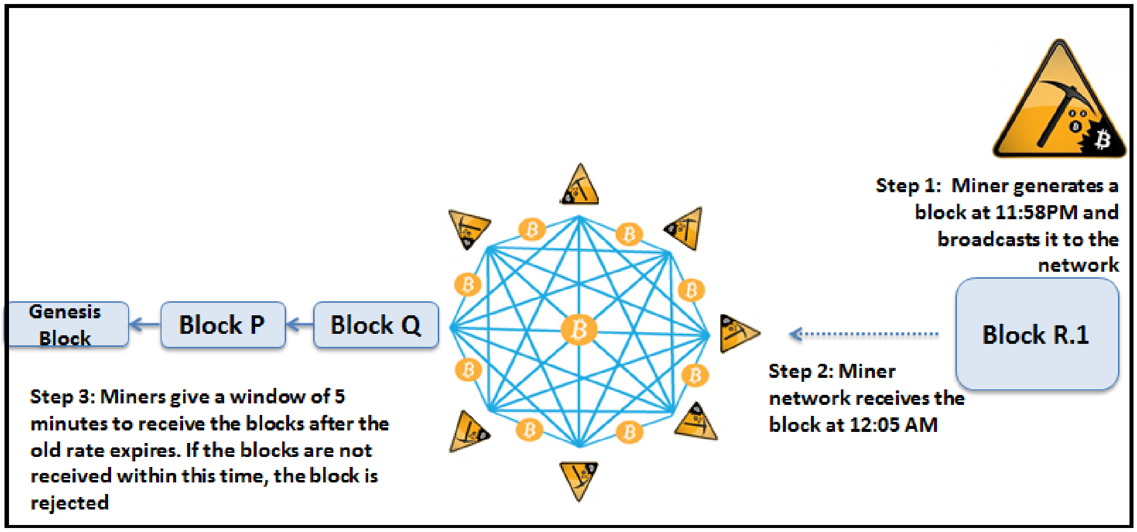
Miners check the timestamp attached to the block. If timestamp < 11:59AM the Block is accepted else rejected.

3.16 Scenario 13: Miner network receives a block with rate X after 12:05 AM where as the last valid time for rate X was 11:59 PM

- Consider a situation wherein the miners in the network receive a block with proof-of-work computed with rate X after 12:05AM. The last valid time for rate X was 11:59PM.

- As per our previous discussion, the block is rejected and not added as a part of the global ledger since we assume that even with network delays, the block should be received by at-least one of the miners within the time duration of 12:05AM.

Figure 24: Miner network receives the block > five minutes after the rate expires


Step 1: Miner generates a block at 11:58PM and broadcasts it to the network
Block R.1
Step 2: Miner network receives the block at 12:05 AM
Genesis Block
Block P
Block Q
Step 3: Miners give a window of 5 minutes to receive the blocks after the old rate expires. If the blocks are not received within this time, the block is rejected

3.17 Scenario 14: Miner is a cheater and tries to change the timestamp of a block and then broadcast it

- Timestamp is based on the time the block is computed by the miner to add to the block chain
- A cheater may try to change the timestamp of the block to get it included in the global ledger
- It T2 is less than T1 : Reject the block. Cannot process a block with timestamp lesser than a block already in the block chain. Else accept the block.

3.18 Scenario 15: There are two histories of transactions and forks keep growing equally

- There are two chains with a shared genesis block and are identical up until the forking point, after which they exist exclusively in parallel unless one is abandoned thereby creating two separate networks.

- Consider a user Alice who frequently uses the bitcoins for transactions. Coins in Alice's possession before the fork remain Alices's on both chains after the fork and both chains agree on those transactions since they were all before the fork.

- After the fork, each transaction continues in parallel on separate chains.

- It would be expected that each chain and the coins on each of those chains have unique names.

CHAPTER 4

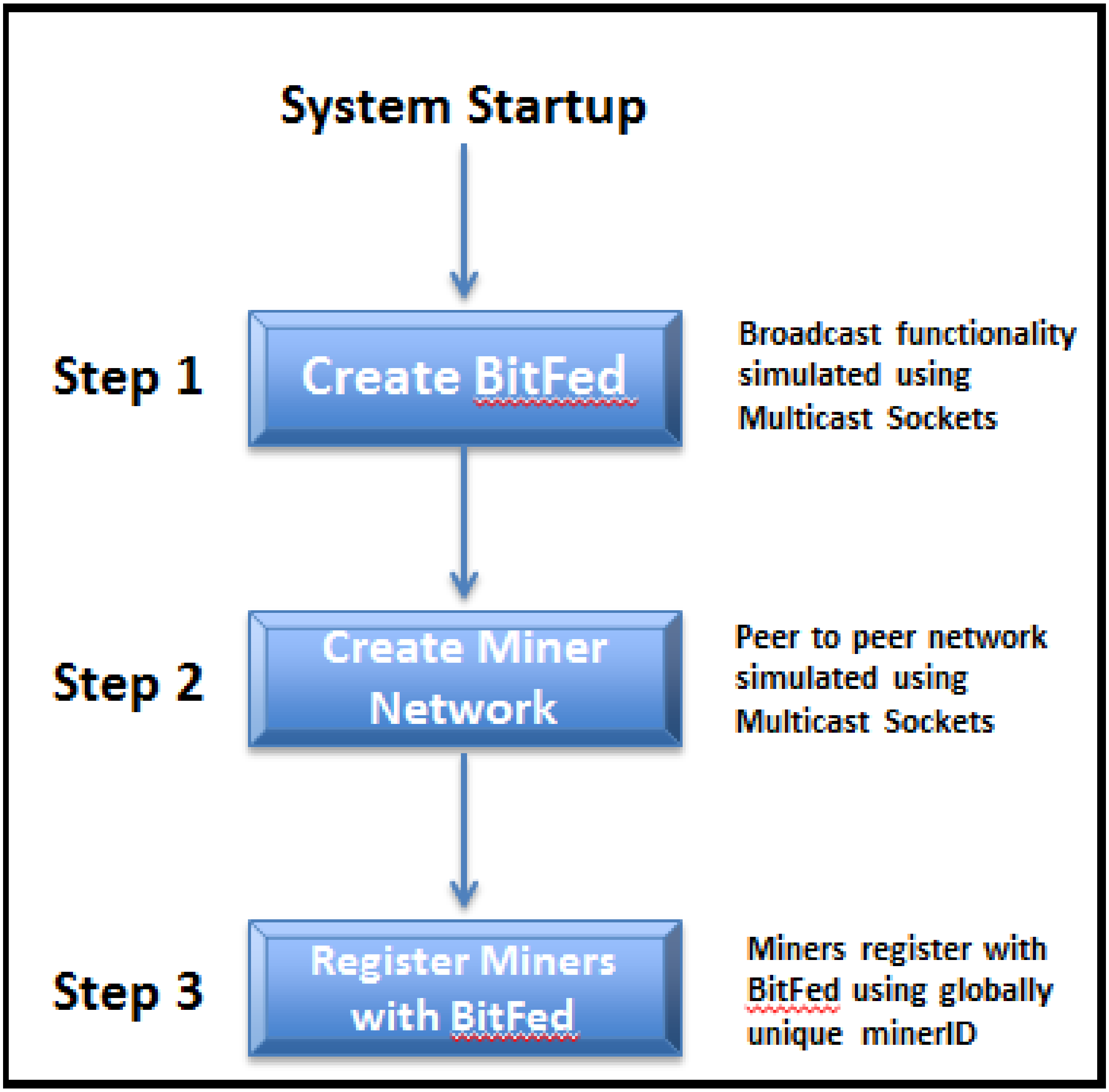
Design and Implementation of the Simulator

This chapter has more information on the protocol design and implementation of the simulator.

4.1 Step 1 : System Startup

System startup is the first thing that happens which initializes the BitFed and the miner network. All the required private and public keys as well as bitcoin addresses are initialized in this step.

Figure 25: System Startup


System Startup
Step 1
Create BitFed
Broadcast functionality simulated using Multicast Sockets
Step 2
Create Miner Network
Peer to peer network simulated using Multicast Sockets
Step 3
Register Miners with BitFed
Miners register with BitFed using globally unique minerID

The algorithm is used to initialize the centralized BitFed. It will use the socket ID (currently set to 4447) to send the rate change broadcasts to all the miners that are directly registered with BitFed. The miner ID's are also stored by the BitFed when the miners register with the BitFed for the very first time.

Algorithm 1 BitFed Initialization

```
Set Group address for Multicast Socket communication to a number between 224.0.0.0 to 239.255.255.255. In my code I have used the IP as 230.0.0.1
Create a new Multicast socket with any Socket ID. In my project I have used Socket ID as 4447.
Retrieve Inet Address from Group Address.
The newly created Multicast socket now joins the group of the Inet Addess : socket.joinGroup(address).
Initialize the Old BitFed rate file. Initially on system startup this file is populated with the old transaction rate which was used by BitFed. Until a new rate is received from the BitFed, all the miners use the Old rate for the transactions.
```

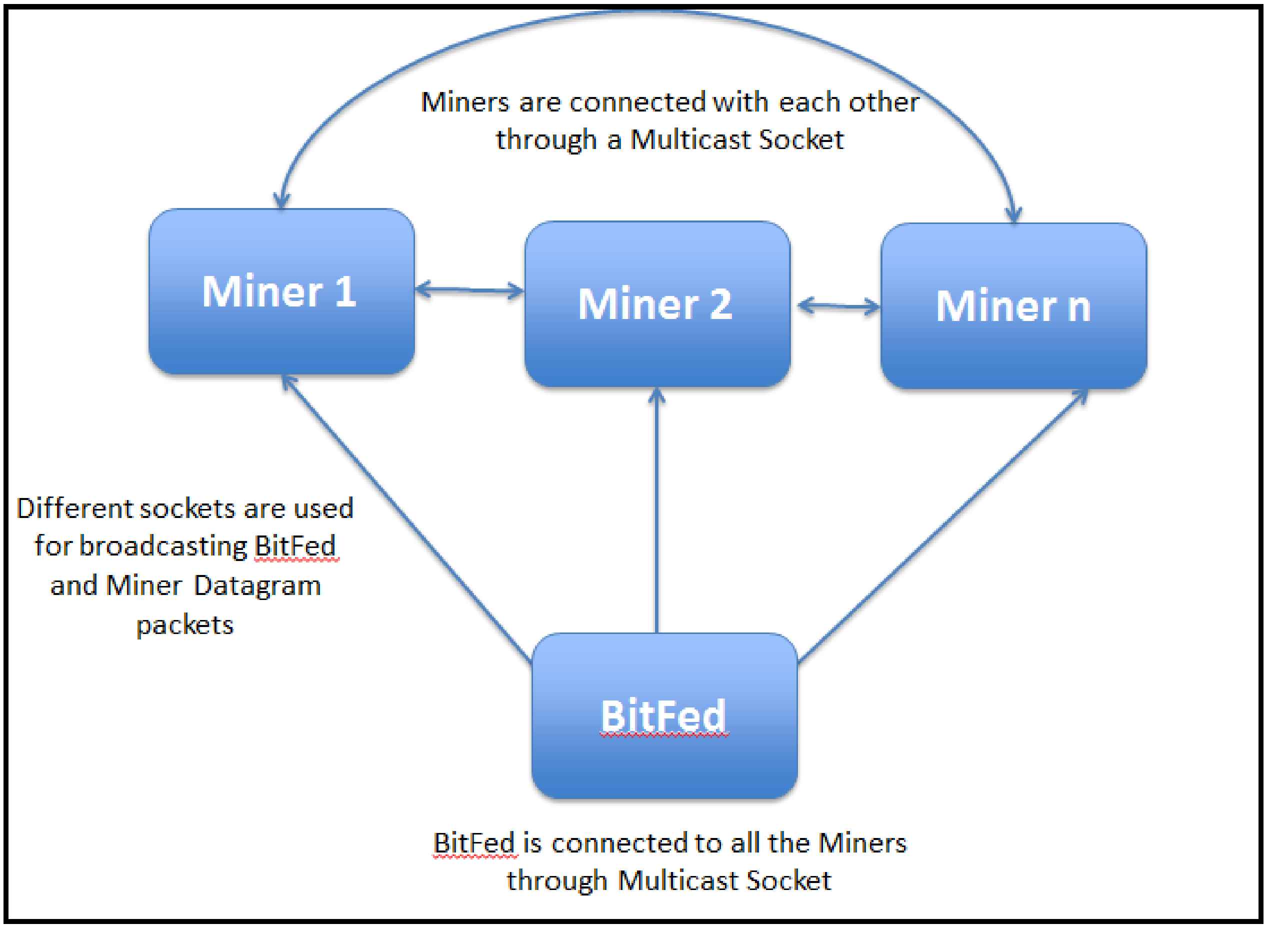
Algorithm 2 Miner Initialization

```
Set Group address for Multicast Socket communication to a number between 224.0.0.0 to 239.255.255.255. In my code I have used the IP as 230.0.0.1
Input is the Miner ID which will be used to registered with BitFed.
Create 3 new Multicast sockets with Socket ID. In my project I have used the following Socket ID = 4446, 4447, 4450.
Retrieve Inet Address from Group Address.
The newly created Multicast sockets now join the group of the Inet Addess : socket.joinGroup(address). This is the address of the broadcast group. The socket numbers are the numbers on which the Miners will listen. Each socket has a specific functionality.
    Socket 4446 : To listen to transactions generated.
    Socket 4447 : To listen to Broadcasts by BitFed.
    Socket 4450 : To send the computed Blocks to all the miners on the network.
For each miner identified by miner ID start a thread which will listen on all the sockets to receive and send data.
```

The algorithm is used to create all the miners in the network who will verify the transactions taking place. The current implementation has a network of six miners. All the miners are directly registered with the BitFed and receive broadcasts from the BitFed on socket ID which was created initially i.e. 4447. The miners have two other sockets which they use to receive transactions and to share the computed block chain with other miners to verify.

