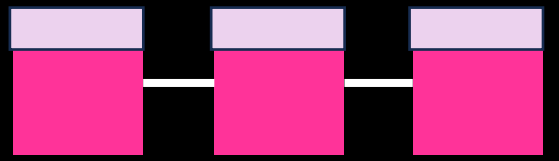
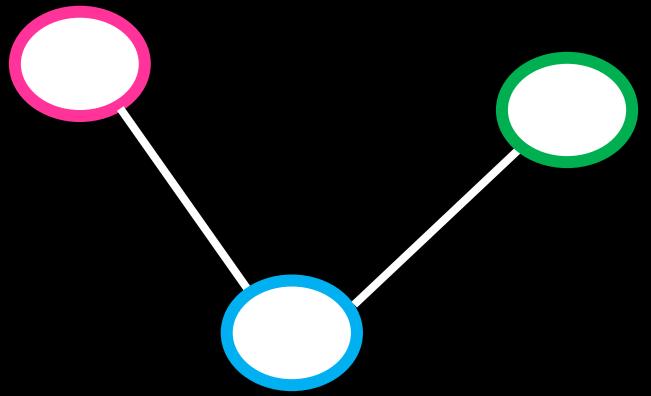
Graph Modelling In Block-Chain P2P (Client Complete Graph) Network Broadcasting via WebSocket





Cookers :

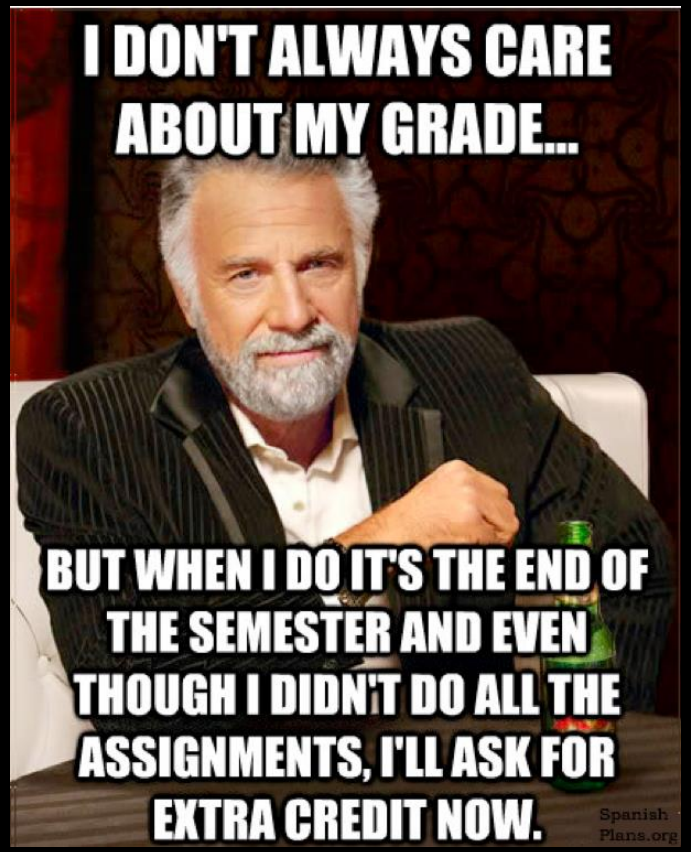
- Abdan Hafidz (5054231021)
- Jeremiah Kevin Alexander J. Malau (5054231)
- M Farhan Arya Wicaksono (5054231)

Idea / Background

As students in a rapidly advancing technological era, it is crucial for us to stay abreast of developments in Smart Contracts, Web 3.0, and Cryptocurrencies, and to actively participate as creators, not just users. Our project, focusing on the implementation of Breadth-First Search (BFS) in a Blockchain Peer-to-Peer (P2P) network using a complete graph model for client connections and WebSocket for broadcasting, aims to enhance our technical skills and contribute to innovative communication methods within blockchain networks. By doing so, we strive to bridge the knowledge gap and correct public misconceptions about blockchain, cryptocurrencies, and smart contracts. Utilizing graph theory, particularly complete graphs where every pair of distinct vertices is connected, our project not only deepens our understanding of blockchain principles but also serves as an educational tool to empower the community with accurate information and foster a more informed society.

