МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра програмних систем і технологій

Курсова робота

Розробка блокчейн вузла у якості автоматизованої системи безготівкових платежів

Виконав:

ст. гр. ІПЗ-33 Гоша Давід

Науковий керівник:

д.т.н. с.н.с Порєв Геннадій Володимирович

Київ – 2023

Анотація

У цій курсовій роботі розглядається аналіз та вирішення проблем, пов'язаних із системами на основі блокчейну, з особливим акцентом на парадокс Пуассона в контексті алгоритму доказу роботи (Proof-of-Work, PoW). Виконано порівняльний аналіз існуючих аналогів, включаючи різні технології блокчейну та їх застосування.

Детально вивчені ключові концепції, включаючи Proof-of-Work, Proof-Of-Elapsed-Time, майнінг, баланс, транзакції та модель UTXO, а також розглянуто підхід до гібридного протоколу консенсусу. Виявлено основні проблеми в платіжних системах на основі блокчейну, зокрема виклики, пов'язані з атаками Sybil та середовищем довіреного виконання (TEE).

Проведено аналіз парадоксу Пуассона, сформульовано гіпотезу та проаналізовано результати моделювання. На основі цього аналізу запропоновано рішення, що включає проектування архітектури системи та визначення кореляції між проблемами та рішеннями.

Описано процес реалізації програмного забезпечення: створення програмного продукту, проектування класів та об'єктів, розробка інтерфейсу користувача, програмного модуля та інструкції для користувача. Робота слугує важливим кроком у розумінні та вдосконаленні платіжних систем на основі блокчейну.

ЗМІСТ

1. Вступ

У сфері технології блокчейн та її застосувань, що постійно розширюється, одним з найважливіших питань, що викликають інтерес, є її потенціал для революційної перебудови платіжних систем. Однак, існуючі моделі стикаються з рядом обмежень і проблем, які перешкоджають їх широкому впровадженню і використанню. Ця курсова робота мотивована необхідністю вирішення цих проблем, прокладаючи шлях до більш ефективної та надійної платіжної системи на основі блокчейну.

Основною метою даної роботи є дослідження та розробка прототипу системи, яка перевершує існуючі аналоги в критичних аспектах. Ця система покликана забезпечити вирішення загальних проблем, що спостерігаються в поточних проектах, таких як масштабованість, швидкість транзакцій, енергоспоживання і безпека, тим самим демонструючи інноваційний підхід в середовищі блокчейн-технологій.

У цій роботі досліджується процес моделювання та експериментів для демонстрації переваг запропонованого прототипу системи. У ній зроблена спроба розробити відповідний критерій для формалізації поліпшень системи, виміряних за допомогою конкретних метрик, таким чином забезпечуючи науковий результат, який обґрунтовує запропоновані рішення.

Усуваючи виявлені прогалини в існуючих системах і пропонуючи реалістичні рішення, це дослідження сподівається зробити значний внесок у сферу технології блокчейн. Потенційний вплив цих удосконалень охоплює більш ефективну систему транзакцій, підвищену масштабованість, посилену безпеку і зниження енергоспоживання, що розширює межі можливого на сьогоднішній день.

Ця робота складається з основних розділів, в яких розглядаються основні концепції технології блокчейн, дослідницький підхід, математичне моделювання системи та програмна реалізація прототипу. Наприкінці наводяться підсумки

отриманих результатів та їх значення в більш широкому контексті платіжних систем на основі блокчейну.

1.1 Мета роботи

Мета цьої курсової роботи полягає у вирішенні проблем, які заважають сучасному стану платіжних систем на основі блокчейну. Рушійною силою цих зусиль є підвищення стійкості, надійності та загальної продуктивності таких систем шляхом впровадження інноваційної концепції стану блокчейну та нового гібридного механізму консенсусу.

Основна мета полягає в концептуалізації та розробці моделі системи, яка втілює ці вдосконалення. Завдяки ретельному процесу глибоких досліджень, інноваційного мислення, розробки програмного забезпечення та вичерпного тестування, ми прагнемо створити платіжну систему на основі блокчейну, здатну суттєво зменшити затримку транзакцій та уникнути проблем централізації, які зазвичай асоціюються з сучасними впровадженнями блокчейну.

Особливістю нашого проекту є використання гібридного механізму консенсусу - революційної інновації, яка об'єднує доказ роботи (PoW)[1] і доказ часу, що минув (PoET)[10]. Очікується, що це об'єднання підтримає децентралізацію, одночасно забезпечуючи ефективну перевірку транзакцій - баланс, який залишається недосяжним в сучасних блокчейн-системах.

Ця курсова робота актуальним в епоху, коли технологія блокчейн набуває широкого поширення в різних секторах. Вивчаючи притаманні їй проблеми та пропонуючи життєздатні рішення, це дослідження має на меті зробити значний внесок у розвиток галузі, пропагуючи використання більш надійних, ефективних та справедливих систем блокчейн у практичному застосуванні.

1.2 Актуальність дослідження

В останні роки популярність криптовалют зростає в геометричній прогресії.

Однак, оскільки індустрія продовжує розширюватися, проблеми, пов'язані з існуючими технологіями, стають все більш очевидними. Основними проблемами сучасного ринку є високі транзакційні витрати в таких мережах, як Ethereum, та значні затримки транзакцій в таких мережах, як Bitcoin. Актуальність цього дослідження полягає в тому, що в ньому розглядається новий підхід до вирішення цих проблем, що значно підвищує продуктивність і покращує користувацький досвід платіжних систем, заснованих на блокчейні.

Крім того, зростаюче занепокоєння у криптовалютному просторі викликає централізація мереж у майнінг-пули[4]. Така централізація підриває одну з фундаментальних філософій технології блокчейн - децентралізацію. Запропонований у цьому дослідженні гібридний протокол консенсусу має на меті вирішити цю проблему, сприяючи децентралізації та підвищенню безпеки і надійності блокчейну.

Це дослідження є не тільки своєчасним, але й дуже актуальним, оскільки воно пропонує потенційні рішення поточних проблем і робить свій внесок у дискусію про майбутній розвиток технології блокчейн. У зв'язку зі стрімким зростанням і все більшим поширенням криптовалют, потреба галузі в більш ефективному, надійному і децентралізованому рішенні стає все більш нагальною.

Ця робота потенційно може змінити поточне середовище технології блокчейн, розширити межі можливого і прокласти шлях до нового покоління криптовалют. Отримані результати можуть вплинути на розвиток майбутніх проектів і відкрити нові напрямки досліджень у цій захоплюючій і швидкозростаючій галузі.

1.3 Порівняння аналогів

У цьому дослідженні пропонується новий гібридний протокол консенсусу, який намагається обійти ключові проблеми, що спостерігаються в наступних аналогах.

- **Litecoin**: однорангова криптовалюта, Litecoin була розроблена як

"полегшена версія Bitcoin". Вона має на меті обробляти блок кожні 2,5 хвилини (порівняно з 10 хвилинами у Біткоїна) і забезпечує більш швидке підтвердження транзакцій. Однак, як і Біткоїн, вона використовує алгоритм консенсусу Proof-of-Work (PoW), що може призвести до збільшення споживання енергії та затримки транзакцій у сценаріях з високим трафіком. На противагу цьому, запропонована система зменшує споживання енергії завдяки використанню гібридного протоколу консенсусу.

- **Dogecoin**: В основному використовується для отримання чайових в інтернеті, Dogecoin також використовує алгоритм PoW. Хоча він має швидший час обробки блоків, ніж Bitcoin і Litecoin, його залежність від PoW все ще викликає занепокоєння щодо масштабованості та енергоспоживання.

- **Ethereum**: Будучи другою за величиною криптовалютою, Ефіріум запровадив концепцію смарт-контрактів[5]. Тим не менш, Ethereum стикається з проблемами масштабованості і піддається критиці за високу комісію за транзакції. Мережа Ethereum також використовує консенсус PoW, що призводить до подібних проблем зі споживанням енергії та затримкою

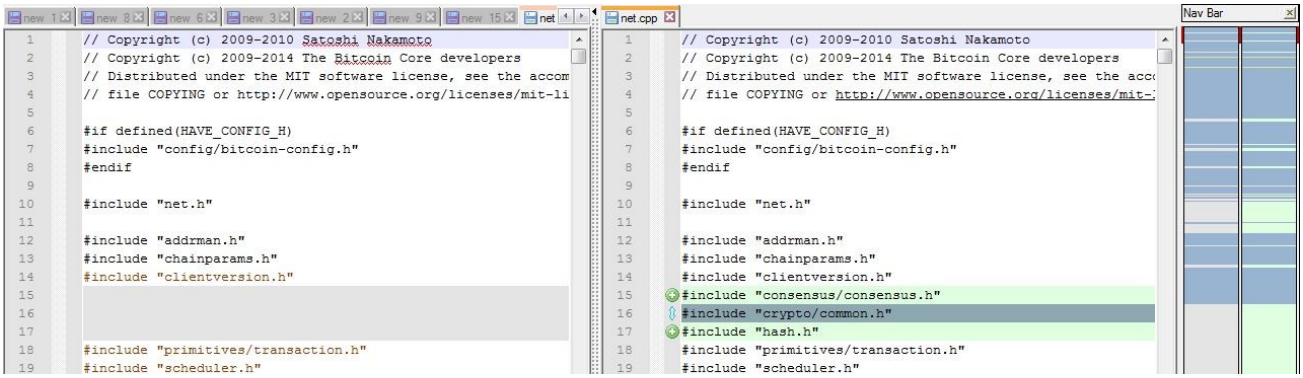


Рисунок 1 Порівняння вихідного коду net.cpp (ліворуч Dogecoin, праворуч Bitcoin) (1.1)

транзакцій. Ці проблеми вирішуються в запропонованій системі за допомогою гібридного протоколу консенсусу, спрямованого на зниження транзакційних витрат і поліпшення масштабованості.