The Miners and BitFed are connected as shown below. A Multicast socket helps them in communicating with each other.

Figure 26: Miner and BitFed network setup


Miners are connected with each other through a Multicast Socket
Miner 1
Miner 2
Miner n
Different sockets are used for broadcasting BitFed and Miner Datagram packets
BitFed
BitFed is connected to all the Miners through Multicast Socket

Algorithm 3 Miner registration with BitFed

```
for $i = 0$ to $NumberofMiners$ do
    Add miner ID to a list.
    Send the miner ID list to BitFed to store the miners who are registered directly.
end for
```

The algorithm enables miners to register with the BitFed. Each miner has a globally unique ID with that it registers with the BitFed. This ID will be used later to send broadcasts to all the registered miners about the rate change messages.

4.2 Step 2 : Broadcast BitFed rate change message

In this step, the BitFed broadcasts the new rate to be used by miners for transactions and also the time from when the new rate is effective. This allows miners a small window to adhere to the new rate, after which their blockchains will be rejected by the rest of the miners on the network if they are not using the new rate.

Algorithm 4 Broadcast rate change message

```
BitFed has intelligent economists who manually calculate the new rate based on
   current supply and demand statistics.
Set the new rate for mining and the time from when it is effective.
Create a timer task and broadcast the rate change message on a regular basis.
Datagram packet is created to send the rate change message to all the nodes on
   the socket.
```

4.3 Step 3 : Transaction, Block, and Block Chain

The figure shows the class diagram for the main components of Bitcoin : Transaction, Block, and Block Chain. These main components comprise every transaction.

Figure 27: Transaction, Block Class diagram

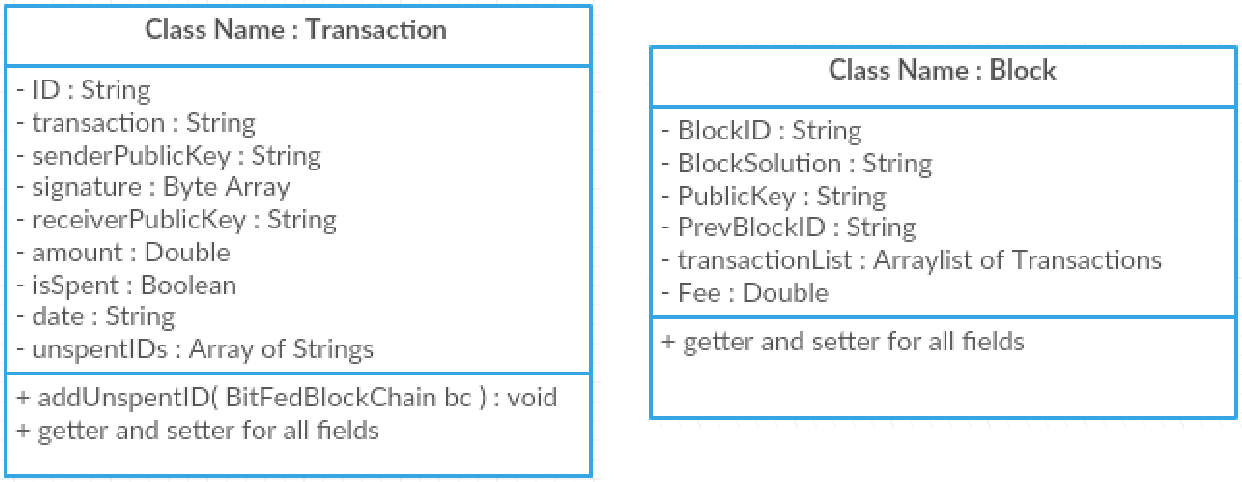
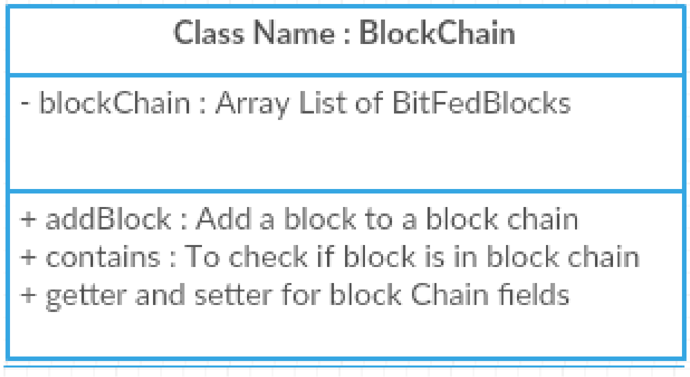


Figure 28: Block chain class diagram



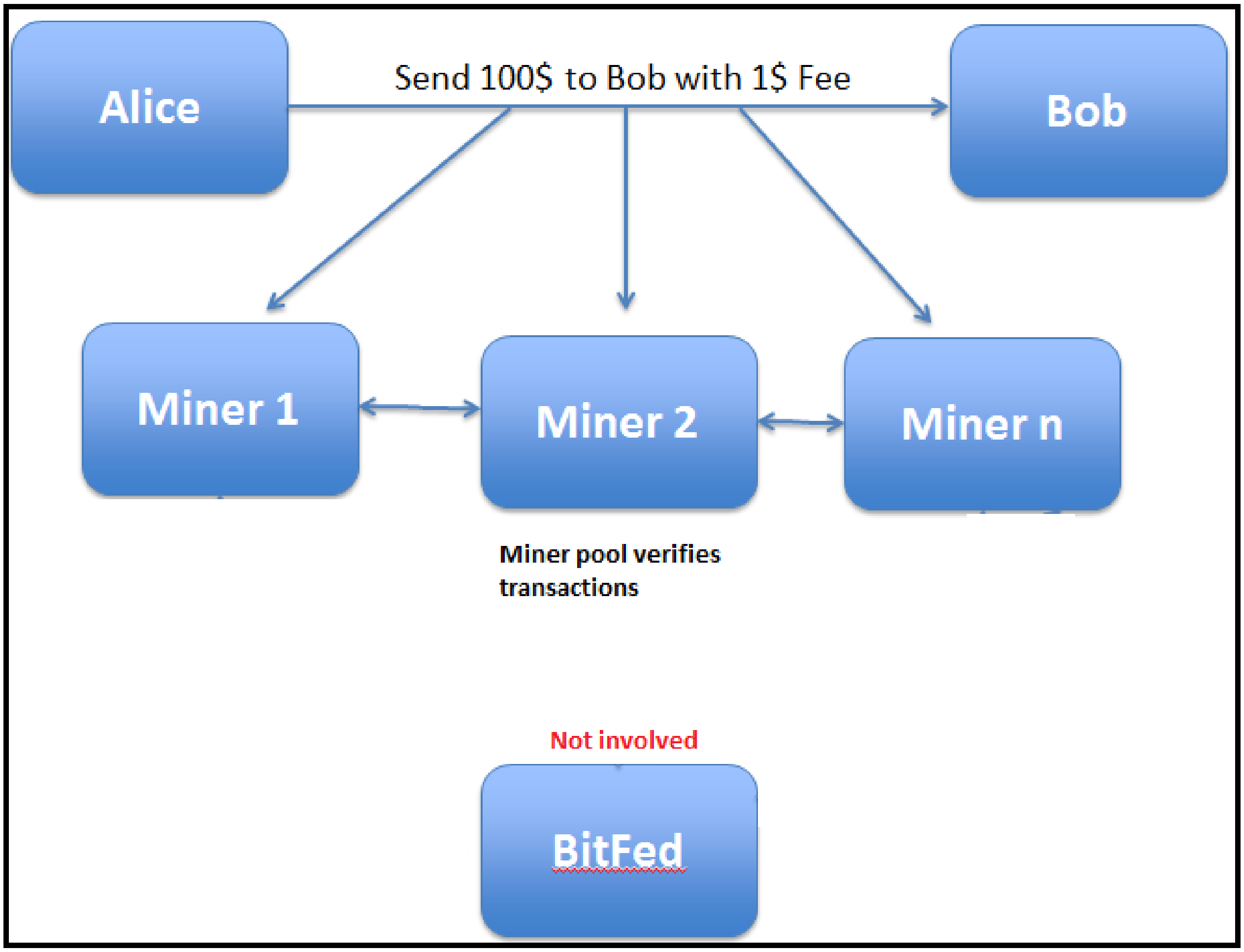
Class Name : BlockChain

- blockChain : Array List of BitFedBlocks

+ addBlock : Add a block to a block chain
+ contains : To check if block is in block chain
+ getter and setter for block Chain fields

Once the network is in place the next thing is to show details of how a transaction takes place on the network.

Figure 29: Transaction from Alice to Bob


Alice
Send 100$ to Bob with 1$ Fee
Bob
Miner 1
Miner 2
Miner n
Miner pool verifies transactions
Not involved
BitFed

Algorithm 5 Create and Broadcast a Transaction

```
Select the receiver of bitcoins
Select the amount to be sent to receiver and small transaction fee for the miners.
Sign the transaction message with the senders private key.
Broadcast the message to the miner network.
Datagram packet is created to send the transaction message to all the miners on the network.
```

4.4 Step 4 : Miners receive BitFed Message

The miners receive a message from BitFed, they use the new rate for processing.

Algorithm 6 Rate change algorithm

```
Miners are divided into good miners and cheaters. Good miners use the correct rate and cheaters try to cheat by using the higher rate.
Miners broadcast messages to be added to the block chain.
Based on the rate used for mining and time the transaction is broadcast, the other miners verifying the transaction decide whether to accept the transaction or not.
There is a vote file maintained which has the votes by different miners on the network. It indicates the number of miners who have accepted a block chain.
Based on the majority decision, the transaction is either accepted to be added to the block chain and if not it is rejected.
```

4.5 Step 5 : Miners process transaction

Once the miners receive a transaction the miners process the transaction to solve the proof of work problem and try their best to get the transaction included in the block chain.

Algorithm 7 Miners process transaction

```
Every transaction is added to a block as an unconfirmed transaction.
Miners work on solving the proof of work problem.
If they find a solution to be valid and the transaction to be valid, they add the transaction to the block chain.
If transaction is invalid it is rejected and miners continue to work on a new transaction.
Once a transaction is added as a part of the block chain it is broadcast to the rest of the miner network.
The miner network accepts / rejects the transaction. If accepted, all the miners synchronize their block chain.
```

4.6 Step 6 : Algorithm to solve proof-of-work problem

Once the miners receive a transaction the miners process the transaction to solve the proof of work problem. The algorithm to solve the proof of work problem is as shown.