Project Aims

- **Technical Proficiency**: Developing and implementing BFS in a blockchain P2P network hones our technical skills and deepens our understanding of the underlying principles of blockchain technology.
- **Innovation:** By exploring the integration of WebSocket for network broadcasting, we investigate efficient and real-time communication methods within blockchain networks, pushing the boundaries of current technological capabilities.
- **Educational Outreach:** Through this project, we aim to create educational content and presentations that elucidate complex concepts in blockchain and related technologies. This will help demystify these technologies for the general public, fostering a more informed and literate society.Real-World
- **Applications:** Understanding and implementing these technologies equips us to develop practical applications, such as decentralized applications (DApps) and smart contracts, contributing to the broader ecosystem of Web 3.0.
- ***Dapat Nilai A, Bismillah hehehe***



What Is Blockchain?

Let that you have a data instanced “Transaction Information”

“Naufal membayar sebesar 20 BTC kepada Restoran Warung Pohon”

We can store the information above to a record-block

Naufal didn't want there to be any doubt so he made his signature

We can add a timestamp as additional information

NOPALJMK68

```
Message : “Naufal membayar
sebesar 20 BTC kepada Restoran
Warung Pohon”

TimeStamp : 2024-06-01 03:00:00
```

What Is Blockchain?

What if we add another transaction to the with new generation block

Message : "Naufal membayar sebesar 20 BTC kepada Restoran Warung Pohon"

TimeStamp : 2024-06-01 03:00:00

NOPALJMK68

Message : "Naufal menerima 100 BTC dari Abdan"

TimeStamp : 2024-06-01 04:00:00

NOPALJMK68

Message : "Naufal ngutang 120 BTC ke Kepin"

TimeStamp : 2024-06-01 05:00:00

NOPALJMK68

Now, if you are aware, someone else could fake the information data block as long as they know that the signature is *NOPALJMK68*.

What Is Blockchain?

We don't want that to happen, so the only solution is to secure your signature. Here we will work with an encryption technique based on Cryptography called *"hash"*

In computing, a hash is a function that converts an input (or 'message') into a fixed-size string of bytes. The output is typically a 'digest' that uniquely represents the input data. Hash functions are commonly used in data integrity verification, password storage, and quick data retrieval.

hash("NOPAL SAYANG KEPIN") = 387491986286848829899

In the project we are developing, we use a fairly well-known hash cryptographic algorithm, namely sha256

SHA256("NOPAL SAYANG KEPIN") = 248ec6fffdc81ea41fab761f1ae87b83598ba46bfd712785b9cb8e13399a6afb

FYI : Sha256 is used in bitcoin technology

What Is Blockchain?

We don't want that to happen, so the only solution is to secure your signature. Here we will work with an encryption technique based on Cryptography called *"hash"*

Hash :
248ec6fffdc81ea41fab761f1ae87b83598ba46bfd712785b9cb8e13399a6afb
Message : "Naufal membayar sebesar 20 BTC kepada Restoran Warung Pohon"
TimeStamp : 2024-06-01 03:00:00

Hash :
248ec6fffdc81ea41fab761f1ae87b83598ba46bfd712785b9cb8e13399a6afb

Message : "Naufal menerima 100 BTC dari Abdan"

TimeStamp : 2024-06-01 04:00:00

Hash :
248ec6fffdc81ea41fab761f1ae87b83598ba46bfd712785b9cb8e13399a6afb

Message : "Naufal ngutang 120 BTC ke Kepin"

TimeStamp : 2024-06-01 05:00:00

What Is Blockchain?

But it doesn't stop there, so that the hash of each block has a unique/different value and this is very effective, we follow the principles of blockchain cryptography in general as applied to bitcoin technology

hash(currentBlock) = `hash_currentBlock_information` + `hash_previous_block` + `privateKey` + `nonce`

The nonce "number used only once" is a random number that is generated where this number meets the consensus on the specified hash.