- **Solana**: Solana - це високопродуктивний блокчейн, який обіцяє швидкі

та безпечні децентралізовані додатки та криптовалюти. Він використовує унікальну систему міток часу під назвою Proof of History (PoH) в поєднанні з механізмом консенсусу PoS (Proof of Stake). Однак, були висловлені занепокоєння з приводу централізації мережі. Запропонована система націлена на вирішення цієї проблеми, забезпечуючи децентралізацію за допомогою гібридного протоколу консенсусу.

- **Cardano**: Cardano використовує унікальний алгоритм PoS під назвою Ouroboros, який є менш енергоємним, ніж PoW. Хоча він пропонує більш енергоефективну альтернативу Ethereum, мережа все ще стикається з проблемами швидкості транзакцій і масштабованості. Запропонована нами система має на меті покращити ці аспекти за допомогою гібридного механізму консенсусу.

- **Polygon**: Polygon - це фреймворк для створення та підключення сумісних з Ethereum блокчейн-мереж. Вона спрямована на усунення обмежень Ethereum, включаючи пропускну здатність, поганий користувацький досвід (висока швидкість і затримка транзакцій) і відсутність суверенітету для розробників. Однак деякі критики вказують на можливі проблеми з централізацією та безпекою, пов'язані з моделлю консенсусу PoS.

Підсумовуючи, можна сказати, що хоча кожна з вищезгаданих платформ зробила цінний внесок у цю галузь, вони також стикаються з певними проблемами - такими як енергоефективність, затримка транзакцій, масштабованість і централізація. Запропонована нами система має на меті вирішити ці проблеми шляхом прийняття нового гібридного протоколу консенсусу. Цей підхід дозволяє нам використовувати сильні сторони існуючих систем, одночасно покращуючи їх слабкі сторони.

2. Теоретична інформація

У цьому розділі викладено теоретичні основи, що стосуються нашого дослідження, пропонуючи всебічний огляд існуючих технологій блокчейн і їх застосування, з подальшим вивченням гібридного протоколу консенсусу. Ці елементи забезпечують базові знання, необхідні для розуміння природи досліджуваної проблеми, запропонованого рішення і унікальних аспектів розробленої системи.

2.1 Існуючі технології блокчейн та їх застосування

Технологія блокчейн, з моменту її появи з біткоїном, зробила революцію в тому, як проводяться транзакції та зберігаються записи в децентралізованому, безпечному режимі. Однак система блокчейну значно еволюціонував, і нові технології пропонують унікальні підходи до масштабованості, безпеки та різноманітності застосувань[19].

1. Біткоїн: Піонер технології блокчейн, біткоїн, представив концепцію децентралізованих цифрових валют. Він спирається на механізм консенсусу Proof-of-Work (PoW) для підтвердження і запису транзакцій. Успіх біткоїна вплинув на наступні технології блокчейну, але він також має недоліки, пов'язані, насамперед, з масштабованістю та швидкістю транзакцій.

2. Ефіріум: Ethereum розширив концепцію блокчейну за межі простих транзакцій, запровадивши програмовані смарт-контракти. Це сприяло розвитку децентралізованих додатків (DApps) та первинних пропозицій монет (ICO). Однак Ethereum[8], як і Bitcoin, стикається з проблемами масштабування та високими комісіями за транзакції.

3. Litecoin і Dogecoin: спочатку представлені як альтернатива біткоїну, Litecoin і Dogecoin пропонують швидший час генерації блоків, тим самим прагнучи забезпечити швидке підтвердження транзакцій. Однак, швидший час

створення блоків може призвести до більшої ймовірності розгалуження.

4. Cardano: Cardano пропонує унікальну дворівневу архітектуру для відокремлення реєстру значень рахунків від причини, по якій значення переміщуються з одного рахунку на інший. Це розділення має на меті покращити функціональність смарт-контракту. Механізм консенсусу - Proof-of-Stake (PoS), який вважається більш енергоефективним, ніж PoW.

5. Solana: Solana впроваджує нову систему міток часу для підвищення ефективності мережі, яка має на меті обробляти тисячі транзакцій в секунду. Однак були висловлені занепокоєння щодо централізації.

6. Polygon (Matic): Як рішення для масштабування поза ланцюжком для Ethereum, Polygon забезпечує швидші та дешевші транзакції. Тим не менш, були виявлені проблеми з безпекою, пов'язані з його механізмом вибору валідатора.

З точки зору застосування, технології блокчейн розгортаються у сферах, що виходять далеко за межі криптовалют. Децентралізовані фінанси (DeFi), відстеження ланцюжків поставок, цифрова ідентифікація особи, системи голосування та не взаємозамінні токени (NFT) - це лише кілька прикладів трансформаційного потенціалу технологій блокчейн.

Ці технології дають цінну інформацію та слугують важливими орієнтирами для розробки запропонованої платіжної системи, яка має на меті поєднати сильні сторони та пом'якшити недоліки існуючих рішень. Наступний розділ присвячений одному з таких інноваційних підходів: гібридному протоколу консенсусу.

2.2 Proof-Of-Work

У цьому розділі обговорюється реалізація консенсусу в системі блокчейн Біткоїн. Сатоші Накамото у своєму документі про Біткоїн посилався на систему Hashcash Адама Бека, яка вперше представила алгоритм Proof of Work (PoW) як універсальну технологію захисту від спаму.

Концепція, що лежить в основі, проста: Якщо вузол повинен виконати певну обчислювальну роботу, перш ніж підтвердити блок, йому буде невигідно атакувати мережу тисячами транзакцій в секунду. Хоча ця технологія може використовуватися в інших системах для запобігання спаму, в контексті Біткоїна консенсус відіграє життєво важливу роль у рівномірному розподілі емісії монет і виборі лідера для додавання нового блоку до блокчейну.

Біткоїн використовує алгоритм хешування SHA-256[11]. Він бере набір транзакцій у блоці і повертає 256-бітний хеш. Встановивши правило, що мережа прийматиме хеш лише з певною кількістю початкових нулів, можна збільшити складність пошуку відповідного хешу. Відповідно, це зменшує діапазон прийнятних хешів і збільшує час, необхідний для хешування.

Щоб гарантувати, що однакові вхідні дані не завжди дають однаковий хеш, було введено поняття, яке називається "nonce". Подаючи випадкові дані (nonce) разом з транзакціями на функцію хешування, можна згенерувати різні хеші для одного і того ж блоку. Вузли будуть продовжувати хешування з різними nonce до тих пір, поки один з них не знайде хеш в прийнятному діапазоні.

Обчислювальна потужність мережі, яка залежить від кількості вузлів-учасників, є динамічною. Якщо кількість майнерів подвоюється, швидкість підтвердження блоків також подвоюється, що прискорює процес підтвердження, але потенційно призводить до збільшення навантаження на мережу і швидкості емісії монет. Щоб запобігти цьому, мережа коригує рівень складності приблизно

кожні 2016 блоків, або приблизно кожні два тижні, щоб підтримувати середній час підтвердження блоку на рівні 10 хвилин.

Форки, або розгалуження в блокчейні, можуть виникати, коли два різних блоки одночасно знаходять правильний хеш. У цій ситуації мережа фактично розділяється на дві частини, кожна з яких продовжує свою гілку. Ця проблема вирішується дотриманням правила "найдовший ланцюжок перемагає": та гілка, яка першою додає наступний блок, визнається головною гілкою. Вузли, які працювали на коротшій гілці, повинні перейти на нову головну гілку. Тому для забезпечення підтвердження транзакції важливо дочекатися більш ніж одного підтвердження від мережі.

2.3 Proof-Of-Elasped-Time

Proof of Elapsed Time (PoET) - це алгоритм консенсусу, який використовується в системах блокчейн, зокрема на платформі Intel Sawtooth Lake[12]. PoET розроблений для забезпечення справедливого і високомасштабованого процесу підтримки консенсусу в децентралізованій мережі, при цьому пом'якшуючи деякі з істотних проблем споживання ресурсів, пов'язаних з іншими алгоритмами консенсусу.

Основа PoET відносно проста. Мета полягає в тому, щоб визначити легітимність і порядок транзакцій в децентралізованій системі, що є критично важливим для будь-якої мережі блокчейн. Замість того, щоб покладатися на величезні обчислювальні потужності, як в Proof of Work (PoW), або володіння великою часткою в мережі, як в Proof of Stake (PoS), PoET використовує систему випадкової лотереї для вибору вузла, який додає наступний блок до ланцюжка.

В алгоритмі PoET кожен вузол, що бере участь в мережі, генерує випадковий час очікування і засинає на цей час. Вузол, який прокидається першим - тобто вузол з найкоротшим часом очікування - додає новий блок до блокчейну і транслює його решті мережі. Цей процес повторюється для додавання кожного нового блоку. Це

наче кожен вузол - поет, який чекає на натхнення; той, хто прокинеться першим, напише наступний рядок "поеми", якою є блокчейн.