Algorithm 8 Proof-of-work problem solving

```
Convert the block into a string. This string will be used to compute the proof-of-work..
Create a random string and concatenate it to the block string and a long number which acts as a Nonce and is increased every time.
If a hash of this concatenation contains a number of zeroes equal to N, solution is found and it can be added to a block chain and shared with the rest of the miners.
```

Algorithm 9 Query BitFed rate

```
Create a miner object.
Every time a miner object is created, the miner queries the BitFed in order to get the current rate being used and the time from when this is effective.
The BitFed returns a string representation of the rate and the time, which can be immediately used by the miners for the transactions that they process.
When the miner is offline and then comes back online, it queries the other miners about the rate being used.
```

Algorithm 10 Synchronize blockchain

```
When a block chain is received from other miners, the block chain has to be added to the local replica.
There is a vote file which is being maintained. It contains the Block Chain ID and the number of miners who have accepted the block chain.
Every miner first verifies if the transactions in the blockchain are valid. If they are valid it then checks the number of votes received for the block chain.
If the block chain has a valid number of votes, the block chain is added to the local replica and the number of votes is increased by 1.
```

CHAPTER 5

Experimentation and results

This chapter includes the results of the experiments and test cases run in this project. As discussed before, the final goal of this project is to develop a BitFed simulator which is patterned on Bitcoin protocol and can control deflation in the currency.

5.1 System Specifications

- Operating System Used : Windows 7
- Language and Environment: Java 1.7.02 , JavaSE −1.7
- System memory and Core: 8GM RAM and 64-bit Operating System

5.2 Environment Setup

5.2.1 IDE used

The IDE's used for development are Eclipse Kepler and NetBeans 8.0.1

5.2.2 JAR files

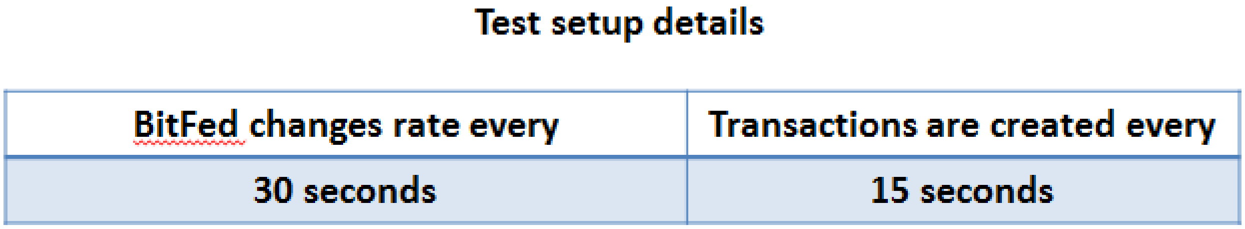
Several JAR files need to be added for the cryptographic algorithms to work appropriately

- Apache Commons
- Codec under Apache Commons
- Common Lang3

- Commons Lang3 Version 3.4

5.3 System Testing

For the system testing, we have several test setups



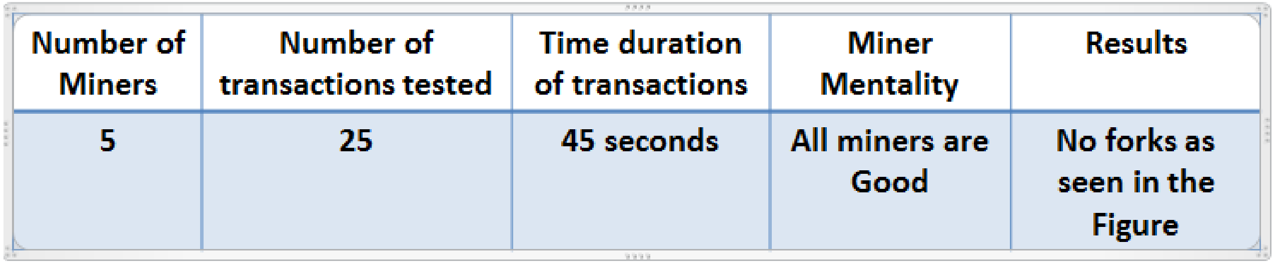
Test setup details

BitFed changes rate every	Transactions are created every
30 seconds	15 seconds

5.3.1 All miners are good and there are no cheaters

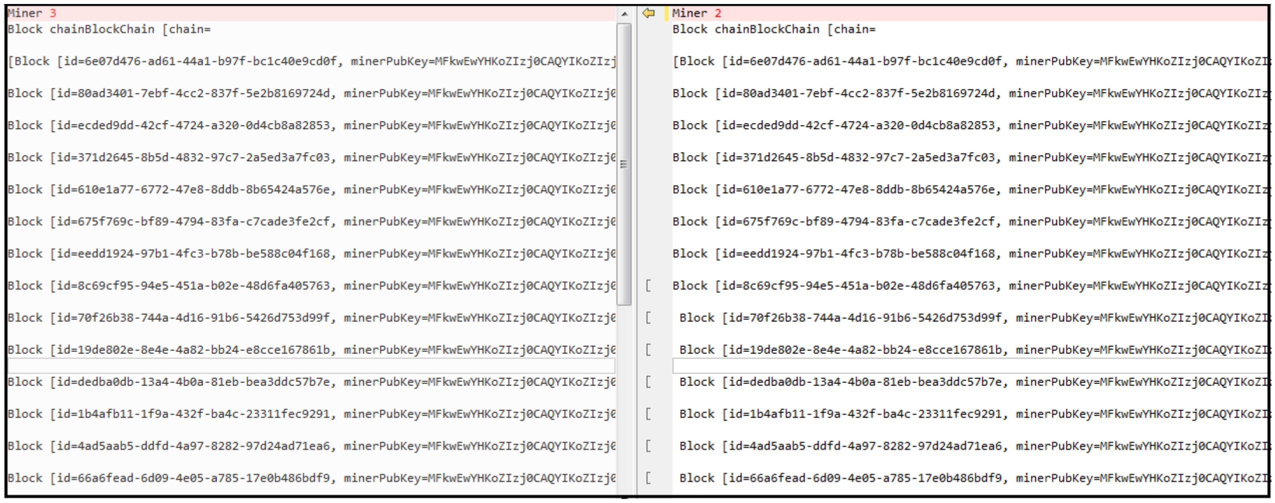
- This system test case is when all the miners are good and there are no cheaters.
- BitFed sends the rate change message once every 30 seconds.
- The testing is done for 25 transactions which approximately take 45 seconds to execute.
- On test completion there is no fork observed in the transaction chain.
- We have shown below the comparison of two block chains of different miners and we observe that all the transaction ID's in the block chain are the same implying that the ledger is in sync.

Figure 30: Test Settings : All miners are good and there are no cheaters



Number of Miners	Number of transactions tested	Time duration of transactions	Miner Mentality	Results
5	25	45 seconds	All miners are Good	No forks as seen in the Figure

Figure 31: Test Results : All miners are good and there are no cheaters



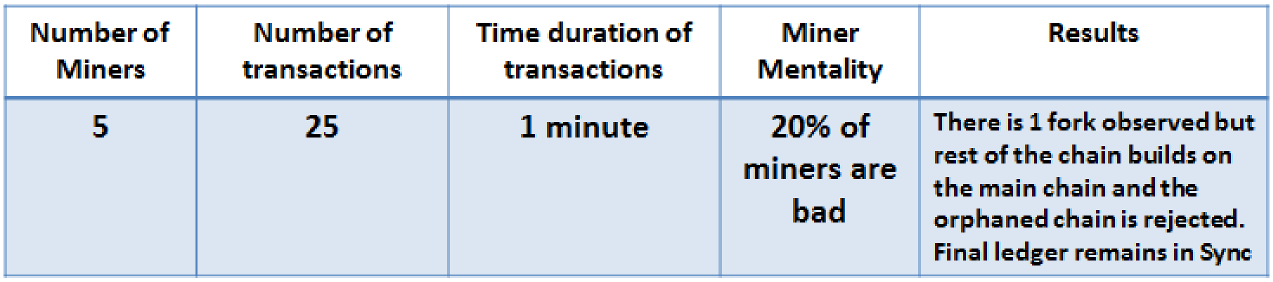
```
Miner 3
Block chainBlockChain [chain=

[Block [id=6e07d476-ad61-44a1-b97f-bc1c40e9cd0f, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj

Block [id=80ad3401-7ebf-4cc2-837f-5e2b8169724d, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=ecded9dd-42cf-4724-a320-0d4cb8a82853, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=371d2645-8b5d-4832-97c7-2a5ed3a7fc03, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=610e1a77-6772-47e8-8ddb-8b65424a576e, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=675f769c-bf89-4794-83fa-c7cade3fe2cf, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=eedd1924-97b1-4fc3-b78b-be588c04f168, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=8c69cf95-94e5-451a-b02e-48d6fa405763, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=70f26b38-744a-4d16-91b6-5426d753d99f, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=19de802e-8e4e-4a82-bb24-e8cce167861b, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=dedba0db-13a4-4b0a-81eb-bea3ddc57b7e, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=1b4afb11-1f9a-432f-ba4c-23311fec9291, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=4ad5aab5-ddfd-4a97-8282-97d24ad71ea6, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=66a6fead-6d09-4e05-a785-17e0b486bdf9, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
```

```
Miner 2
Block chainBlockChain [chain=

[Block [id=6e07d476-ad61-44a1-b97f-bc1c40e9cd0f, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZI

Block [id=80ad3401-7ebf-4cc2-837f-5e2b8169724d, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=ecded9dd-42cf-4724-a320-0d4cb8a82853, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=371d2645-8b5d-4832-97c7-2a5ed3a7fc03, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=610e1a77-6772-47e8-8ddb-8b65424a576e, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=675f769c-bf89-4794-83fa-c7cade3fe2cf, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=eedd1924-97b1-4fc3-b78b-be588c04f168, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

[ Block [id=8c69cf95-94e5-451a-b02e-48d6fa405763, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

[  Block [id=70f26b38-744a-4d16-91b6-5426d753d99f, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZI

[  Block [id=19de802e-8e4e-4a82-bb24-e8cce167861b, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZI

[  Block [id=dedba0db-13a4-4b0a-81eb-bea3ddc57b7e, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZI

[  Block [id=1b4afb11-1f9a-432f-ba4c-23311fec9291, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZI

[  Block [id=4ad5aab5-ddfd-4a97-8282-97d24ad71ea6, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZI

[  Block [id=66a6fead-6d09-4e05-a785-17e0b486bdf9, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZI
```

5.3.2 20 percent of miners are cheaters

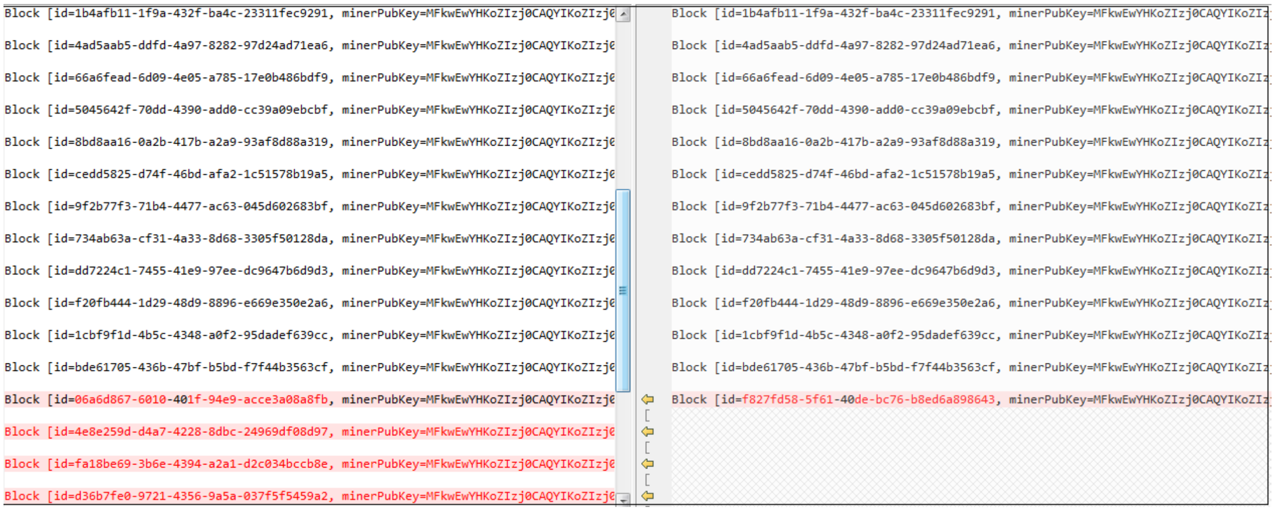
- This system test case is when 20 percent of the miners are good and rest are cheaters.
- BitFed sends the rate change message once every 30 seconds.
- The testing is done for 25 transactions which approximately take 60 seconds to execute.
- On test completion there is one fork observed in the transaction chain. The rest of the network builds on top of one of the block chains and thereby the orphaned chain is rejected.
- We have shown below the comparison of two block chains of different miners and we observe that all the transaction ID's in the block chain are the same except for the end. At the end we can see that one of the chain has grown longer and has more blocks in it. Thereby the orphaned chain is not used by rest of the network and everyone builds on top of the longest chain.