For example, everyone agrees to determine that the nonce is a random number which, if entered into the hash, means that the bits in front of the hash have 6 "0" characters.

sha256("Message : I <3 U2" + "Message :I <3 U" + NOPALJMK68 + __) = 000000___

sha256("Message : I <3 U2" + "Message :I <3 U" + NOPALJMK68 + 14333) =
000000101000100010100101110001110001001010001000100110011011101011000011001010001011001011010110010001100010101110101001111111001000101000100010001100011001010110111101100110100011111100110001010001111101110100110000101101001000

Nonce : 1433

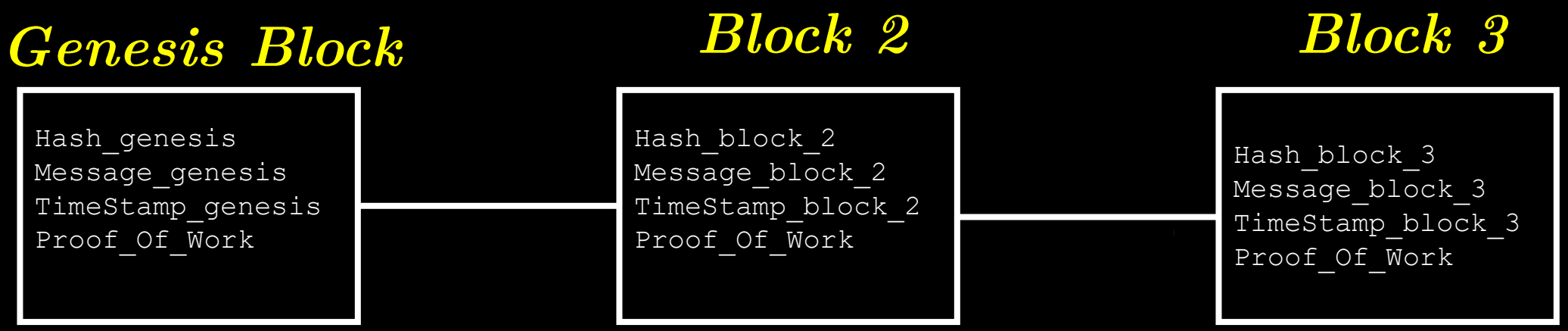
sha256("Message : I <3 U2" + "Message :I <3 U" + NOPALJMK68 + 12) = 100000001 …

What Is Blockchain?

Nonce also acts as a verification element which can prevent someone from creating lots of blockchain spam, DDos

So in the verification we can go through "proof of work", namely to generate a nonce - block that follows the consensus, for example the hash must contain 6 bits "0". How many steps does it take?

For example, to find the number 1433, I needed 5 computational steps using a randomize algorithm



A block chain can be represented as a graph with *N* vertices and *N - 1* edges where each vertex is connected to a maximum of one other vertex, and is connected by one edge *(vertex = block)*.

P2P-Network

The initial idea of a block chain like Bitcoin, for example, was to disconnect a centralized party from being the bridge in a data transaction, we call this decentralization, where clients can directly interact with other clients without going through a particular party.

So the method we chose in this block-chain network simulation is to use P2P Network Broadcasting

A Peer-to-Peer (P2P) network in blockchain is a decentralized system where each participant (or peer) has equal capabilities and responsibilities, eliminating the need for a central authority. This network architecture ensures data and tasks are distributed across multiple nodes, enhancing security, robustness, and resistance to censorship. In this setup, nodes interact directly with each other, initiating, validating, and propagating transactions throughout the network. This client-to-client interaction is fundamental to blockchain's operation, as it ensures that transactions are processed efficiently and transparently without relying on a centralized server.

With P2P Network we can broadcast every recorded transaction and share the block-chain with others