Критично важливим аспектом PoET є забезпечення цілісності часу очікування. Для цього PoET використовує розширення Intel Software Guard Extensions (SGX), які дозволяють програмам запускати надійний код у захищених контейнерах, відомих як анклави. SGX гарантує, що код, який генерує випадковий час очікування і спить протягом цього часу, працює, як очікувалося, і не був підроблений, тим самим забезпечуючи чесність лотерейної системи.

PoET має кілька переваг як алгоритм консенсусу. Він є енергоефективним, оскільки вузлам не потрібно виконувати обчислювально інтенсивні завдання, і вони можуть переходити в режим сну з низьким енергоспоживанням під час очікування. PoET також підтримує високий ступінь масштабованості, оскільки додавання нових вузлів до мережі не призводить до значного збільшення обчислювальної потужності, необхідної для досягнення консенсусу. Нарешті, PoET сприяє справедливості, оскільки кожен вузол, незалежно від його обчислювальної потужності або частки в мережі, має рівні шанси бути обраним для додавання наступного блоку.

Незважаючи на ці переваги, з PoET пов'язані також проблеми і критика. Він покладається на надійне середовище виконання, надане Intel SGX, що викликає занепокоєння щодо централізації та довіри. Крім того, він може стати вразливим, якщо зловмисник знайде спосіб скомпрометувати SGX або маніпулювати процесом генерації випадкових чисел.

На закінчення, Proof of Elapsed Time представляє унікальний та інноваційний підхід до питання консенсусу в мережах блокчейн. Поєднуючи елементи випадковості, справедливості та енергоефективності, він означає значний відхід від традиційних механізмів консенсусу, що вимагають значних ресурсів. Однак, як і всі технології, вона не позбавлена потенційних проблем і повинна постійно

перевірятися, тестуватися і розвиватися, щоб зменшити будь-які вразливості і підтримувати цілісність систем, які вона підтримує.

2.4 Майнінг

Майнінг нерозривно пов'язаний з механізмом консенсусу щодо доказів роботи. Через відсутність центрального органу влади в мережі за замовчуванням виникає проблема, відома як "проблема візантійських генералів". Це класичний виклик в криптографії, що передбачає прийняття рішень в потенційно ворожому середовищі. У Біткоін вона вирішується шляхом випадкового вибору вузла, блок якого визнається дійсним всією мережею.

По суті, учасники мережі, включаючи майнерів, грають у своєрідну криптографічну лотерею. Вони перевіряють цілісність блоку, порівнюючи його хеш з хешами інших вузлів; якщо в блоці є навіть незначні зміни, хеш буде кардинально відрізнятися, що робить подальші зусилля з майнінгу марними, оскільки мережа відхилить такий блок.

Після перевірки блоку всі учасники намагаються знайти відповідний хеш за допомогою nonce. Як тільки такий хеш знайдено, успішний вузол отримує винагороду від бази монет і право додавати та розповсюджувати новий блок. Враховуючи, що пошук відповідного хешу є обчислювально інтенсивним завданням, майнеру, як правило, невигідно намагатися обдурити мережу; будь-які спроби змінити дані блоку, швидше за все, будуть помічені іншими вузлами, які потім відхилять блок майнера.

База монет - це спеціальна сутність в блокчейні, яка не має приватного ключа. Вона служить для виплати майнерам з власних резервів за їх внесок в мережу. База монет може бути відновлюваною або невідновлюваною.

У випадку з невідновлюваною базою монет

1. Майнер видобуває блок

2. Монетна база платить майнеру зі своїх резервів
3. Майнер також збирає частину комісії за транзакції

У випадку з відновлюваними монетними базами:

1. Майнер видобуває блок
2. Частина комісії за транзакції перераховується на монетну базу
3. Коін-чейн платить майнеру зі своїх резервів

Коли блок видобувається декількома майнерами одночасно, виникає конкуренція або "майнінгова гонка". Майнери з більшою обчислювальною потужністю мають більше шансів на перемогу. Однак координація цих зусиль для уникнення дублювання є складним завданням. Щоб вирішити цю проблему, створюються майнінг-пули, які об'єднують майнерів для роботи над однією проблемою і спільного використання ресурсів. Ці пули координують зусилля окремих майнерів, щоб вони не дублювали роботу один одного.

2.5 Баланс

Топологія мережі, з якою ми маємо справу, є децентралізованою одноранговою, оскільки приклад, який ми розглядаємо, - це Біткоїн. Така мережева архітектура означає відсутність центральної точки управління. Така технологія породжує певні неоднозначні проблеми, такі як подвійні витрати. Проблема подвійних витрат була вирішена за допомогою ланцюжка цифрових підписів. Щоб зрозуміти це, нам потрібно абстрагуватися від традиційної моделі обміну фіатних грошей.

Біткоїн не має поля "баланс" або фізичних монет як таких. Існує лише один реєстр, який є ланцюжком усіх транзакцій. Звідси ми можемо математично визначити баланс вузла. По суті, це можна представити як ланцюжок передачі прав власності на частину загальної емісії валюти. Самі монети знаходяться в стані мономорфності[7] і існують виключно для спрощення людського сприйняття. Підсумкове поле балансу користувача ніде не з'являється, ця величина є

поліморфною.

При створенні наступної транзакції деякий сторонній вузол повинен перевірити реєстр - ланцюжок транзакцій, пов'язаних хешами попередніх транзакцій, щоб переконатися, що кількість входів більша або дорівнює кількості виходів. По суті, це перевірка, щоб підтвердити, чи має відправник достатньо коштів. Враховуючи, що один відправник може мати кілька вхідних і вихідних транзакцій, ми виводимо наступну формулу для визначення достовірності суми

переказу:

$$\sum_{k=0}^{i} i \geq \sum_{k=0}^{i} o \qquad (2.1)$$

, де і = сума всіх вхідних переказів, о = сумою вихідних переказів

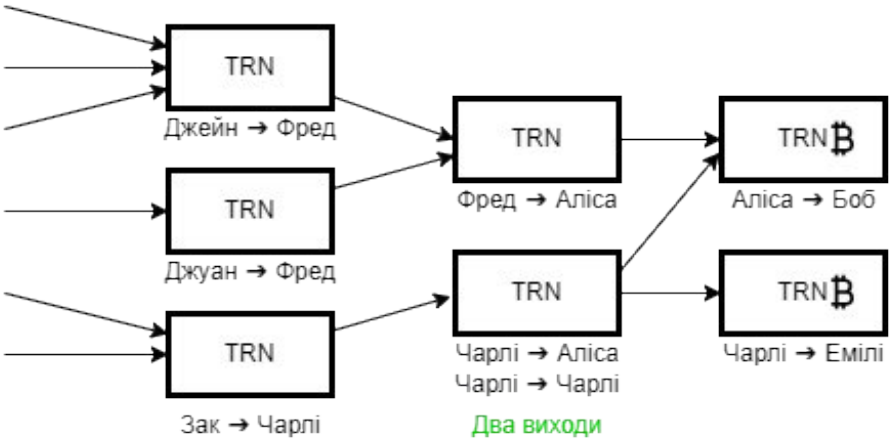


Рисунок 2 Спрощений ланцюжок транзакцій(2.2)

Блокчейн можна реалізувати різними способами, навіть якщо сфера застосування відома заздалегідь. Наприклад, якщо програма, що розробляється на основі блокчейну, є платіжною системою, то отримання балансу користувача вже може бути реалізовано двома способами: детермінованим і недетермінованим, не кажучи вже про склад блоку, обмеження блоку, винагороду за майнінг тощо. Така ситуація є більш негативною, оскільки безпека кінцевого продукту не буде визначатися загальноприйнятими стандартами, які пройшли відкритий і тривалий

аналіз. Щоб протистояти цьому фактору, поширеною практикою є дотримання стандартів де-факто, таких як біткоїн (який діє як класична платіжна система) та Ethereum (який діє як платформа для смарт-контрактів).

2.6 Транзакції та UTXO модель

Модель UTXO (Unspent Transaction Output) - це підхід до управління транзакціями без додаткової необхідності підтвердження права власності на кошти. Це означає, що при створенні переказу ми можемо розпоряджатися тільки всією частиною отриманих монет, пам'ятаючи про функціонал балансу. Оскільки фізичного поняття балансу не існує, це просто функція, яка рекурсивно відновлює кількість зобов'язань перед об'єктом по ланцюжку транзакцій. Таким чином, ви можете витратити залишок тільки повністю. Часткова витрата призведе до розгалуження гілок і неузгодженості в ланцюжку блокчейну. Механізм переказу монет працює наступним чином: об'єкт-відправник повинен переказати необхідну суму об'єкту-одержувачу, а якщо сума менша за залишок, то об'єкт повинен повернути залишок собі (див. рис. 2.2).