Figure 32: Test Settings : 20 percent of miners are cheaters



Number of Miners	Number of transactions	Time duration of transactions	Miner Mentality	Results
5	25	1 minute	20% of miners are bad	There is 1 fork observed but rest of the chain builds on the main chain and the orphaned chain is rejected. Final ledger remains in Sync

Figure 33: Test Results : 20 percent of miners are cheaters



```
Block [id=1b4afb11-1f9a-432f-ba4c-23311fec9291, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
Block [id=4ad5aab5-ddfd-4a97-8282-97d24ad71ea6, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
Block [id=66a6fead-6d09-4e05-a785-17e0b486bdf9, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
Block [id=5045642f-70dd-4390-add0-cc39a09ebcbf, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
Block [id=8bd8aa16-0a2b-417b-a2a9-93af8d88a319, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
Block [id=cedd5825-d74f-46bd-afa2-1c51578b19a5, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
Block [id=9f2b77f3-71b4-4477-ac63-045d602683bf, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
Block [id=734ab63a-cf31-4a33-8d68-3305f50128da, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
Block [id=dd7224c1-7455-41e9-97ee-dc9647b6d9d3, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
Block [id=f20fb444-1d29-48d9-8896-e669e350e2a6, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
Block [id=1cbf9f1d-4b5c-4348-a0f2-95dadef639cc, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
Block [id=bde61705-436b-47bf-b5bd-f7f44b3563cf, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
Block [id=06a6d867-6010-401f-94e9-acce3a08a8fb, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
Block [id=4e8e259d-d4a7-4228-8dbc-24969df08d97, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
Block [id=fa18be69-3b6e-4394-a2a1-d2c034bccb8e, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
Block [id=d36b7fe0-9721-4356-9a5a-037f5f5459a2, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
```

```
Block [id=1b4afb11-1f9a-432f-ba4c-23311fec9291, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz
Block [id=4ad5aab5-ddfd-4a97-8282-97d24ad71ea6, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz
Block [id=66a6fead-6d09-4e05-a785-17e0b486bdf9, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz
Block [id=5045642f-70dd-4390-add0-cc39a09ebcbf, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz
Block [id=8bd8aa16-0a2b-417b-a2a9-93af8d88a319, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz
Block [id=cedd5825-d74f-46bd-afa2-1c51578b19a5, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz
Block [id=9f2b77f3-71b4-4477-ac63-045d602683bf, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz
Block [id=734ab63a-cf31-4a33-8d68-3305f50128da, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz
Block [id=dd7224c1-7455-41e9-97ee-dc9647b6d9d3, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz
Block [id=f20fb444-1d29-48d9-8896-e669e350e2a6, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz
Block [id=1cbf9f1d-4b5c-4348-a0f2-95dadef639cc, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz
Block [id=bde61705-436b-47bf-b5bd-f7f44b3563cf, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz
Block [id=f827fd58-5f61-40de-bc76-b8ed6a898643, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz
```

5.3.3 40 percent of miners are cheaters

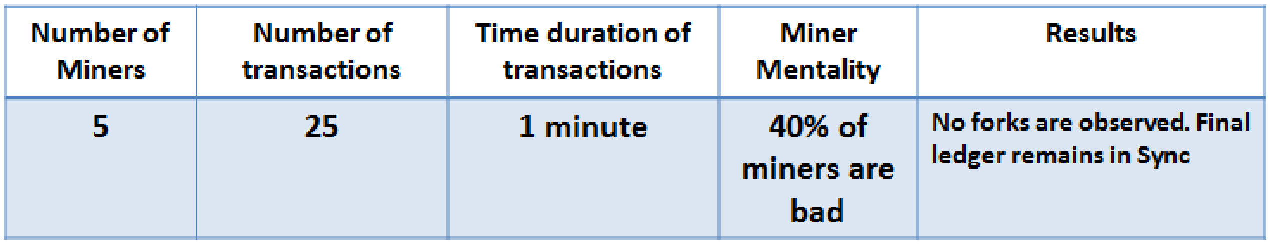
- This system test case is when 40 percent of the miners are good and rest are cheaters.
- BitFed sends the rate change message once every 30 seconds.
- The testing is done for 25 transactions which approximately take 60 seconds to execute.
- On test completion there is one fork observed in the transaction chain. The rest of the network builds on top of one of the block chains and thereby the

orphaned chain is rejected.

- We have shown below the comparison of two block chains of different miners and we observe that all the transaction ID's in the block chain are the same implying that the ledger is in sync.

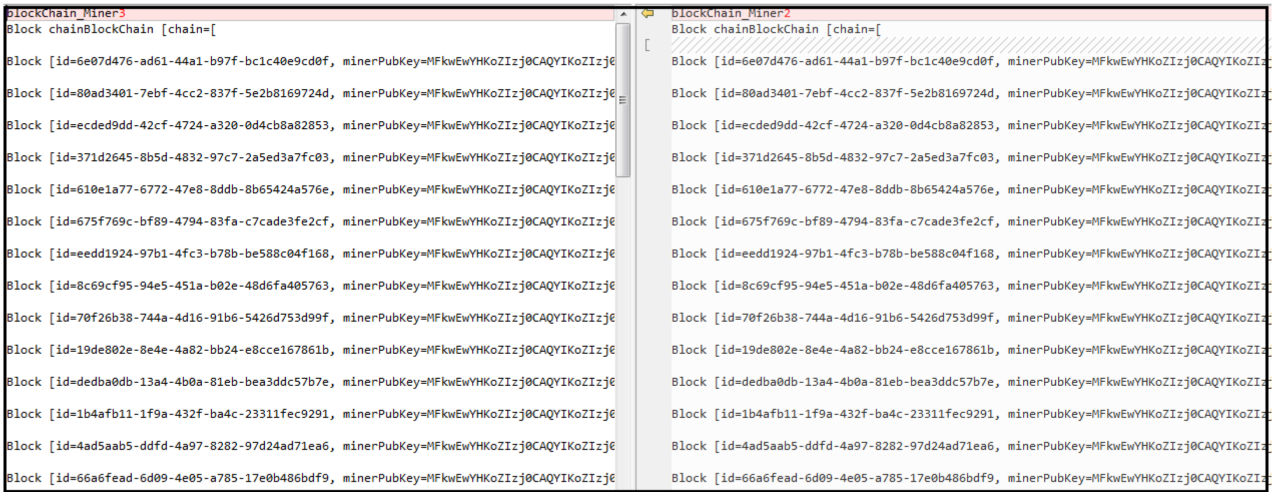
- From the above three tests we can conclude that in most of the cases the ledger remains in sync when the number of cheaters is below 40 percent. For odd cases when there is a fork, the rest of the network builds on top of the correct block chain, and the network rejects the orphaned block that is created.

Figure 34: Test Settings : 40 percent of miners are cheaters



Number of Miners	Number of transactions	Time duration of transactions	Miner Mentality	Results
5	25	1 minute	40% of miners are bad	No forks are observed. Final ledger remains in Sync

Figure 35: Test Results : 40 percent of miners are cheaters



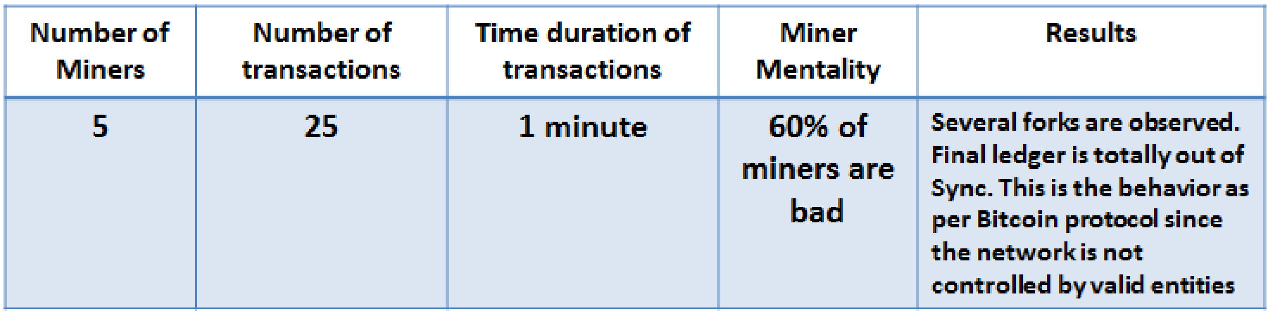
```
blockChain_Miner3
Block chainBlockChain [chain=[

Block [id=6e07d476-ad61-44a1-b97f-bc1c40e9cd0f, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=80ad3401-7ebf-4cc2-837f-5e2b8169724d, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=ecded9dd-42cf-4724-a320-0d4cb8a82853, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=371d2645-8b5d-4832-97c7-2a5ed3a7fc03, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=610e1a77-6772-47e8-8ddb-8b65424a576e, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=675f769c-bf89-4794-83fa-c7cade3fe2cf, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=eedd1924-97b1-4fc3-b78b-be588c04f168, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=8c69cf95-94e5-451a-b02e-48d6fa405763, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=70f26b38-744a-4d16-91b6-5426d753d99f, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=19de802e-8e4e-4a82-bb24-e8cce167861b, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=dedba0db-13a4-4b0a-81eb-bea3ddc57b7e, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=1b4afb11-1f9a-432f-ba4c-23311fec9291, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=4ad5aab5-ddfd-4a97-8282-97d24ad71ea6, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=66a6fead-6d09-4e05-a785-17e0b486bdf9, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
```

```
blockChain_Miner2
Block chainBlockChain [chain=[

Block [id=6e07d476-ad61-44a1-b97f-bc1c40e9cd0f, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=80ad3401-7ebf-4cc2-837f-5e2b8169724d, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=ecded9dd-42cf-4724-a320-0d4cb8a82853, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=371d2645-8b5d-4832-97c7-2a5ed3a7fc03, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=610e1a77-6772-47e8-8ddb-8b65424a576e, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=675f769c-bf89-4794-83fa-c7cade3fe2cf, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=eedd1924-97b1-4fc3-b78b-be588c04f168, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=8c69cf95-94e5-451a-b02e-48d6fa405763, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=70f26b38-744a-4d16-91b6-5426d753d99f, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=19de802e-8e4e-4a82-bb24-e8cce167861b, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=dedba0db-13a4-4b0a-81eb-bea3ddc57b7e, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=1b4afb11-1f9a-432f-ba4c-23311fec9291, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=4ad5aab5-ddfd-4a97-8282-97d24ad71ea6, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz

Block [id=66a6fead-6d09-4e05-a785-17e0b486bdf9, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIz
```

5.3.4 60 of miners are cheaters

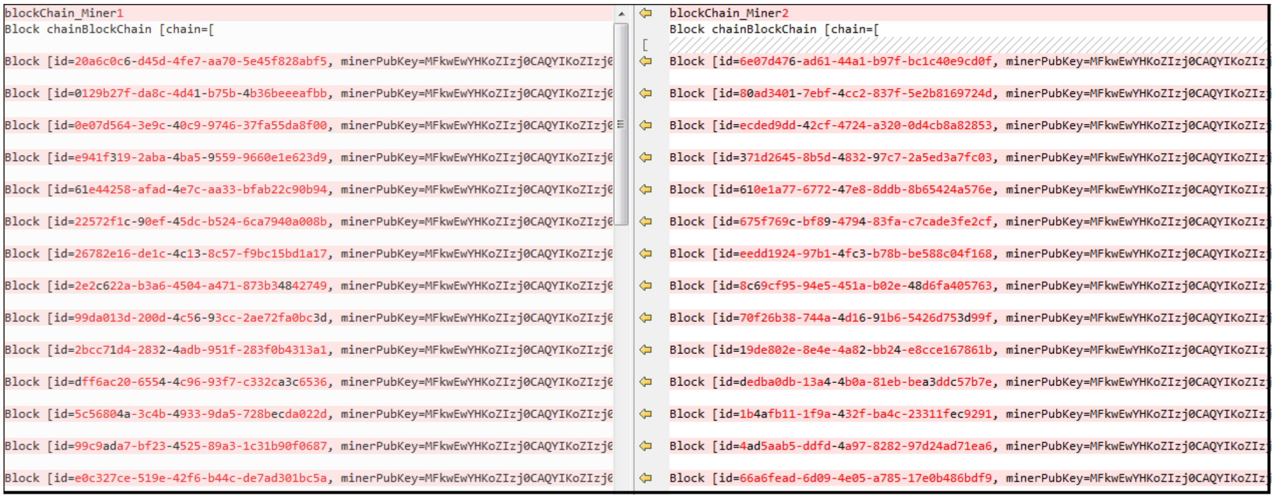
- This system test case is when 60 percent of the miners are good and rest are cheaters.
- BitFed sends the rate change message once every 30 seconds.
- The testing is done for 25 transactions which approximately take 60 seconds to execute
- On test completion there are several forks on the block chain. The block chain is totally out of sync.
- We have shown below the comparison of two block chains of different miners and many transactions in the chain do not match.
- As per the Bitcoin protocol as well, the network becomes extremely unstable when the number of cheaters in the network increase to a value of more than 51 percent. Until then, since the network is mostly controlled by sensible miners, the block chains remain in sync and achieve the pre-requisites of the protocol.