P2P-Network

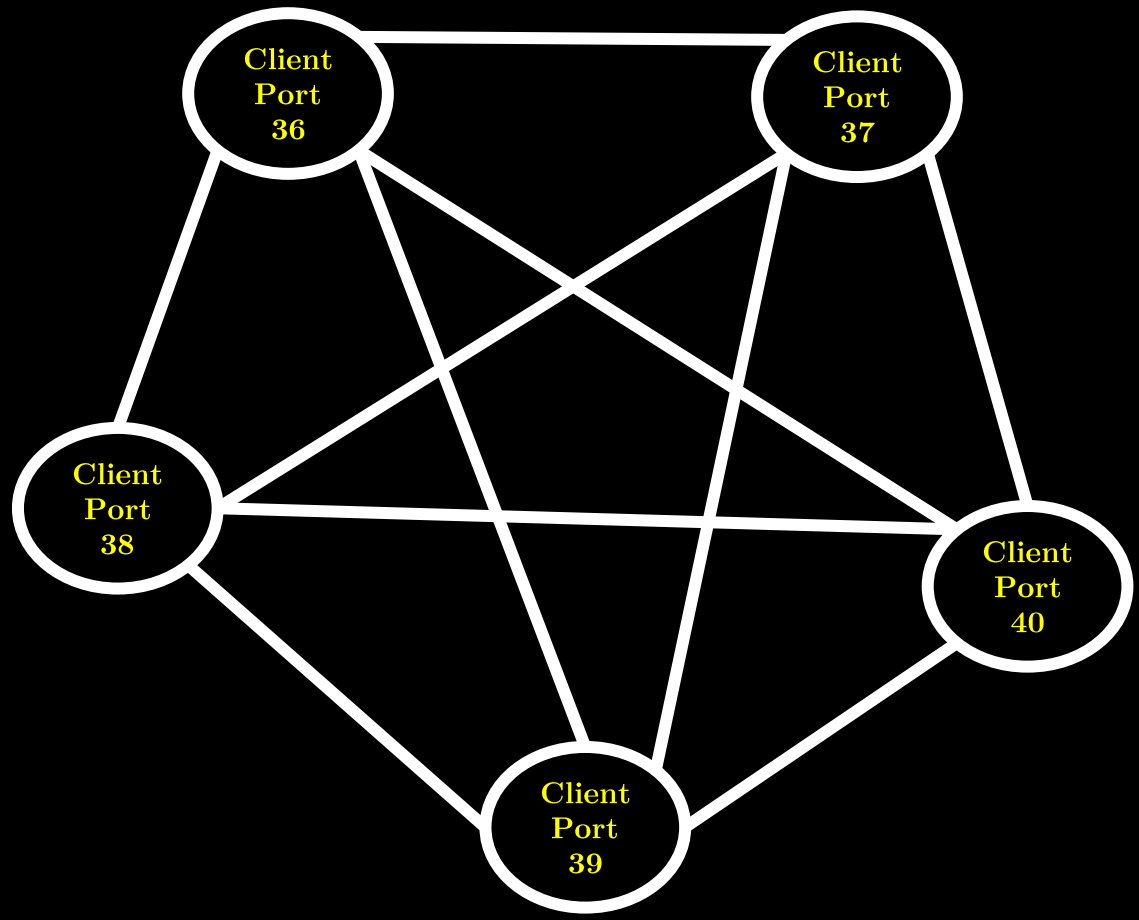
P2P Network is a network that can be simulated as a Complete Graph.

There are N clients as vertices, because each client can broadcast / mass transfer to all clients, each client will be connected to N - 1 other clients.