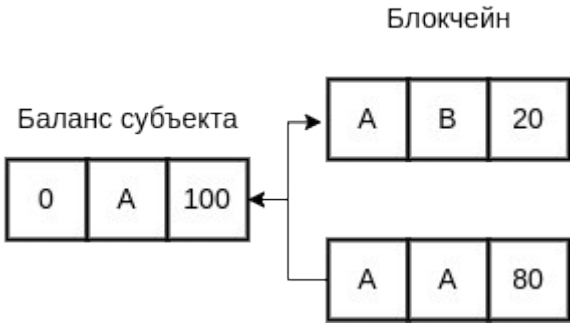


Рисунок 3 Специфікація UTXO транзакції(2.2)

Вирішення проблеми подвійних витрат є дуже важливим. Враховуючи, що в мережі немає посередника або валідатора, якому за замовчуванням довіряють всі без винятку вузли, виникає проблема з прив'язкою транзакцій. Відправник А може відправити користувачеві В транзакцію, еквівалентну його балансу, а потім відправити таку ж суму користувачеві С. Мережа повинна мати певний алгоритм і

ряд властивостей, щоб запобігти подібним типам атак. Для цього кожна транзакція містить посилання на попередню, тобто останню загальноприйняту на даний момент, і власний унікальний ідентифікаційний номер[11], який буде використовуватися для початку нових транзакцій в майбутньому. У Bitcoin такий ідентифікатор генерується за допомогою хешу самої транзакції і транзакції, формуючи таким чином спрямований список і захищаючи транзакцію від майбутніх змін.

Після цього транзакція потрапляє в пул пам'яті - місце, де вона вже потрапила в мережу, але ще не була записана в жодному з блоків, а отже, не була ефективно задекларована. Транзакція залишається в цьому стані до того моменту, поки її не буде додано до блоку. Блок - це колекція транзакцій розміром в один мегабайт. Блоки необхідні для оптимізації продуктивності мережі, оскільки підтвердження тисячі транзакцій одночасно є менш ресурсоємним і більш ефективним. Формування блоку починається зі створення реєстру - криптографічно підтвердженого впорядкування транзакцій, покликаного запобігти вразливості подвійних витрат. Після формування реєстру блок хешується за допомогою алгоритму дерева Меркла[9].

2.7 Вступ до гібридного протоколу консенсусу

Технологія блокчейн базується на протоколах консенсусу, які визначають, як підтверджуються транзакції і як додаються нові блоки до блокчейну. Традиційно блокчейн-платформи використовують єдині механізми консенсусу, такі як Proof of Work (PoW) або Proof of Stake (PoS). Однак кожен з цих механізмів консенсусу має свої сильні і слабкі сторони, що викликає інтерес до гібридних підходів, які намагаються об'єднати найкращі характеристики обох.

Гібридний протокол консенсусу об'єднує різні механізми консенсусу з метою використання їхніх переваг і мінімізації недоліків. У цьому дослідженні ми

зосередимося на гібридному протоколі, який поєднує в собі PoW і Proof of Elapsed Time (PoET).

- Доказ роботи (PoW): PoW - це оригінальний механізм консенсусу, запроваджений Біткоїном. У системі PoW майнери змагаються у вирішенні складної математичної задачі, і той, хто першим знайде рішення, отримує право додати наступний блок до блокчейну. Цей механізм забезпечує надійну безпеку, але є енергоємним і може призвести до збільшення часу виконання транзакцій.
- Підтвердження часу, що минув (PoET): PoET - це відносно новий механізм консенсусу, розроблений компанією Intel. Він розроблений як недорога, енергозберігаюча альтернатива PoW. У системі PoET мережа випадковим чином вибирає творця наступного блоку, виходячи з найменшого "часу очікування". Випадковість запобігає постійному додаванню нових блоків одним вузлом, тим самим заохочуючи децентралізацію.

Гібридний протокол консенсусу PoW/PoET має на меті отримати вигоду з безпеки і децентралізації PoW і енергоефективності та масштабованості PoET. Механізм працює таким чином, що PoW використовується для перевірки блоків, підтримуючи надійність системи. На противагу цьому, PoET використовується для створення блоків, забезпечуючи швидке і енергоефективне додавання блоків до ланцюжка.

Цей інноваційний протокол консенсусу є невід'ємною частиною дизайну і функціональності запропонованої платіжної системи, яка прагне вирішити проблеми масштабованості, швидкості транзакцій, енергоспоживання і централізації, які спостерігаються в існуючих технологіях блокчейн. Більш детальний аналіз того, як цей гібридний протокол консенсусу застосовується в архітектурі системи, представлений в розділі 5.1.

3. Ідентифікація проблеми

У сфері технології блокчейн, незважаючи на те, що її інноваційний потенціал широко визнаний, існує чимало перешкод, які стримують її ефективність та швидкість впровадження. Поява цих проблем вимагає глибокого дослідження для розуміння їх першопричини, впливу та можливих рішень. Це має вирішальне значення для розвитку і прогресу систем на основі блокчейну, особливо в контексті платіжних систем, де ці проблеми можуть мати ще більший вплив.

У цьому розділі ми визначаємо і заглиблюємося в критичні проблеми, з якими наразі стикаються платіжні системи на основі блокчейну. Аналізуючи ці проблеми та їх наслідки, ми створюємо міцну основу для розробки нашого рішення, спрямованого на вирішення цих проблем, тим самим підвищуючи функціональність і зручність використання технології блокчейн в платіжних системах.

3.1 Проблеми в платіжних системах на основі блокчейну

Незважаючи на інноваційний потенціал технології блокчейн, певні проблеми в існуючих платіжних системах на основі блокчейну обмежують їх ефективність та широке впровадження. У цьому розділі обговорюються ті ключові проблеми, на вирішення яких спрямований проект:

1. **Затримка транзакцій**: Традиційні платіжні системи на основі блокчейну, такі як Біткоїн[2], мають високу затримку транзакцій. Оскільки час підтвердження блоку в середньому становить близько 10 хвилин, а потенційно може бути набагато довшим, ця затримка робить ці системи непридатними для транзакцій в режимі реального часу, що перешкоджає їх використанню в повсякденній комерційній діяльності.

2. **Масштабованість**: Зі збільшенням кількості транзакцій мережа блокчейн може стати перевантаженою, що призводить до повільного виконання транзакцій і високих комісій за них. Масштабованість є значною проблемою для сучасних технологій блокчейн і нерозривно пов'язана з затримкою транзакцій.

3. **Ризики централізації**: В ідеалі, мережі блокчейн є децентралізованими і демократичними. Однак механізм консенсусу, який використовується в багатьох системах блокчейн, ненавмисно призвів до централізації, коли кілька потужних майнінг-пулів контролюють значну частину майнінгових потужностей мережі.

4. **Споживання енергії**: Майнінг, особливо в системах підтвердження роботи, вимагає значних обчислювальних ресурсів, що призводить до значного споживання енергії. Такий вплив на навколишнє середовище викликає занепокоєння щодо стійкості цих систем.

5. **Високі комісійні витрати**: Коли мережа стає перевантаженою, користувачі повинні платити вищу плату за пріоритетність своїх транзакцій. Це питання особливо актуальне для Ethereum, де високі тарифи на газ стали помітною проблемою.

Проект, з його новим підходом до архітектури блокчейну і механізму консенсусу, вирішує ці проблеми, прагнучи забезпечити масштабовану, ефективну і дійсно децентралізовану платіжну систему на основі блокчейну".

Зверніть увагу, що цей проект базується на загальних проблемах, виявлених в системах блокчейн, і може потребувати коригування, щоб ідеально відповідати специфіці вашого проекту.

3.2 Парадокс Пуассона та PoW

У сфері блокчейну механіка алгоритму консенсусу Proof-of-Work (PoW), особливо в поєднанні з блоком, змодельованим як пуассонівський процес, призводить до інтригуючого парадоксу. Це те, що ми називаємо парадоксом Пуассона, який є чудовим явищем, що додає складнощів роботі в системі блокчейн.

Щоб проаналізувати цей парадокс, нам потрібно викласти специфіку нашої моделі. Блок моделюється як пуассонівський процес зі швидкістю λ. Це означає, що генерація блоків відбувається за пуассонівським розподілом із середнім часом між блоками, позначеним як 1/λ, що визначається бажаним рівнем складності операції майнінгу PoW. У випадку Біткоїна це приблизно 10 хвилин.

Пуассонівський процес характеризується випадковими подіями, які слідують експоненціальному розподілу, з інтервалами між подіями, які є статистично ідентичними і незалежними одна від одної. Ми моделюємо процес майнінгу[14] як пуассонівський процес, фокусуючись виключно на моментах створення дійсних нових блоків.

Процес майнінгу в блокчейні PoW можна порівняти з азартною грою. Кожна спроба майнінгу схожа на підкидання монети, з дуже малим шансом на успіх. Ці спроби незалежні одна від одної і статистично ідентичні, що створює пуассонівський процес.

Наша модель робить кілька припущень. Ми припускаємо, що загальна кількість майнерів і їх колективна обчислювальна потужність є відносно постійною протягом певного періоду часу. Кожна машина майнера постійно намагається знайти правильні хеші - це єдині обчислення, які вона виконує. Таким чином, загальна кількість хешів, обчислених за одиницю часу, є постійною. У цій моделі ми припускаємо, що щосекунди обчислюється мільярд хешів.

Ми також враховуємо високий рівень складності хеш-пазла блоку-кандидата. Наприклад, якщо перші тридцять п'ять бітів хешу повинні бути нульовими для

того, щоб блок був дійсним, ймовірність того, що конкретний nonce буде відповідати цьому критерію, становить 2^-35 або приблизно 3 × 10^-11. Отже, ймовірність того, що будь-який майнер розгадає хеш-пазл за задану секунду, дорівнює 0,03 - відносно невелике число.