Figure 36: Test Settings : 60 percent of miners are cheaters



Number of Miners	Number of transactions	Time duration of transactions	Miner Mentality	Results
5	25	1 minute	60% of miners are bad	Several forks are observed. Final ledger is totally out of Sync. This is the behavior as per Bitcoin protocol since the network is not controlled by valid entities

Figure 37: Test Results : 60 percent of miners are cheaters



```
blockChain_Miner1
Block chainBlockChain [chain=[

Block [id=20a6c0c6-d45d-4fe7-aa70-5e45f828abf5, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=0129b27f-da8c-4d41-b75b-4b36beeeafbb, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=0e07d564-3e9c-40c9-9746-37fa55da8f00, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=e941f319-2aba-4ba5-9559-9660e1e623d9, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=61e44258-afad-4e7c-aa33-bfab22c90b94, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=22572f1c-90ef-45dc-b524-6ca7940a008b, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=26782e16-de1c-4c13-8c57-f9bc15bd1a17, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=2e2c622a-b3a6-4504-a471-873b34842749, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=99da013d-200d-4c56-93cc-2ae72fa0bc3d, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=2bcc71d4-2832-4adb-951f-283f0b4313a1, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=dff6ac20-6554-4c96-93f7-c332ca3c6536, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=5c56804a-3c4b-4933-9da5-728becda022d, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=99c9ada7-bf23-4525-89a3-1c31b90f0687, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0

Block [id=e0c327ce-519e-42f6-b44c-de7ad301bc5a, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0
```

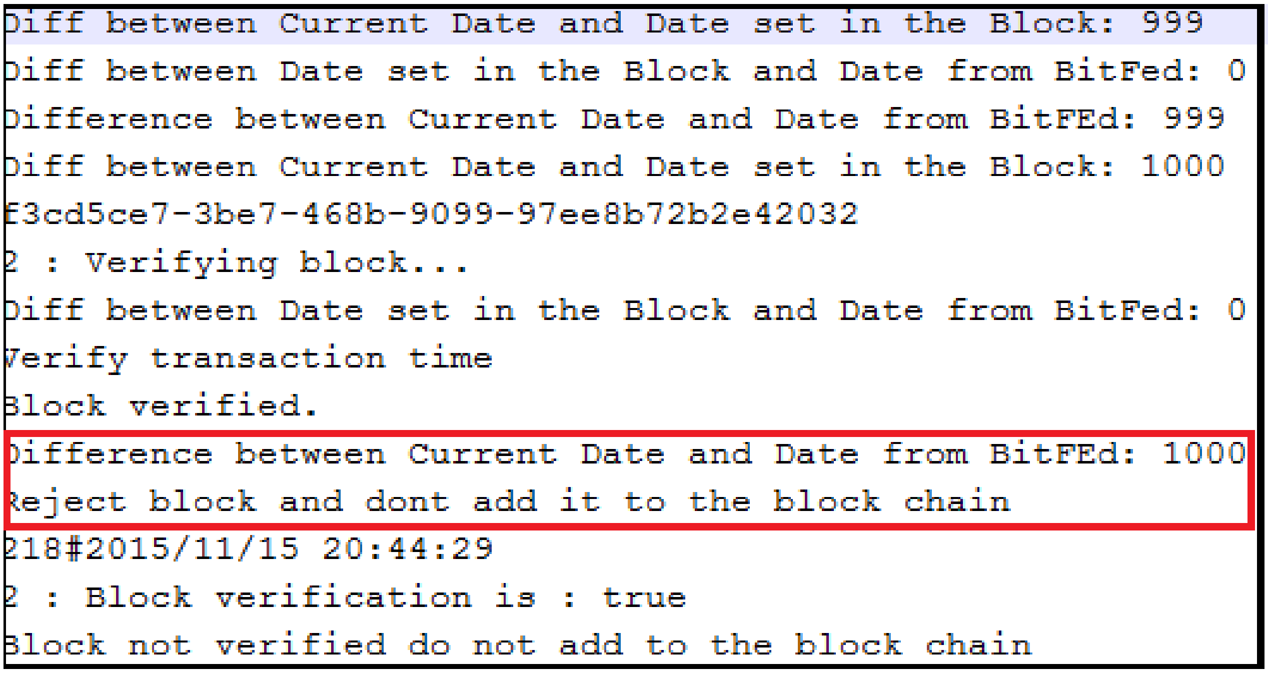
```
blockChain_Miner2
Block chainBlockChain [chain=[

Block [id=6e07d476-ad61-44a1-b97f-bc1c40e9cd0f, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj

Block [id=80ad3401-7ebf-4cc2-837f-5e2b8169724d, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj

Block [id=ecded9dd-42cf-4724-a320-0d4cb8a82853, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj

Block [id=371d2645-8b5d-4832-97c7-2a5ed3a7fc03, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj

Block [id=610e1a77-6772-47e8-8ddb-8b65424a576e, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj

Block [id=675f769c-bf89-4794-83fa-c7cade3fe2cf, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj

Block [id=eedd1924-97b1-4fc3-b78b-be588c04f168, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj

Block [id=8c69cf95-94e5-451a-b02e-48d6fa405763, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj

Block [id=70f26b38-744a-4d16-91b6-5426d753d99f, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj

Block [id=19de802e-8e4e-4a82-bb24-e8cce167861b, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj

Block [id=dedba0db-13a4-4b0a-81eb-bea3ddc57b7e, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj

Block [id=1b4afb11-1f9a-432f-ba4c-23311fec9291, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj

Block [id=4ad5aab5-ddfd-4a97-8282-97d24ad71ea6, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj

Block [id=66a6fead-6d09-4e05-a785-17e0b486bdf9, minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj
```

5.3.5 Detecting cheaters in the Network

- The cheaters in the network are detected if they try to add a package with old rate when the rate has expired and the time window for the rate has expired as well.

- The system checks if the rate used for computing the block is in the validity window. If the block is not in the validity range, the block is rejected otherwise miners add it to their block chain.

Figure 38: Test Results for detecting cheaters in the network

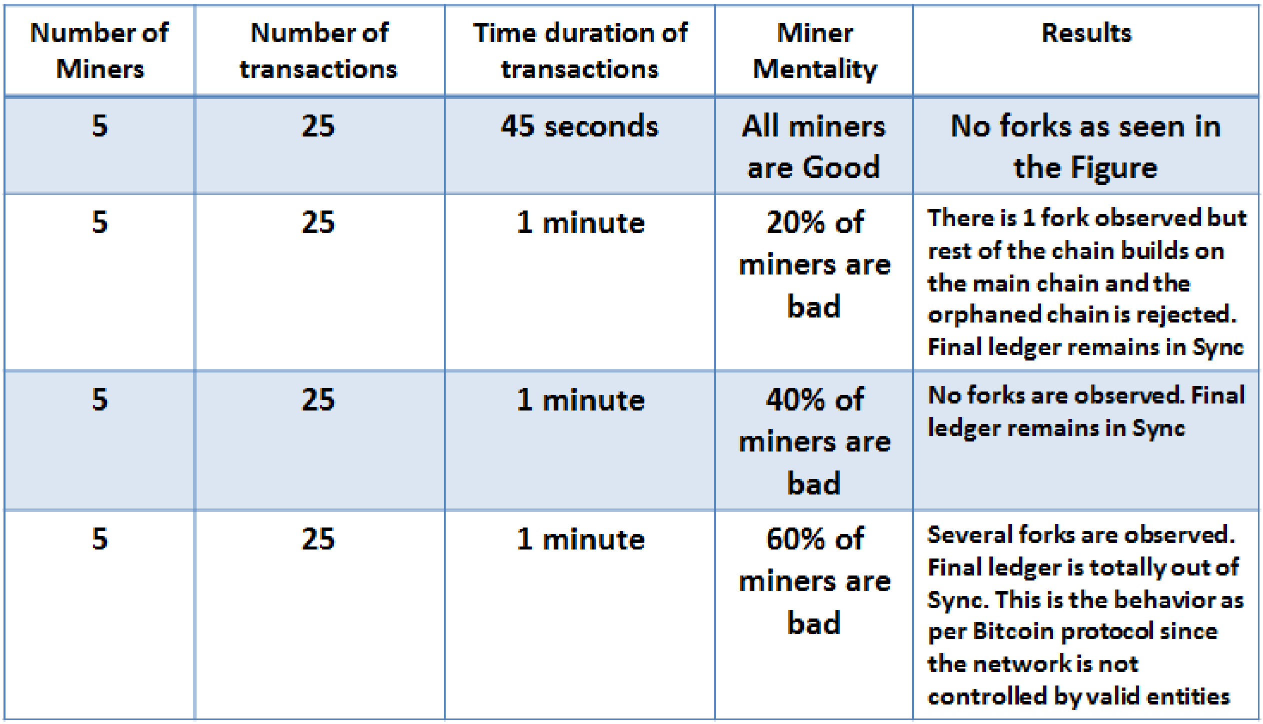


```
Diff between Current Date and Date set in the Block: 999
Diff between Date set in the Block and Date from BitFed: 0
Difference between Current Date and Date from BitFEd: 999
Diff between Current Date and Date set in the Block: 1000
f3cd5ce7-3be7-468b-9099-97ee8b72b2e42032
2 : Verifying block...
Diff between Date set in the Block and Date from BitFed: 0
Verify transaction time
Block verified.
Difference between Current Date and Date from BitFEd: 1000
Reject block and dont add it to the block chain
218#2015/11/15 20:44:29
2 : Block verification is : true
Block not verified do not add to the block chain
```

5.3.6 Result Summary

Figure 39: Test results

- The table summarizes the test results of all the testing that was done on the system.
- The results are as per the Bitcoin protocol where the block chains remain in sync when majority of miners in the network are good (more than 51 percent)
- When more than 51 percent of miners are bad, there are several forks observed and the block chains are no longer in sync.



Number of Miners	Number of transactions	Time duration of transactions	Miner Mentality	Results
5	25	45 seconds	All miners are Good	No forks as seen in the Figure
5	25	1 minute	20% of miners are bad	There is 1 fork observed but rest of the chain builds on the main chain and the orphaned chain is rejected. Final ledger remains in Sync
5	25	1 minute	40% of miners are bad	No forks are observed. Final ledger remains in Sync
5	25	1 minute	60% of miners are bad	Several forks are observed. Final ledger is totally out of Sync. This is the behavior as per Bitcoin protocol since the network is not controlled by valid entities

CHAPTER 6

Conclusion and Future works

We have successfully designed and developed a simulator to test our idea of creating a centralized authority BitFed to control inflation and deflation in a Bitcoin like cryptocurrency. As per the Bitcoin generation algorithm, the number of bitcoins in existence will never exceed 21 million, which will lead to deflation and encourage hoarding. Our approach mitigates this issue as shown by our experiments. The BitFed behaves as a federal reserve for the cryptocurrency economy and acts as a gatekeeper to promote stability in the rates of Bitcoin. It will thereby preserve the purchasing power of bitcoins. The decentralized nature of the Bitcoin protocol has been maintained and the BitFed only decides the rate at which new bitcoins are mined.