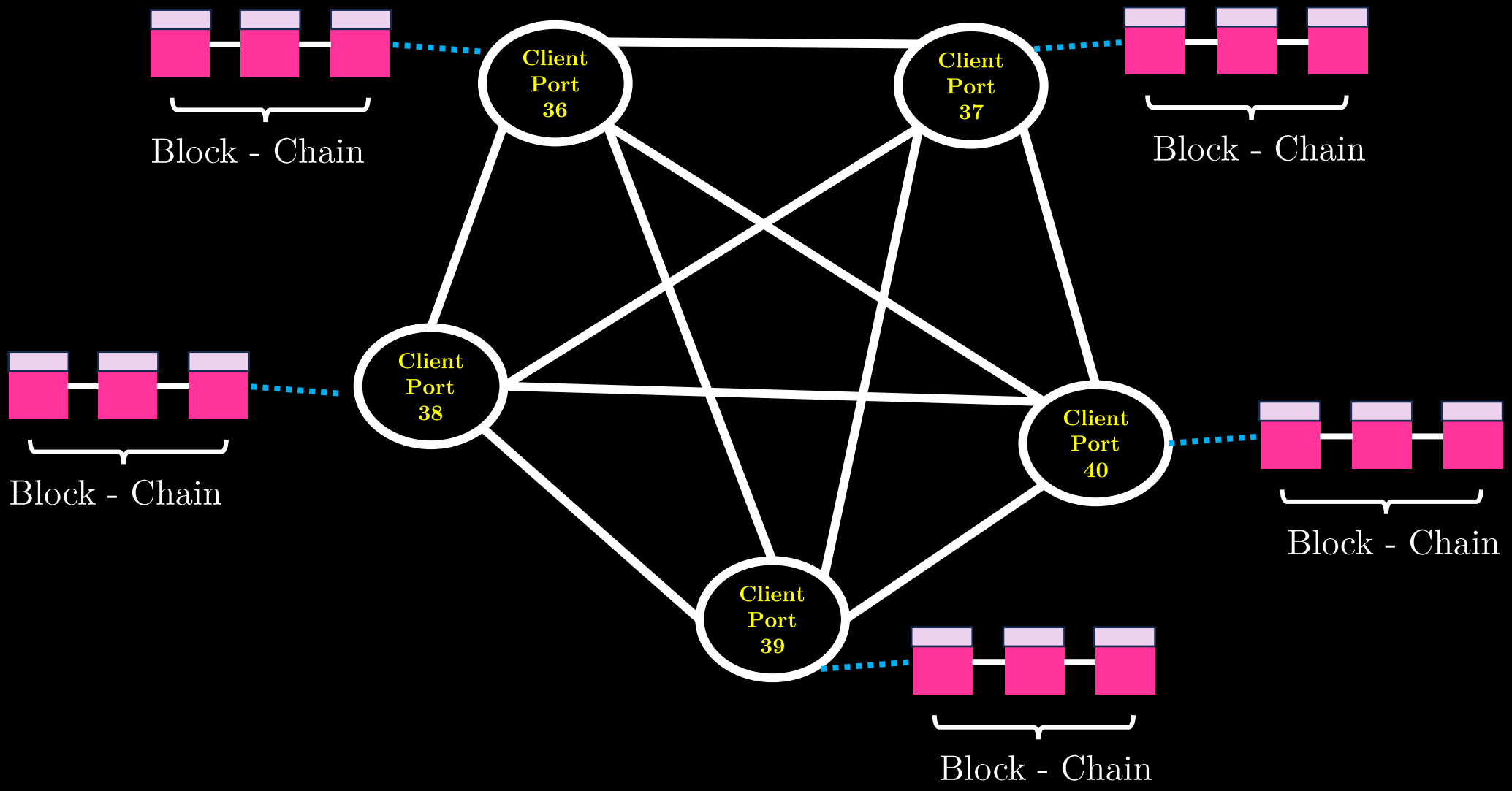
In accordance with Graph Theory,

a Complete Graph Kn has :