Ця модель справедлива незалежно від кількості майнерів, їх індивідуальних обчислювальних потужностей, а також від того, чи працюють різні майнери над одними і тими ж блоками, чи над різними. Коли загальна обчислювальна потужність змінюється, припущення про фіксовану швидкість майнінгу може не спрацювати. Однак, загальна обчислювальна потужність не змінюється стрибкоподібно. Тому протягом короткого проміжку часу швидкість приблизно постійна.

Незважаючи на те, що більшість блоків видобувається протягом 10 хвилин, випадковий характер процесу означає, що завжди будуть певні блоки, на видобуток яких майнери витрачають більше або менше часу. Це призводить до того, що середній час підтвердження транзакції коливається в межах 10 хвилин.

Парадокс полягає в тому, що, незважаючи на те, що середній час підтвердження становить близько 10 хвилин, більшість людей чекають довше. Вибірковий аналіз блоків Біткоїна показує, що 60% блоків видобуваються довше, ніж за 10 хвилин, тоді як лише 40% видобуваються менш ніж за 10 хвилин. Цей перекіс у бік більшого часу підтвердження пояснюється довгим хвостом розподілу Пуассона, що призводить до того, що ми називаємо парадоксом Пуассона[17].

На час підтвердження біткоїн-транзакцій можуть впливати ще кілька факторів, але вони не мають прямого відношення до парадоксу Пуассона. Вони включають час, необхідний для отримання і хешування транзакцій з пулу пам'яті в заголовку блоку, а також час затримки, якщо комісія, пов'язана з транзакціями, занадто низька.

Таке розуміння парадоксу Пуассона в контексті PoW допомагає зрозуміти тонкощі функціонування блокчейн-систем і сприяє подальшій розробці запропонованого нами рішення.

У процесі нашого дослідження ми розробили набір моделей, спрямованих на розуміння нюансів епох майнінгу блоків у блокчейні Біткоїн. Наше дослідження включало симуляції та обширні дані, зібрані з блокчейну, але важливо визнати наявність певних проблем, включаючи глобально невідому швидкість хешування, яка диктує швидкість виявлення блоків, та історичну, але випадкову величину складності майнінгу.

Крім того, хоча дані про час прибуття блоків є стійкими, їх не можна вважати повністю надійними. Ми ввели модель точкового процесу, де процес надходження блоків імітує неоднорідний пуассонівський процес між періодами зміни складності. Швидкість пропорційна відношенню швидкості хешування до складності, але залишається незалежною від зміни складності. Дискретизуючи процес, ми можемо точно визначити момент зміни складності, а отже, і зміну швидкості надходження блоків, якщо припустити глобальну швидкість хешування.

Проте, оскільки глобальна швидкість хешування залишається високою, механізм регулювання складності демонструє значну затримку, що призводить до приблизної швидкості надходження блоків, яка на 11,5% перевищує базову швидкість в шість блоків на годину. В результаті, пропускна здатність транзакцій і загальний дохід майнерів від винагород перевищують базові прогнози. Крім того, моменти зменшення винагороди за блок вдвічі і, зрештою, видобутку всіх біткоїнів, за прогнозами, відбудуться раніше, ніж це було б, якби блоки видобувалися зі швидкістю шість блоків на годину.

Окрім моделювання процесу надходження блоків, ми виявили зв'язок між частотою надходження блоків та експоненціальним зростанням швидкості хешування. Ми запропонували практичне наближення, яке описує поведінку межі,

незалежно від початкових умов і збурень процесу надходження блоків. Це наближення було підтверджено за допомогою симуляцій та даних з блокчейну.

Наше дослідження також підтвердило існування парадоксу Пуассона в контексті часу підтвердження транзакцій. Оскільки розподіл має довгий правий хвіст, більшість користувачів стикаються з довшим, ніж середній, часом підтвердження блоків. У той же час, меншість користувачів користуються швидкими підтвердженнями транзакцій. Розуміння цієї динаміки покращує наше розуміння тонкощів роботи блокчейн-систем, що є безцінним для подальшого розвитку та оптимізації цих платформ.

3.3 Вирішення проблем, пов'язаних з атаками Sybil та середовищем довіреного виконання (TEE)

3.3.1 Пом'якшення наслідків атак Sybil

Для подолання ризику атак Sybil, коли один зловмисник контролює декілька вузлів у мережі, ми пропонуємо наступні рішення:

Рейтинг довіри до мережі: Запровадження системи рейтингу довіри до мережі. Кожен вузол мережі матиме рейтинг довіри, який базуватиметься на його історичній поведінці, що включатиме такі фактори, як кількість оброблених транзакцій, внесок у підтримку мережі та відсутність підозрілих дій. Вищий рейтинг довіри забезпечить кращі мережеві привілеї, стимулюючи вузли до належної поведінки.

Гаранти: Впровадження механізм гарантів, коли добре відомі вузли ручаються за нові вузли. Нові вузли повинні будуть отримати схвалення від вузлів-гарантів, які перевірять їх надійність, перш ніж дозволити їм приєднатися до мережі.

Мережа Face-to-Face (F2F)[18]: Прийняття дизайну мережі F2F може запобігти атакам Sybil, гарантуючи, що тільки вузли, які перевірили один одного за допомогою реальних взаємодій, можуть підтверджувати свої транзакції. Хоча це

може спричинити проблеми в комунікації, але може значно підвищити безпеку мережі.

3.3.2 Вирішення проблем TEE

Середовища довіреного виконання (Trusted Execution Environments, TEE) можуть забезпечити безпеку, гарантуючи, що код виконується так, як він був написаний. Однак вони можуть бути вразливими до обману та інших зловмисних дій через вразливості в ізоляції або наявність помилок у реалізації. Для вирішення цих проблем ми пропонуємо

Регулярні оновлення та патчі безпеки: Суворий графік оновлення системи та патчів допоможе швидко виправити потенційні вразливості в TEO. Такий підхід вимагатиме активної команди безпеки та надійної стратегії управління виправленнями.

- **Багатосторонні обчислення**: Щоб гарантувати, що всі вузли правильно виконують алгоритм консенсусу Proof of Elapsed Time (PoET), ми можемо реалізувати протокол багатосторонніх обчислень. Це дозволить декільком вузлам перевірити правильність виконання коду без шкоди для конфіденційності.
- **Докази з нульовим знанням**: Ми можемо використовувати докази з нульовим знанням для перевірки того, що вузол правильно виконав механізм PoET. Це дозволяє вузлу довести, що код був виконаний правильно, не розкриваючи ніякої додаткової інформації, що допомагає підтримувати конфіденційність і безпеку мережі.

Це проактивні заходи для зменшення вразливостей, пов'язаних з механізмами консенсусу PoET і TEE, що забезпечують більш безпечну і надійну систему. Вони потребують постійного перегляду і оновлень, що відображають досягнення в області кібербезпеки і технології блокчейн.

Впроваджуючи ці стратегії, ми можемо підвищити надійність і достовірність нашої криптовалютної системи, заснованої на блокчейні, тим самим збільшуючи її корисність і практичну значущість.

4. Аналіз парадоксу Пуассона

Під час роботи з технологією блокчейн і, зокрема, з Біткоїном, можна спостерігати цікаве явище. Враховуючи, що система Біткоїн розроблена таким чином, що новий блок створюється приблизно кожні 10 хвилин, було б логічно припустити, що середній час очікування на підтвердження блоку буде коливатися навколо цієї 10-хвилинної позначки. Однак, як свідчать користувачі, багато користувачів повідомляють, що час очікування часто перевищує цей середній 10-хвилинний показник.

4.1 Гіпотеза

Парадокс, з яким ми тут стикаємося, зазвичай називають парадоксом Пуассона. Цей парадокс, що ґрунтується на властивостях розподілу Пуассона, як відомо, призводить до контрінтуїтивних результатів у різних ситуаціях, зокрема, коли йдеться про час очікування і швидкість обслуговування.

У випадку з біткоїном парадокс Пуассона можна виразити через наступну гіпотезу: *Незважаючи на те, що середній час підтвердження блоку встановлений на рівні приблизно 10 хвилин, більшість користувачів в кінцевому підсумку чекають на підтвердження блоку більше 10 хвилин*[6].

У цьому розділі звіту ми зануримося в цю гіпотезу глибше. Ми прагнемо математично описати цей парадокс, провести аналіз на основі даних блокчейну Біткоїна і пояснити реальні наслідки цього особливого явища в екосистемі блокчейну.

Наш аналіз сприятиме кращому розумінню динаміки блокчейн-систем і допоможе нам у розробці та впровадженні ефективних рішень на основі блокчейн-

технологій. Вивчення цього парадоксу є не просто теоретичною вправою; він має глибокі наслідки для того, як користувачі взаємодіють з технологією блокчейн і як розробники проектують системи на основі блокчейну. Розуміння цього явища має вирішальне значення для покращення користувацького досвіду, зміцнення довіри та прийняття блокчейн-систем.

4.2 Математична модель

Намагаючись дослідити, чи надходження блоків слідує пуассонівському процесу, ми розуміємо, що глобальна швидкість хешування H(t), емпірично або параметрично змодельована, визначає швидкість хешування для кожного надходження блоків за допомогою вибіркового стохастичного процесу, позначеного як $X_i(t)$.

Давайте розберемо цей сценарій на три ключові ситуації:

- **Ситуація 1**: Детерміноване коригування складності

У випадку детермінованого коригування складності, коригування виконується в детерміновані моменти часу, позначені через y_n, які не збігаються зі стохастичними моментами надходження блоків. Отже, швидкість надходження блоків, λ(t), залишається нечутливою до надходжень блоків на попередньому відрізку. За відсутності затримки модель узгоджується з неоднорідним пуассонівським процесом в межах кожного сегмента.