Our first set of experiments were conducted without cheaters in the network. We observed the consistency in block chains by testing the system without cheaters.

Our further tests were done by introducing cheaters into the network and checking how the system behaves. We found that the simulator was able to detect cheaters and not allow them to proceed with incorrect mining rates. We are thereby able to show that the protocol will help in overcoming the issues of deflation and thereby making Bitcoin a more stable currency.

Our future work involves:

- Avoid formation of mining pools : Miners may try to form colluding groups and the mining pool can take over the money supply. With the BitFed, the miners cannot form groups and publish block chains Reason : Timestamp and BitFed rates are validated as well!!

- Multiple forking issue : Would the BitFed create excessive forks? We need to analyze and regressively check if the approach of BitFed will create more forks.
- BitFed Offline: What would happen if BitFed went offline? We need to handle cases wherein the BitFed goes offline. Current handling suggests that the miners continue mining with the old BitFed rate. In our future work, if BitFed is offline the previous rate is used for 48 hours and after that the stand-by BitFed can be used.

LIST OF REFERENCES

[1] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008

[2] Ittay, Eyal and Emin Gun, Sirer, Majority is not Enough: Bitcoin Mining is Vulnerable, 2013

[3] Christian. Decker and Roger. Wattenhofer, Information propagation in the bit coin network, *IEEE International Conference on Peer-to-Peer Computing(P2P), Trento, Italy*, 2014

[4] Simon Barber, Xavier Boyen, Elaine Shi, and Ersin Uzun, Bitter to Better : How to Make Bitcoin a Better Currency, *Palo Alto Research Center, University of California, Berkeley*, 2012

[5] Joseph Bonneau, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll., Edward W. Felten. : SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies, *IEEE Symposium on Security and Privacy*, 2015

[6] Francois R. Velde : Bitcoin : A Primer, *Chicago Federal Letter*, 2013

[7] Deflationary Spiral : `https://en.bitcoin.it/wiki/Deflationary_spiral` Accessed 2013

[8] Amaanah Taqwaamani, The Moorish Diarium: A Diary of a Moor - The Great Maze of an International Monetary System, 2014

[9] Jerry Brito and Andrea Castillo, Bitcoin: A Primer for Policymakers, Mercatus Center. George Mason University. Retrieved 22, October 2013

[10] `http://data.bitcoinity.org/markets/price/5y/USD?c=e&r=week&t=l`

[11] `http://www.indianeconomy.net/splclassroom/69/why-deflation-is-dangerous-than-inflation/`

[12] Patricia Buckley Ebrey, Anne Walthall, James B. Palais, East Asia: A Cultural, Social, and Political History *Houghton Mifflin, 2006*

[13] Chaum, David, "Blind signatures for untraceable payments" (PDF). *Advances in Cryptology Proceedings of Crypto*, 1983

[14] Tatsuaki Okamoto and Kazuo Ohta, Universal Electronic Cash *Advances in Cryptology*, 1992

[15] https://bitcointalk.org/index.php?topic=3366.msg47522#msg47522, Accessed 11/10/2015

[16] https://news.google.com/newspapers?nid=1499&dat=19630103&id=lxQoAAAAIBAJ&sjid=ziYEAAAAIBAJ&pg=1429,595644&hl=en, Accessed 11/10/2015

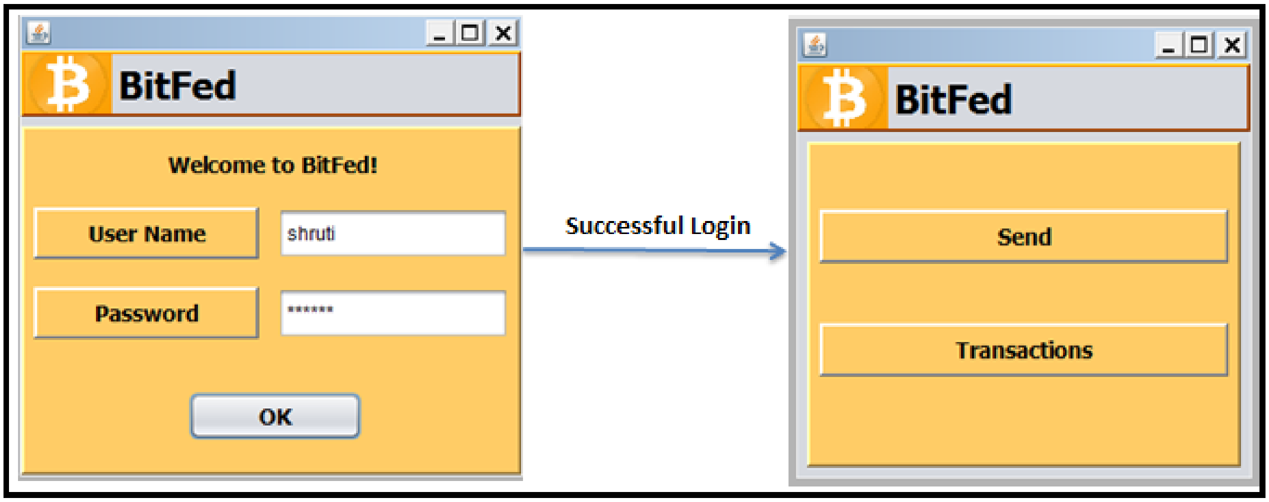
APPENDIX

Unit Testing

A.0.7 Wallet Login : Testing the execution of Wallet

- Every user has a wallet that he has to login to in order to execute a transaction.
- Login page has the user name and a password.
- When the user enters the information and presses OK, his information is verified and if he is the authorized user, he is allowed to login.
- The first part is successful login : Enter username as shruti and Password as sharma; the system allows user to login successfully

Figure A.40: Successful Login



BitFed

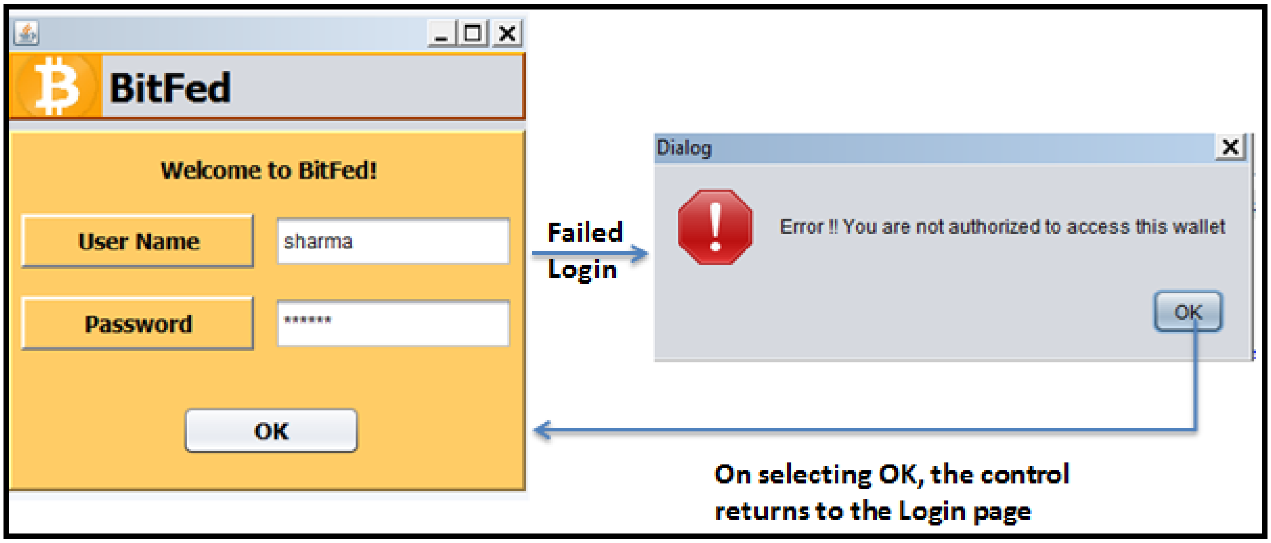
Welcome to BitFed!

User Name: shruti

Password: ******

- The second part is unsuccessful login : Enter username and sharma and any other password; the system displays an error message

Figure A.41: Unsuccessful login



Welcome to BitFed!

User Name: sharma

Password: ******

Failed Login

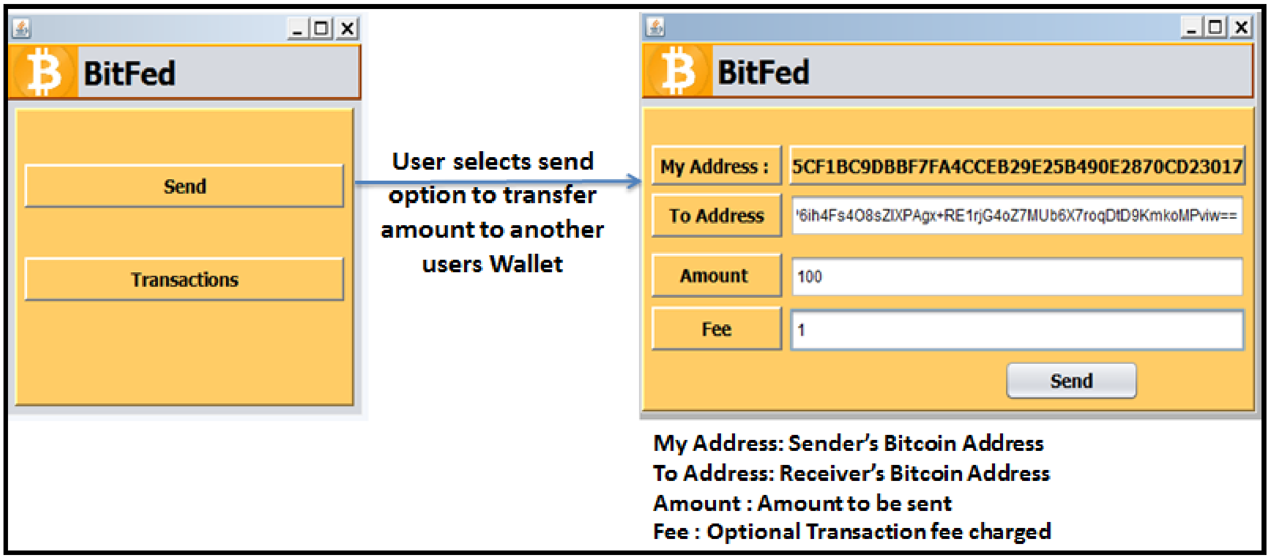
Error !! You are not authorized to access this wallet

On selecting OK, the control returns to the Login page

A.0.8 Wallet Send : Testing sending of transaction to another Wallet

- Sending of amount takes place using the Bitcoin addresses of the Wallet owners.
- The user selects the Bitcoin address to send the amount, Amount to be sent, Fee.
- When the user selects send, the transaction is broadcast to the mining pool for mining.

Figure A.42: Send amount to another user's Wallet



User selects send option to transfer amount to another users Wallet

My Address : 5CF1BC9DBBF7FA4CCEB29E25B490E2870CD23017

To Address '6ih4Fs4O8sZIXPAgx+RE1rjG4oZ7MUb6X7roqDtD9KmkoMPviw==

Amount 100

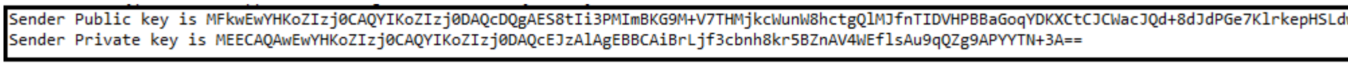
Fee 1

My Address: Sender's Bitcoin Address
To Address: Receiver's Bitcoin Address
Amount : Amount to be sent
Fee : Optional Transaction fee charged

A.0.9 Testing Keypair generation and Bitcoin address generation

- Generate key pair and check if it generates private and public key.
- These keys are what will be used for signing and bitcoin address generation.