$$\frac{n(n-1)}{2} \; edges$$

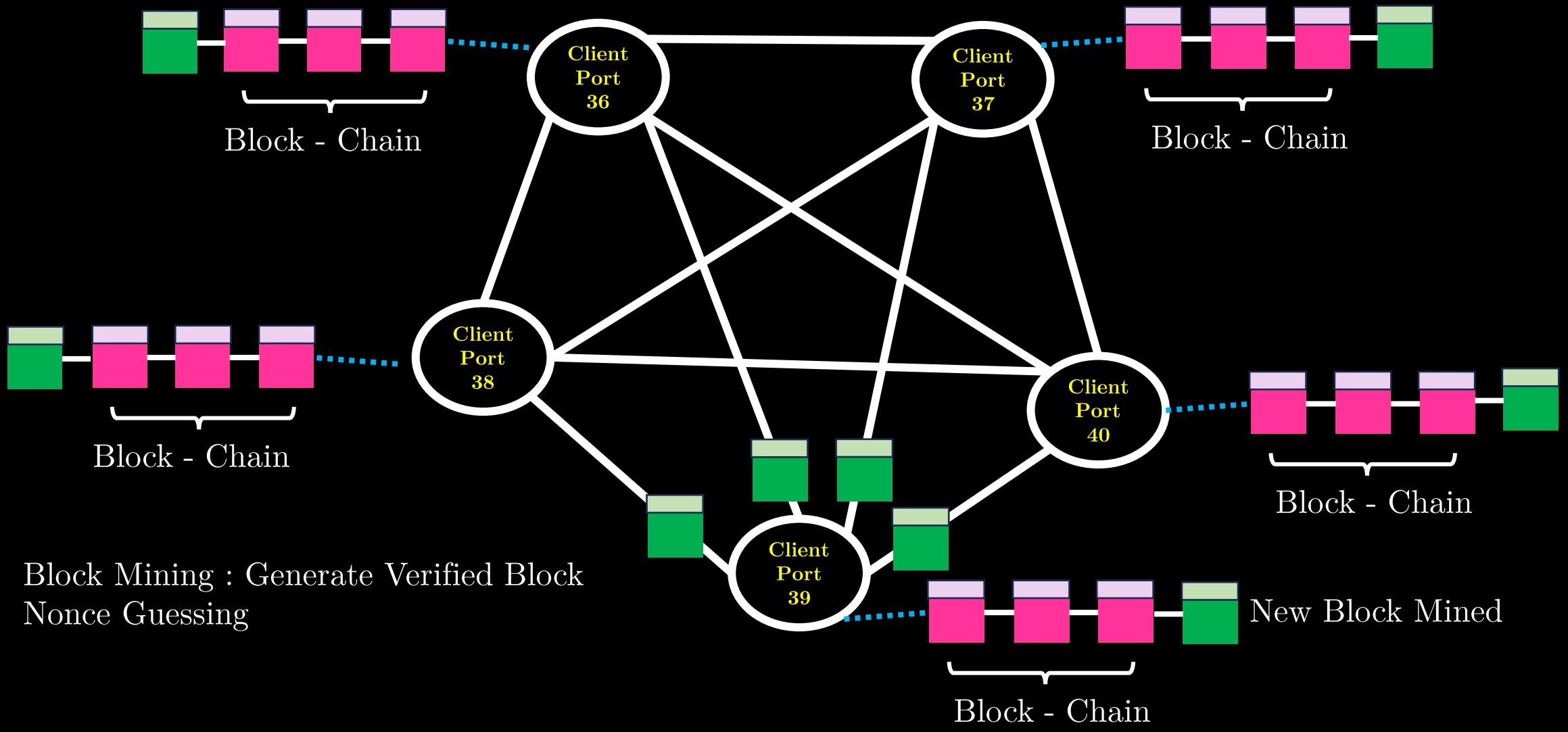


P2P-Network : Broadcasting



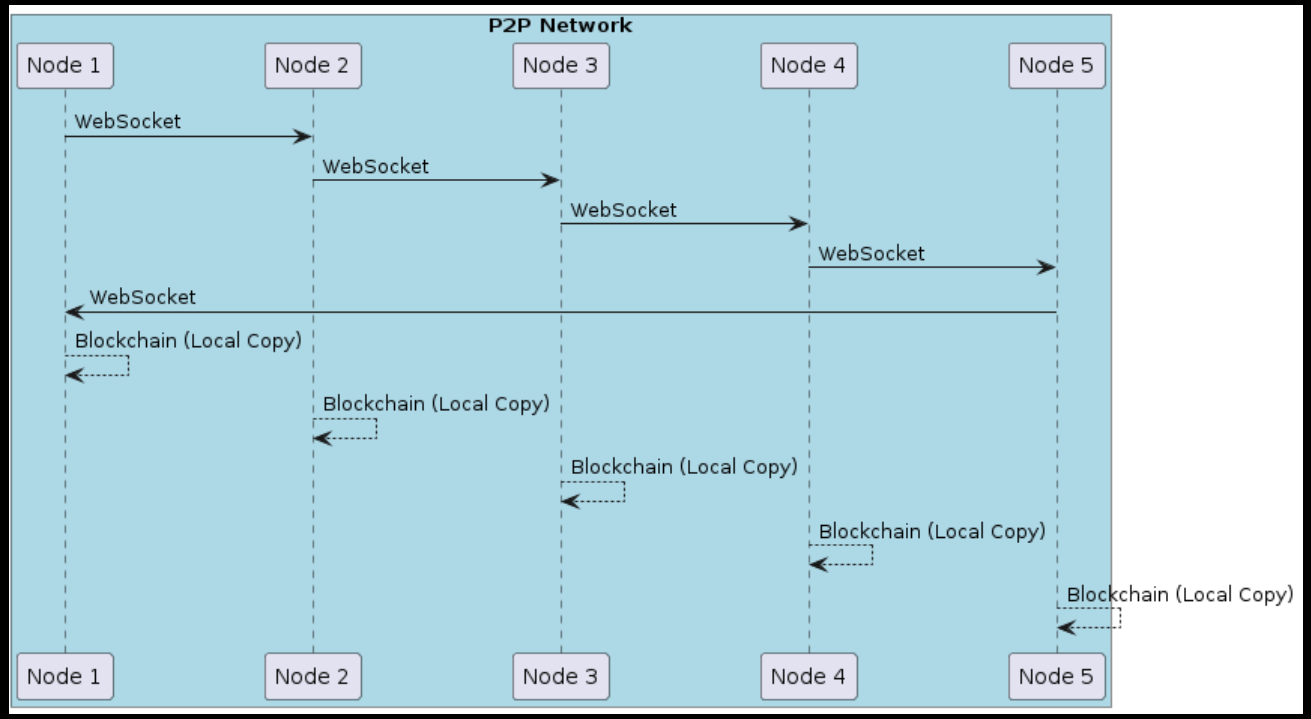