- **Ситуація 2**: Адаптація до стохастичної складності

Коли ми переходимо до стохастичного коригування складності, складність коригується в довільні моменти часу, після кожного сегмента 2016 блоків, дотримуючись попередньо визначеного рівняння. У сценарії, позбавленому затримки поширення, кожен сегмент процесу має форму неоднорідного пуассонівського процесу зі швидкістю λ(t) = H(t)/Di. Враховуючи, що швидкість надходження блоків, λ(t), залежить від початкового і кінцевого надходження блоків в попередньому сегменті блокчейну, процес не відображає пуассонівський

розподіл для послідовних часових періодів сегмента.

- **Ситуація 3**: Затримка поширення

При наявності затримки розповсюдження процес надходження блоків навіть не імітує неоднорідний пуассонівський процес в межах одного сегмента.

У наступному розділі ми більш детально розглянемо ці сценарії і висвітлимо складнощі та особливості процесу прибуття блоків в технології блокчейн. Порівняємо їх моделювання до даних про позначку часу з блокчейну біткоїна.

Отже, ми не будемо враховувати динамічну складність і змоделюємо сегмент з 2016 блоків, що дорівнює приблизно місяцю в реальному часі, або 20160 хвилин. Ми також припускаємо, що розподіл блоків у мережі відбувається миттєво, без затримок.

Точковий процес N є процесом Пуассона на ℝ, якщо він має наступні дві властивості.

1) Випадкова кількість точок N([a, b)) точкового процесу N, розташованих в обмеженому інтервалі [a, b) ⊂ R, є пуассонівською випадковою величиною із середнім Λ([a, b)), де Λ є невід'ємною мірою Радона.

2) Кількість точок точкового процесу N, розташованих на k інтервалах [a1, b1), . . . , [ak, bk) утворюють k незалежних пуассонівських випадкових величин із середніми Λ([a1, b1)), . . . ,Λ([ak, bk)).

Відтепер будемо записувати N([a, b)) як N(a, b) і Λ([a, b)) = Λ [a, b) для зручності. Перша властивість передбачає що

$$\mathbb{P}\left(N\left(a,b\right)=n\right)=\frac{\Lambda(a,b)^{n}e^{-\Lambda(a,b)}}{n!} \qquad (4.1)$$

і $E\left[N\left(qb\right)\right]=\Lambda\left(a,b\right)$, а друга властивість – це Основна причина придатності процесу точки Пуассона і зазвичай це основа статистичних тестів, які вимірюють адекватність моделей Пуассона. Розподіл Пуассона N(a, b) означає, що його

дисперсія Var[N(a, b)] = Λ(a, b), факт який також використовується як статистичний тест. Міра Λ відома як міра інтенсивності або середнє значення міри процесу точки Пуассона. Припустимо, що а існує така функція λ(t), що

$$\Lambda(a,b) = \int_a^b \lambda(t)\,dt \qquad (4.2)$$

Тоді λ(t) визначена як функція швидкості. Якщо λ(t) є сталою λ > 0, то процес називається однорідним точковим процесом Пуассона. Інакше процес називають неоднорідним або **неоднорідним точковим процесом Пуассона[7]**. Якщо обмежити нашу увагу інтервалом невід'ємних чисел [0, ∞), міра інтенсивності задається формулою

$$\Lambda(t) := \Lambda([0,t]) = \int_0^t \lambda(t)\,dt \qquad (4.3)$$

Для пуассонівського процесу N з мірою інтенсивності Λ ймовірність існування n точок в інтервалі [a, b) дорівнює

$$\mathbb{P}(N(a,b) = n) = \frac{[\Lambda(b)-\Lambda(a)]^n e^{-[\Lambda(b)-\Lambda(a)]}}{n!} \qquad (4.4)$$

Час надходження та час між надходженнями: розглянемо точковий процес $\{X_{(i)}\}_{i\geq 1}$, визначений на невід'ємних дійсних числах із майже напевно кінцевою кількістю точок у будь-якому обмеженому інтервалі. Тоді ми можемо інтерпретувати точки процесу як часи добування нових блоків та розмістити їх у порядку зростання, $X_1 \leq X_2 \leq \ldots$. Тоді відстані між сусідніми точками дорівнюють $T_i := X_i - X_{i-1}$ для i = 2, 3, . . . і $T_1 = X_1$. Випадкові величини T_i відомі як час очікування або час між надходженнями. Для однорідного процесу Пуассона зі швидкістю λ відповідні часи між надходженнями є незалежними та однаково розподіленими експоненціальними випадковими величинами із середнім значенням 1/λ

$$\mathbb{P}(T_k < t) = 1 - e^{-\lambda t} \qquad (4.5)$$

Де властивість експоненціального розподілу без пам'яті було використано. Це

не стосується неоднорідного точкового процесу Пуассона з інтенсивністю λ(t), де перший час між надходженнями $T_1 = X_1$ має розподіл

$$\mathbb{P}(T_1 \le t_1) = 1 - e^{-\int_0^{t_1} \lambda(s)ds} \qquad (4.6)$$

За першого часу очікування $T_1 = t_1$ умовний розподіл другого часу очікування T_2 є

$$\mathbb{P}(T_2 \le t_2 | T_1 \le t_1) = 1 - e = \int_{t_1}^{t} \lambda(s)ds \qquad (4.7)$$

і так далі для $k \ge 2$

$$\mathbb{P}(T_k \le t_k | T_{k-1} \le t_{k-1}) = 1 - e^{\int_{t_{k-1}}^{t_k} \lambda(s)ds} \qquad (4.8)$$

Можна показати, що k-й час надходження X_k має розподіл

$$\mathbb{P}(x_k \le t) = e^{-\Lambda(t)} \sum_{n=k}^{\infty} \frac{\Lambda(t)^k}{k!} \qquad (4.9)$$

З щільністю

$$fx_k(t) = \frac{\lambda(x) \wedge (t)^{k-1}}{(k-1)!} e^{-\Lambda(t)} \qquad (4.10)$$

Умова на n точок $\{U_i\}_{i=0}^{n}$ пуассонівського процесу, що існує в деякому обмеженому інтервалі [0, t]. Ми називаємо ці точки умовним часом надходження блоку. Якщо процес Пуассона є однорідним, то умовні часи надходження рівномірно і незалежно розподілені, утворюючи n рівномірних випадкових величин на [0, t]. Ця різниця між часом очікування T_i та умовним часом надходження U_i відіграє роль у тесті Пуассона.

Для неоднорідного пуассонівського процесу кожна точка U_i незалежно розподілена на інтервалі [0, t] із розподілом

$$\mathbb{P}(U_i \le u) = \frac{\Lambda(u)}{\Lambda(t)}, u \in [0,t] \qquad (4.11)$$

Якщо розподіл кожного U_i відомий і оборотний, то кожен U_i може бути

перетворений в рівномірну випадкову величину на [0, 1], що призводить до n незалежних рівномірних випадкових величин. Іншими словами, Λ(t) перетворює процес Пуассона на однорідний процес Пуассона з густиною один на відрізку дійсних чисел. Отже, статистичні методи для неоднорідних процесів Пуассона часто передбачають перетворення даних перед виконанням аналізу.