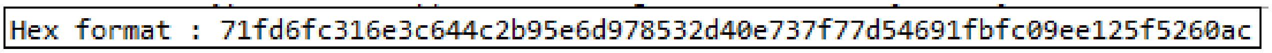
Figure A.43: Key Pair generation : Private and Public Key



```
Sender Public key is MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAES8tIi3PMImBKG9M+V7THMjkcWunW8hctgQlMJfnTIDVHPBBaGoqYDKXCtCJCWacJQd+8dJdPGe7KlrkepHSLd
Sender Private key is MEECAQAwEwYHKoZIzj0CAQYIKoZIzj0DAQcEJzAlAgEBBCAiBrLjf3cbnh8kr5BZnAV4WEflsAu9qQZg9APYYTN+3A==
```

- Generate Bitcoin address using public key.

Figure A.44: Bitcoin Address generation



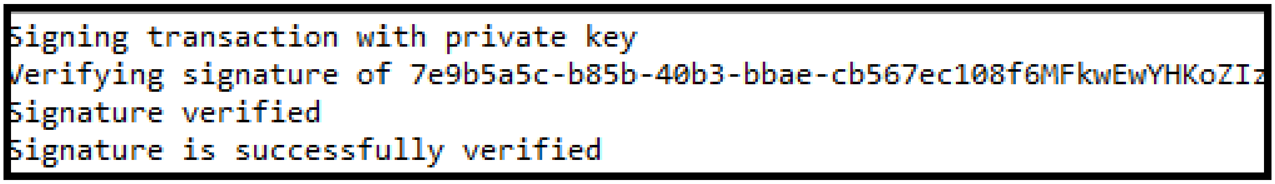
```
Hex format : 71fd6fc316e3c644c2b95e6d978532d40e737f77d54691fbfc09ee125f5260ac
```

A.0.10 Testing Signature verification

- This test case is to sign a transaction message and verify the signature.

- The message signed includes ID, public key of sender and the amount.

Figure A.45: Signature Verification



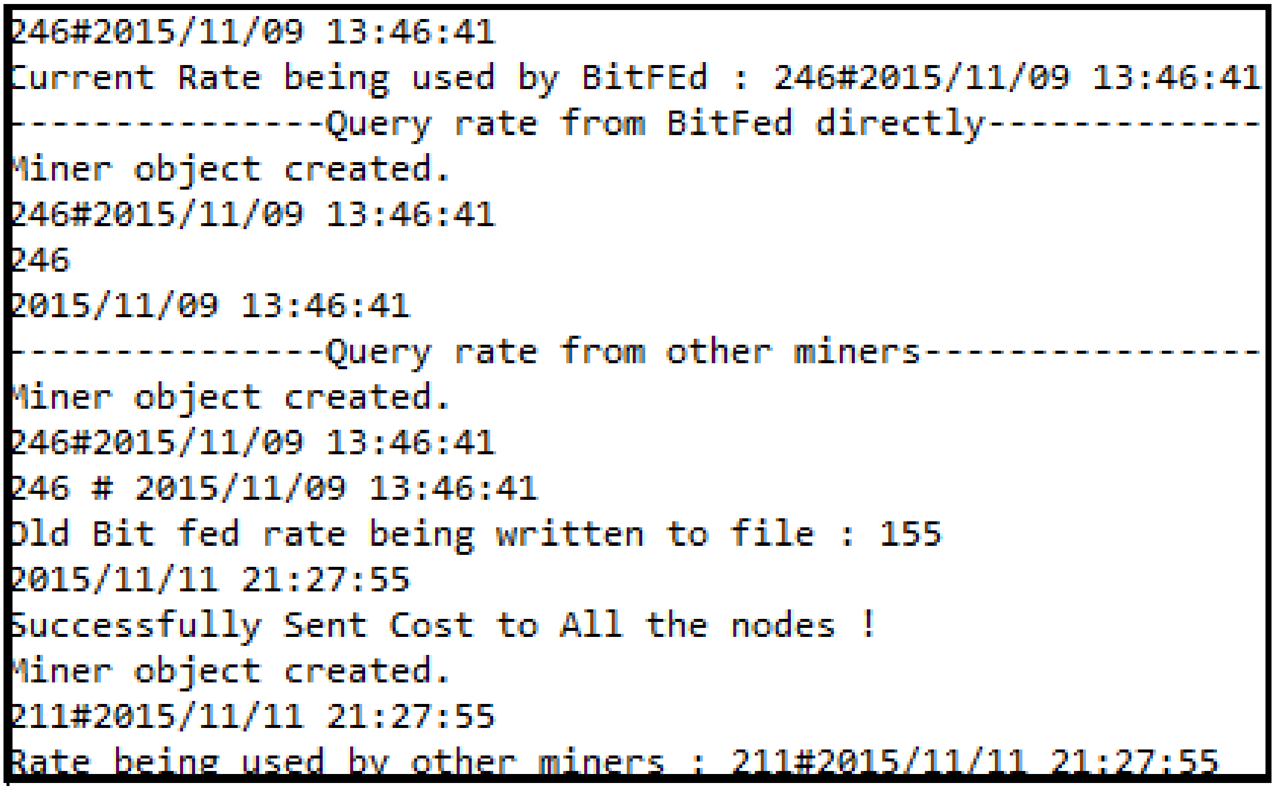
```
Signing transaction with private key
Verifying signature of 7e9b5a5c-b85b-40b3-bbae-cb567ec108f6MFkwEwYHKoZIz
Signature verified
Signature is successfully verified
```

A.0.11 Query BitFed rate

- Miners who were offline can query the current BitFed rate from other miners on the network.

- This testcase helps to verify if the functionality of rate query works fine or not.

Figure A.46: Offline miner querying for BitFed rate



```
246#2015/11/09 13:46:41
Current Rate being used by BitFEd : 246#2015/11/09 13:46:41
---------------Query rate from BitFed directly-------------
Miner object created.
246#2015/11/09 13:46:41
246
2015/11/09 13:46:41
---------------Query rate from other miners----------------
Miner object created.
246#2015/11/09 13:46:41
246 # 2015/11/09 13:46:41
Old Bit fed rate being written to file : 155
2015/11/11 21:27:55
Successfully Sent Cost to All the nodes !
Miner object created.
211#2015/11/11 21:27:55
Rate being used by other miners : 211#2015/11/11 21:27:55
```

A.0.12 Testing Transaction and Block chain creation

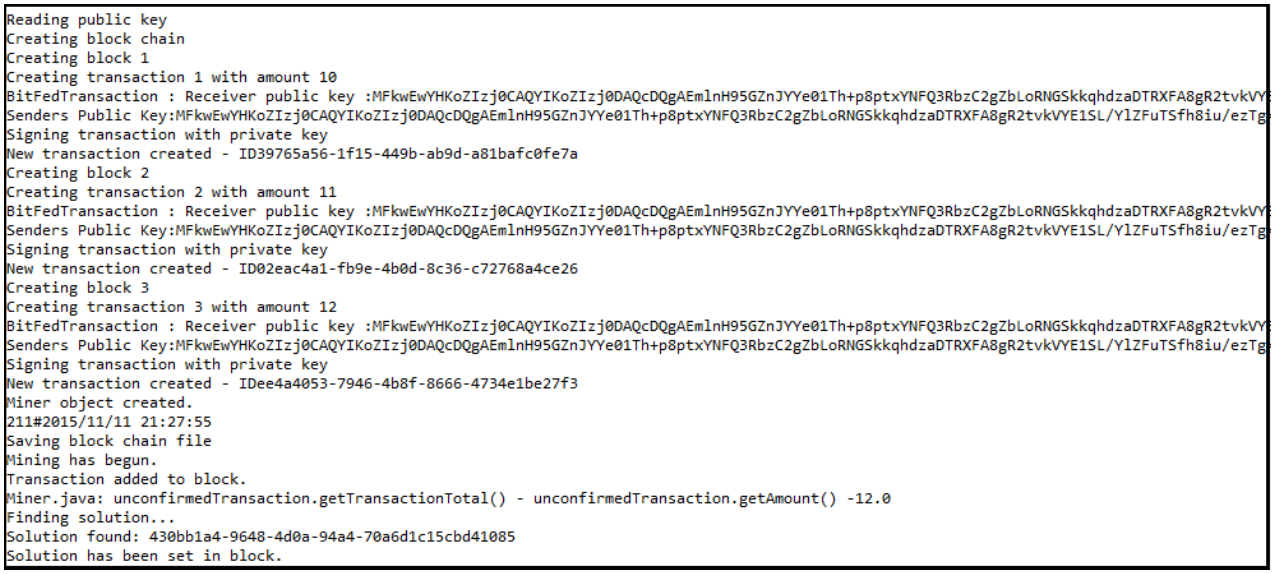
- This testcase is to verify if a block chain is getting created successfully or not.

- Block chain is a very important feature of the protocol.

- The block chain will be shared amongst other miners once computed.

- This test case involves creation of unconfirmed transaction, adding it to block, and verifying the proof of work algorithm.

Figure A.47: Transaction and Block creation

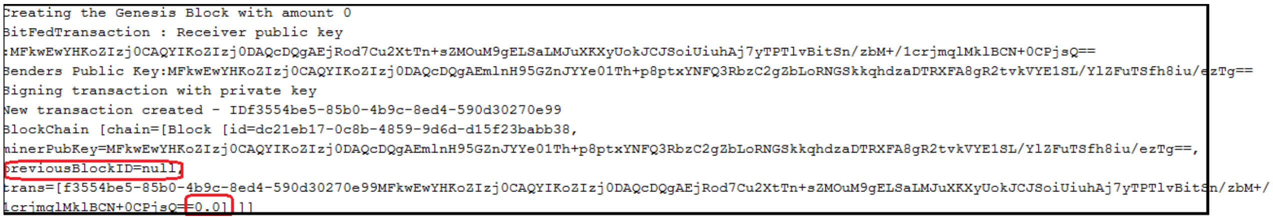


```
Reading public key
Creating block chain
Creating block 1
Creating transaction 1 with amount 10
BitFedTransaction : Receiver public key :MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEmlnH95GZnJYYe01Th+p8ptxYNFQ3RbzC2gZbLoRNGSkkqhdzaDTRXFA8gR2tvkVY
Senders Public Key:MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEmlnH95GZnJYYe01Th+p8ptxYNFQ3RbzC2gZbLoRNGSkkqhdzaDTRXFA8gR2tvkVYE1SL/YlZFuTSfh8iu/ezTg
Signing transaction with private key
New transaction created - ID39765a56-1f15-449b-ab9d-a81bafc0fe7a
Creating block 2
Creating transaction 2 with amount 11
BitFedTransaction : Receiver public key :MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEmlnH95GZnJYYe01Th+p8ptxYNFQ3RbzC2gZbLoRNGSkkqhdzaDTRXFA8gR2tvkVY
Senders Public Key:MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEmlnH95GZnJYYe01Th+p8ptxYNFQ3RbzC2gZbLoRNGSkkqhdzaDTRXFA8gR2tvkVYE1SL/YlZFuTSfh8iu/ezTg
Signing transaction with private key
New transaction created - ID02eac4a1-fb9e-4b0d-8c36-c72768a4ce26
Creating block 3
Creating transaction 3 with amount 12
BitFedTransaction : Receiver public key :MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEmlnH95GZnJYYe01Th+p8ptxYNFQ3RbzC2gZbLoRNGSkkqhdzaDTRXFA8gR2tvkVY
Senders Public Key:MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEmlnH95GZnJYYe01Th+p8ptxYNFQ3RbzC2gZbLoRNGSkkqhdzaDTRXFA8gR2tvkVYE1SL/YlZFuTSfh8iu/ezTg
Signing transaction with private key
New transaction created - IDee4a4053-7946-4b8f-8666-4734e1be27f3
Miner object created.
211#2015/11/11 21:27:55
Saving block chain file
Mining has begun.
Transaction added to block.
Miner.java: unconfirmedTransaction.getTransactionTotal() - unconfirmedTransaction.getAmount() -12.0
Finding solution...
Solution found: 430bb1a4-9648-4d0a-94a4-70a6d1c15cbd41085
Solution has been set in block.
```

A.0.13 Testing Genesis Block creation

- This testcase is to verify if the Genesis block is getting created successfully or not.
- Genesis Block is the first block of the block chain and marks the beginning of the block chain.
- This block has the previous block ID set to null since there are no other blocks before this block.
- This test case involves creation of transaction with amount set to 0.