P2P-Network : Broadcasting

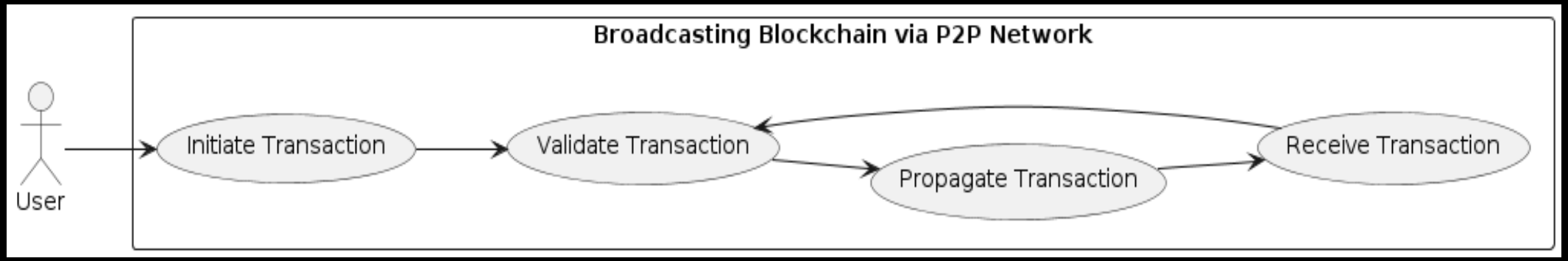
Broadcasting : /Level Transverse ->
Breadth First Search