4.2 Аналіз результатів моделі

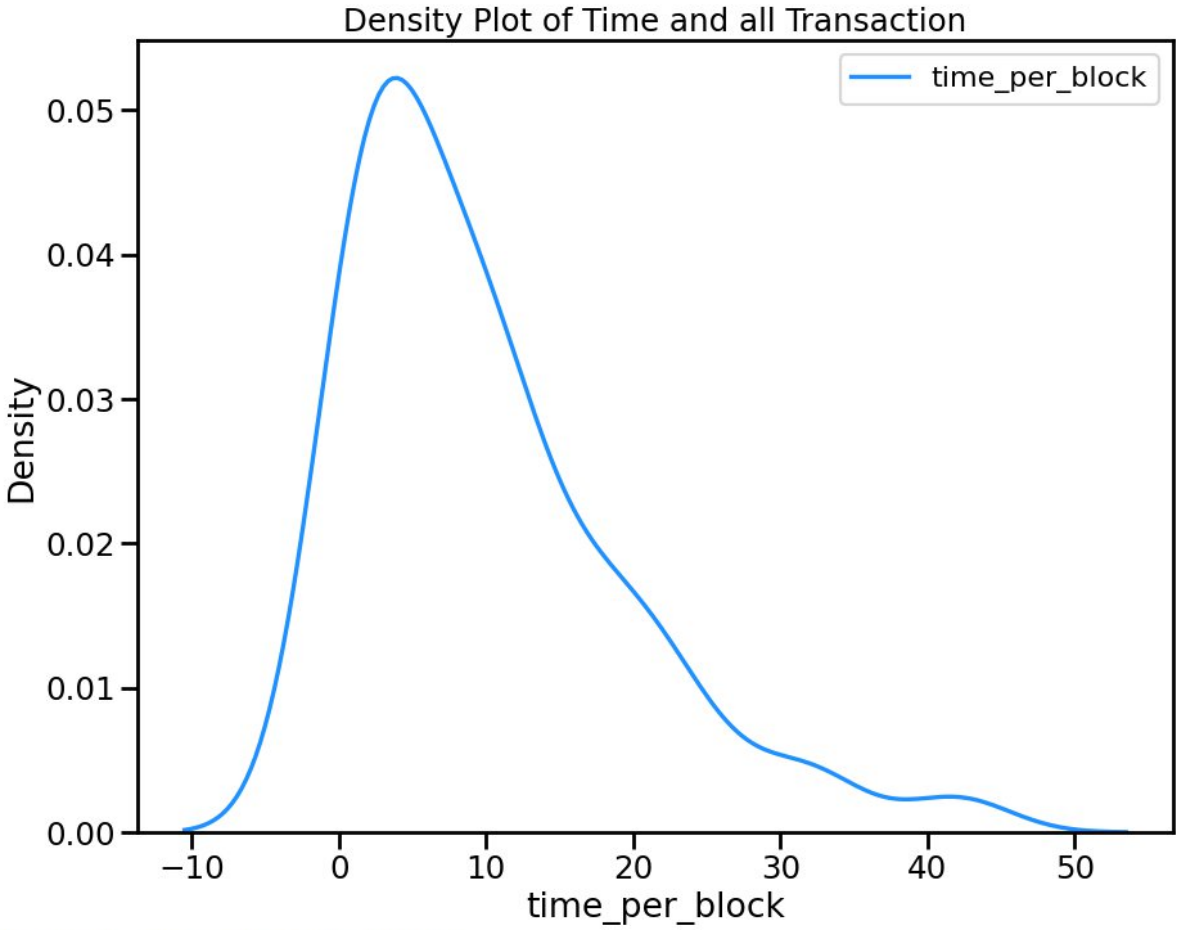


Рисунок 4 Графік щільності часу в залежності від кількості транзакцій(4.1)

Підводячи підсумок, швидкість видобутку біткоїна підпорядковується пуассонівському розподілу, що означає, що більшість блоків видобувається протягом 10-хвилинного періоду. Однак, враховуючи випадкову природу біткоїна, завжди будуть певні блоки, на видобуток яких майнерам знадобиться більше або менше часу. Отже, середній час підтвердження транзакції повинен коливатися в

межах 10 хвилин, як було продемонстровано вище.

Парадокс - це твердження, яке на перший погляд здається абсурдним, але при

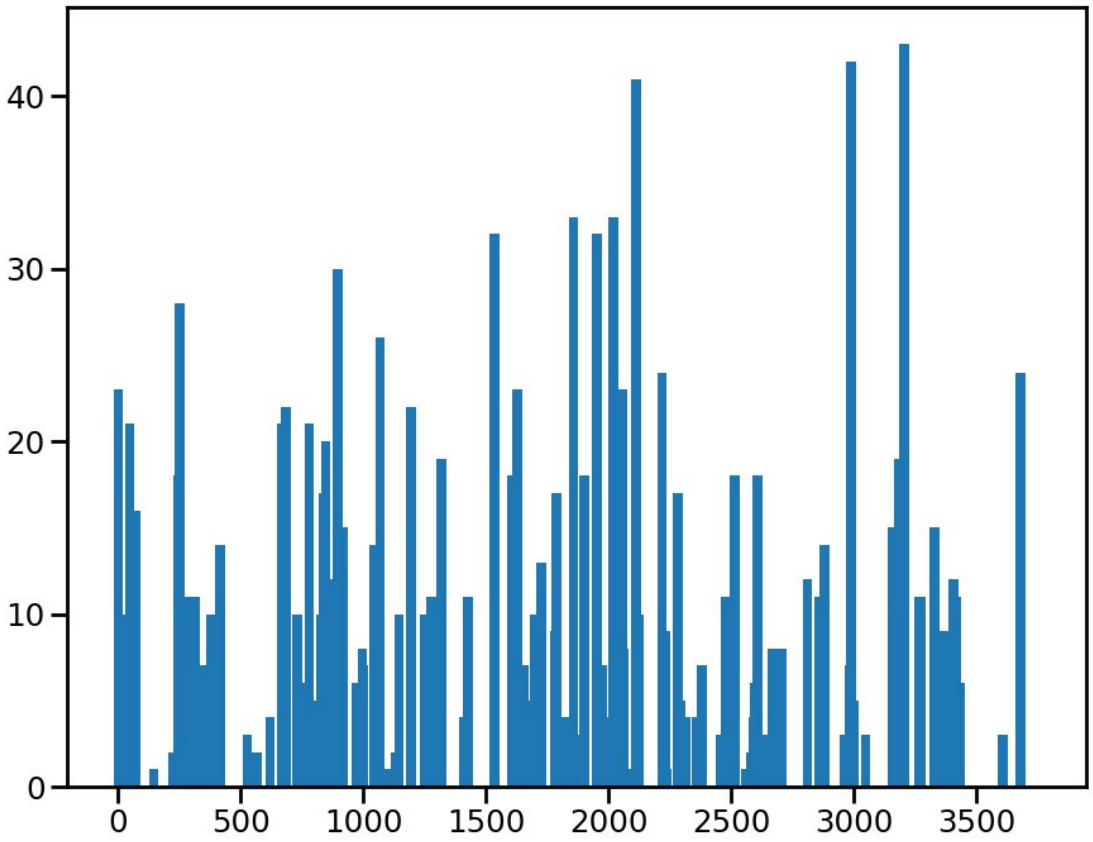


Рисунок 5 Діаграма кількості транзакцій в залежності від часу підтвердження блоку(4.2)

більш детальному розгляді виявляється одночасно обґрунтованим і суперечливим. Розглянемо, наприклад, припущення, що більшість людей очікують більшого часу підтвердження транзакції, хоча в середньому він становить 10 хвилин. Щоб перевірити цю гіпотезу, ми використали невелику вибірку з 140 блоків біткоїнів з номерами від 759149 до 759289.

Наступна гістограма ілюструє, що час підтвердження транзакцій дійсно підпорядковується пуассонівському розподілу. Більшість транзакцій, 78%, підтверджуються між 5 і 20 хвилинами. Середній час пошуку блоку становить близько 9,9 хвилин.

На графіку нижче показано хвилини між блоками, представлені помаранчевою

лінією. Обсяг транзакцій за блок зображено синіми стовпчиками. Наші результати показують, що середня кількість транзакцій за блок становить приблизно 1805.

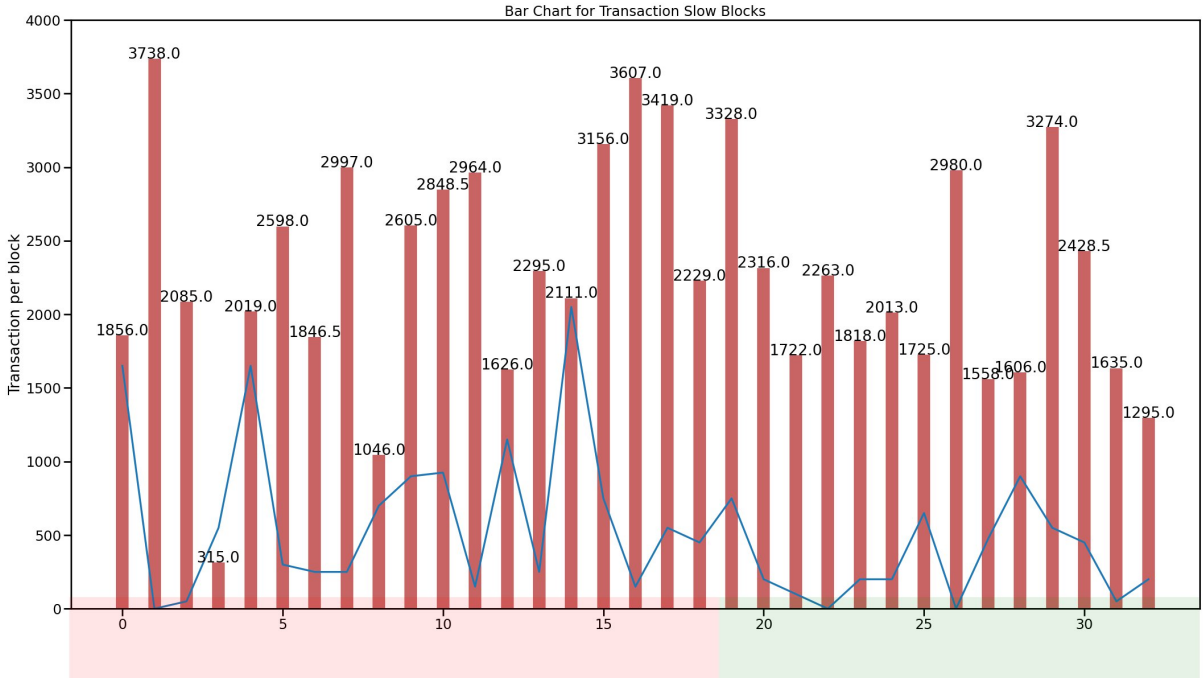


Рисунок 6 Середня кількість транзакцій на блок(4.3)

Синя лінія відображає час, витрачений на очікування підтвердження блоку. Вона показує зв'язок між часом очікування блоку і обсягом транзакцій в наступному блоці. По суті, якщо підтвердження відбувається швидко, як показано на 10.47, наступний блок буде порожнім. Навіть якщо 2000 транзакцій було підтверджено протягом 5 хвилин, блоки, що їх містили, були заповнені лише наполовину, що свідчить про те, що менше користувачів отримали швидке підтвердження порівняно з користувачами о 13.21, коли майже 11 000 транзакцій чекали на підтвердження 40 хвилин.