Figure A.48: Genesis block creation



```
Creating the Genesis Block with amount 0
BitFedTransaction : Receiver public key
:MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEjRod7Cu2XtTn+sZMOuM9gELSaLMJuXKXyUokJCJSoiUiuhAj7yTPTlvBitSn/zbM+/1crjmqlMklBCN+0CPjsQ==
Senders Public Key:MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEmlnH95GZnJYYe01Th+p8ptxYNFQ3RbzC2gZbLoRNGSkkqhdzaDTRXFA8gR2tvkVYE1SL/YlZFuTSfh8iu/ezTg==
Signing transaction with private key
New transaction created - IDf3554be5-85b0-4b9c-8ed4-590d30270e99
BlockChain [chain=[Block [id=dc21eb17-0c8b-4859-9d6d-d15f23babb38,
minerPubKey=MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEmlnH95GZnJYYe01Th+p8ptxYNFQ3RbzC2gZbLoRNGSkkqhdzaDTRXFA8gR2tvkVYE1SL/YlZFuTSfh8iu/ezTg==,
previousBlockID=null,
trans=[f3554be5-85b0-4b9c-8ed4-590d30270e99MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEjRod7Cu2XtTn+sZMOuM9gELSaLMJuXKXyUokJCJSoiUiuhAj7yTPTlvBitSn/zbM+/
1crjmqlMklBCN+0CPjsQ==0.0]]]
```

A.0.14 Testing Proof-of-work algorithm

- This is the main portion of the BitFed protocol and every miner has to solve the proof-of-work problem in order to get their transactions added to the global ledger.

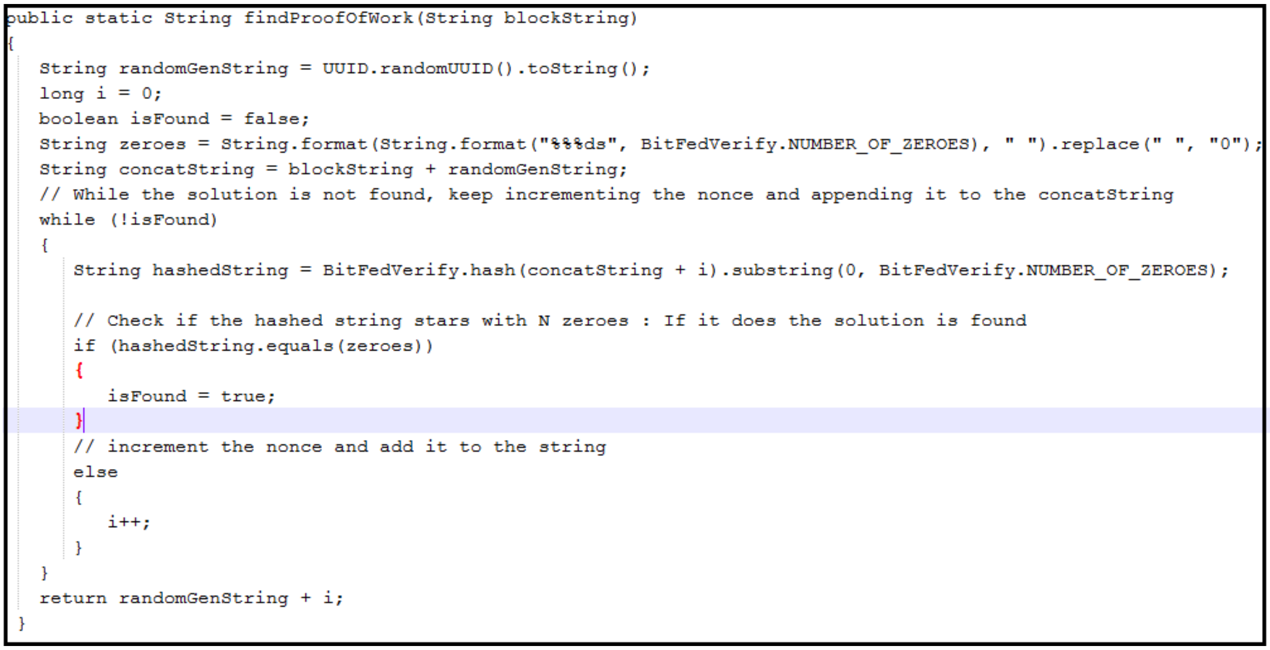
- The algorithm involves solving cryptographic puzzles to find a solution which starts with N zeroes.

- Value of N is preset and changed when the toughness of the problem requires a change.

- This test case involves creation of transaction with amount set to 0.

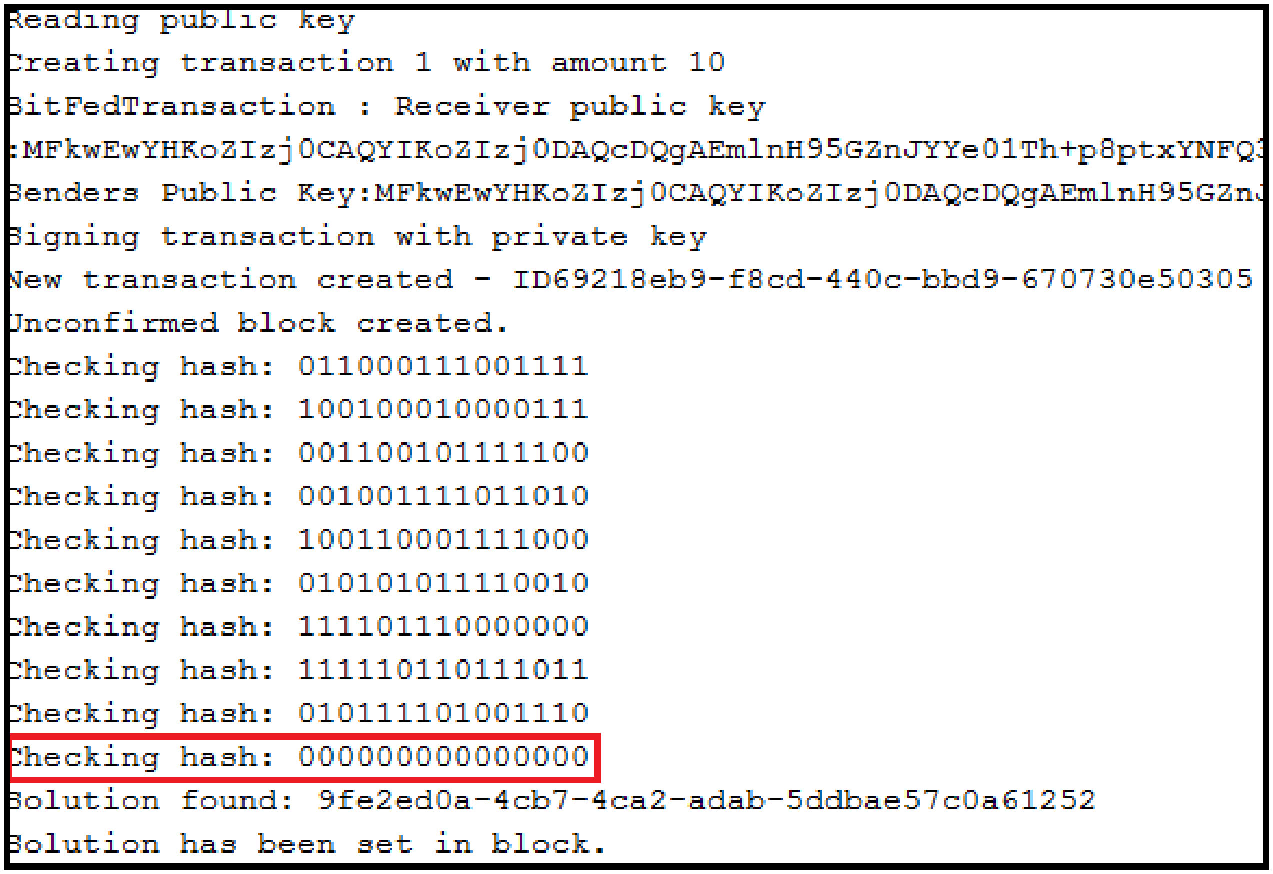
- Code snippet of the proof-of-work algorithm is shown in figure below.

Figure A.49: Code snippet of the proof-of-work algorithm



```
public static String findProofOfWork(String blockString)
{
   String randomGenString = UUID.randomUUID().toString();
   long i = 0;
   boolean isFound = false;
   String zeroes = String.format(String.format("%%%ds", BitFedVerify.NUMBER_OF_ZEROES), " ").replace(" ", "0");
   String concatString = blockString + randomGenString;
   // While the solution is not found, keep incrementing the nonce and appending it to the concatString
   while (!isFound)
   {
      String hashedString = BitFedVerify.hash(concatString + i).substring(0, BitFedVerify.NUMBER_OF_ZEROES);

      // Check if the hashed string stars with N zeroes : If it does the solution is found
      if (hashedString.equals(zeroes))
      {
         isFound = true;
      }
      // increment the nonce and add it to the string
      else
      {
         i++;
      }
   }
   return randomGenString + i;
 }
```

Figure A.50: Test case results

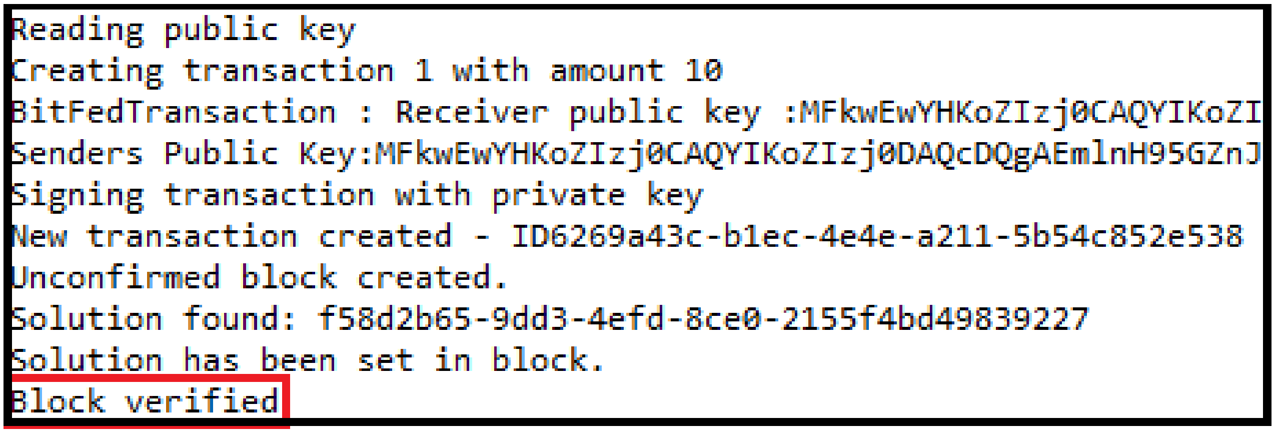


```
Reading public key
Creating transaction 1 with amount 10
BitFedTransaction : Receiver public key
:MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEmlnH95GZnJYYe01Th+p8ptxYNFQ
Senders Public Key:MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEmlnH95GZn
Signing transaction with private key
New transaction created - ID69218eb9-f8cd-440c-bbd9-670730e50305
Unconfirmed block created.
Checking hash: 011000111001111
Checking hash: 100100010000111
Checking hash: 001100101111100
Checking hash: 001001111011010
Checking hash: 100110001111000
Checking hash: 010101011110010
Checking hash: 111101110000000
Checking hash: 111110110111011
Checking hash: 010111101001110
Checking hash: 000000000000000
Solution found: 9fe2ed0a-4cb7-4ca2-adab-5ddbae57c0a61252
Solution has been set in block.
```

A.0.15 Verifying results of proof-of-work

- This testcase is to verify if the proof-of-work shared by the miner is accurate or not.
- Based on this result, the user decides whether to include the block chain in the global ledger or not.

Figure A.51: Verifying the results of proof-of-work



```
Reading public key
Creating transaction 1 with amount 10
BitFedTransaction : Receiver public key :MFkwEwYHKoZIzj0CAQYIKoZI
Senders Public Key:MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEmlnH95GZnJ
Signing transaction with private key
New transaction created - ID6269a43c-b1ec-4e4e-a211-5b54c852e538
Unconfirmed block created.
Solution found: f58d2b65-9dd3-4efd-8ce0-2155f4bd49839227
Solution has been set in block.
Block verified
```