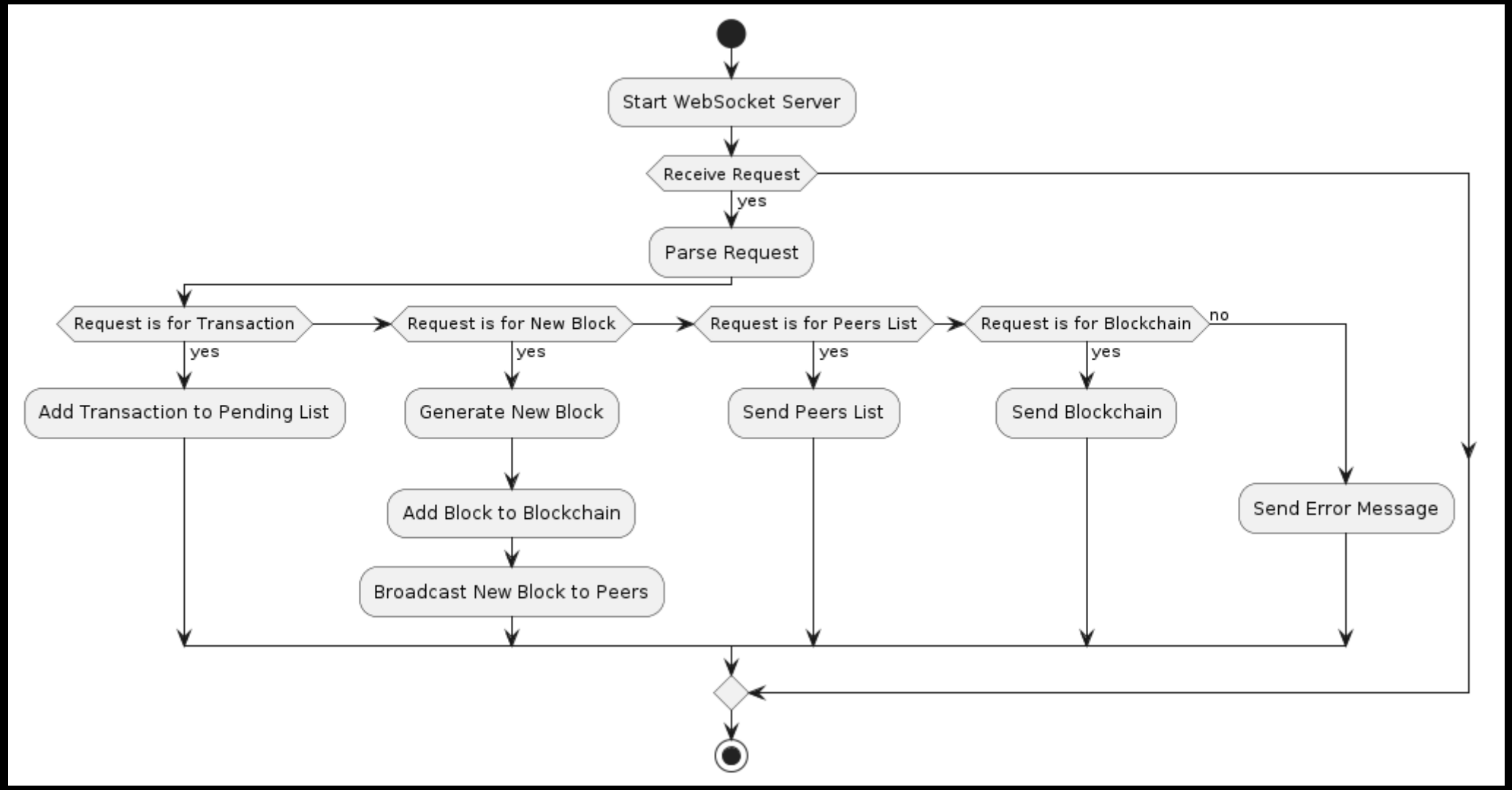
Block Mining : Generate Verified Block
Nonce Guessing

Use Case Diagram





FlowChart


Start WebSocket Server
Receive Request
yes
Parse Request
Request is for Transaction
yes
Add Transaction to Pending List
Request is for New Block
yes
Generate New Block
Add Block to Blockchain
Broadcast New Block to Peers
Request is for Peers List
yes
Send Peers List
Request is for Blockchain
yes
Send Blockchain
no
Send Error Message

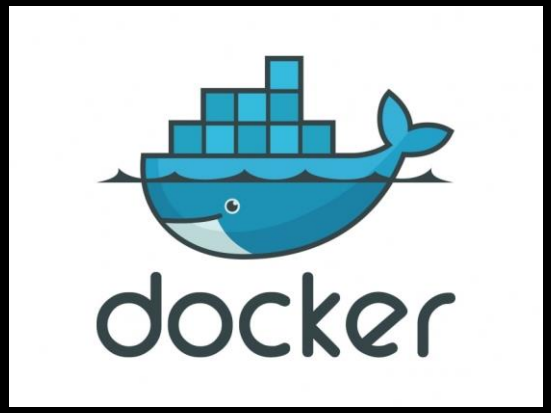
Method & Requirements

Methods :

1. Hashing (SHA256 + Randomize Algorithm)
2. Recursive
3. P2P Network – Broadcasting, WebSocket
4. Asynchronous / Multi-Threading
5. Graph Transverse : BFS / Level Ordering

Requirements :

1. Docker Container for Client Topology Simulation
2. Python Libraries :




```
import hashlib
import asyncio
from websockets.server import serve
from websockets.sync.client import connect
```