Таким чином, не обов'язково, що всі блоки, які отримали швидке підтвердження, є невеликими. Наприклад, о 10:55 один блок потребував 5 хвилин майнінгу, але підтвердив лише 200 транзакцій. І навпаки, о 15:28 блок був виявлений за 0 хвилин і містив 2982 транзакції. Отже, розмір блоку залежить від кількості транзакцій, що очікують підтвердження в пулі пам'яті (перевантаження),

а не від швидкості надходження блоків.

Хоча цей аналіз даних не є остаточним доказом існування парадоксу Пуассона в біткоїні, доказ насправді криється в початковій гістограмі, а саме в її довгому хвості. Більшість респондентів у нашій вибірці чекають на підтвердження більше 10 хвилин, хоча середній час очікування становить 9,9 хвилин. Це пов'язано з довгим правим хвостом розподілу Пуассона. Іншими словами, існує більше можливостей виявити блок між 10-40 хвилинами, оскільки цей часовий інтервал в чотири рази більший, ніж інтервал 0-10 хвилин. Для кращого розуміння зверніться до діаграми нижче.

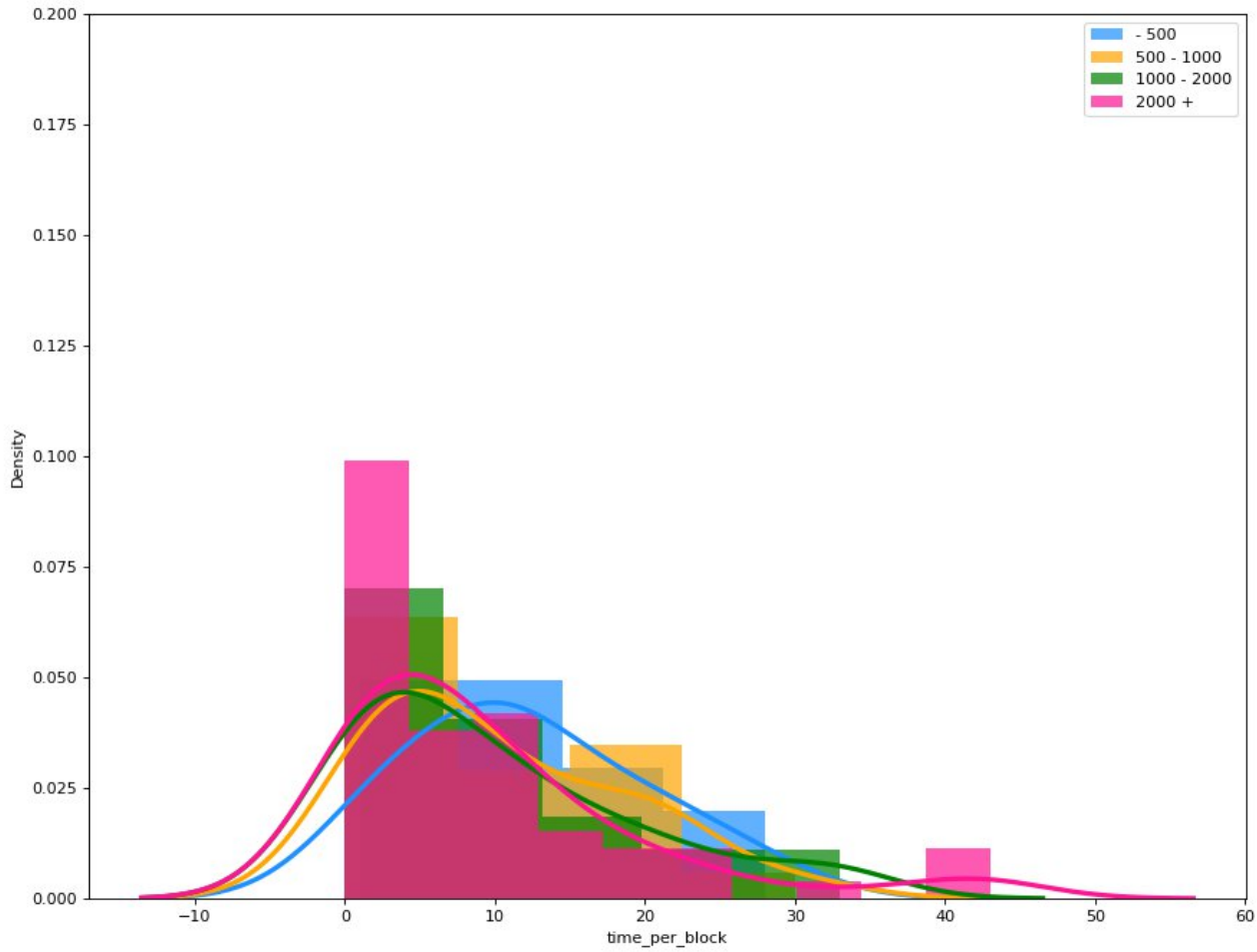


Рисунок 7 Графік щільності часу в залежності від кількості транзакцій(4.4)

Таблиця 4.1 - Результати виборки

Час підтвердження блоку	Відсоток вибірки
0 – 10 хвилин	40 %
10 – 40 хвилин	60 %

Таким чином, 2/5 нашої вибірки отримали підтвердження транзакції менш ніж за 10 хвилин, тоді як решта 3/5 були свідками того, що час підтвердження перевищував 10 хвилин. Це те, що ми називаємо парадоксом Пуассона.

Існує кілька інших факторів, які можуть призвести до того, що транзакції Bitcoin будуть довшими або коротшими за середній 10-хвилинний часовий проміжок. Однак вони не пов'язані безпосередньо з парадоксом Пуассона.

По-перше, майнери витягують і хешують транзакції з пулу пам'яті в заголовок блоку перед тим, як шукати наступне значення nonce. Це може призвести до відставання. Наприклад, нові транзакції, які майнери ще не забрали, залишаються в пулі пам'яті, поки не буде видобуто попередній блок. Це включає в себе час, необхідний для виявлення нових блоків. Транзакції також можуть застрягти, якщо пов'язані з ними комісії занадто низькі[15].

Тим не менш, якщо транзакція застрягла, її комісію можна збільшити за допомогою таких методів, як Child-Pays-for-Parent (CPFP) або Replace-by-Fee (RBF). Збільшення комісії за транзакцію підвищує її пріоритет, що підвищує ймовірність того, що майнери включать її в наступний блок. Крім того, деякі майнінг-пули мають можливість додавати транзакції безпосередньо до своїх блоків, що може прискорити час проведення транзакцій.

5. Запропоноване рішення

Запропоноване нами рішення - це платіжна система на основі блокчейну, розроблена для вирішення існуючих проблем у сучасних технологіях блокчейну. Архітектура системи складається з декількох взаємопов'язаних компонентів: вузлів блокчейну, реєстру блокчейну, пулу пам'яті, процесів верифікації транзакцій, веб-гаманця та консольного інтерфейсу.

Ці компоненти працюють разом, обробляючи дані відповідно до стандартних протоколів блокчейну. У веб-гаманці дані шифруються і обробляються авторизацією, в той час як в ядрі блокчейну відбувається обмін пакетами даних, включаючи повідомлення про блоки і транзакції, між вузлами. Ця система може бути реалізована на різних апаратних платформах або хмарних середовищах, пропонуючи надійну безпеку і масштабованість.

5.1 Архітектура системи

Запропонована блокчейн-система складається з унікальної та комплексної архітектури, спрямованої на подолання деяких загальних проблем в існуючих блокчейн-платформах. Її архітектурними компонентами є

- **Ноди блокчейну**: Кожен вузол у цій мережі може виступати як клієнтом (робити запити), так і сервером (отримувати запити), що сприяє децентралізації системи. Кожен вузол також містить копію реєстру блокчейну, що забезпечує високу відмовостійкість системи і стійкість до поділу мережі.
- **Блокчейн-леджер**: Це децентралізований журнал транзакцій, доступний лише для додатків, який розподілений між усіма вузлами мережі. Він забезпечує прозорість і незмінність записаних транзакцій. У нашій системі реєстр слідує за ланцюжком блоків, де кожен блок пов'язаний зі своїм попередником за допомогою хешу[6].

- **Mempool**: Mempool зберігає непідтверджені транзакції, які очікують на включення в наступний блок. Він слугує буфером для транзакцій до того, як вони будуть офіційно записані в реєстрі блокчейну. Кожен вузол в мережі підтримує власну версію пулу пам'яті.
- **Процес перевірки транзакцій**: Транзакції спочатку розподіляються по мережі в пул пам'яті кожного вузла. Транзакція включається в блокчейн тільки після того, як вона підтверджена як дійсна. Система використовує гібридний механізм консенсусу (PoET і PoW) для перевірки транзакцій, який балансує між обчислювальною ефективністю і безпекою.
- **Веб-гаманець**: Веб-гаманець - це інтерфейс, який дозволяє користувачам взаємодіяти з мережею блокчейн. Він включає в себе функціонал для створення транзакцій, перегляду історії транзакцій та перевірки балансу. Транзакції, ініційовані через веб-гаманець, транслюються в мережу і додаються до пулу пам'яті.
- **Консольний інтерфейс**: Консольний інтерфейс - це більш просунутий інтерфейс, який надає додаткові функції для системних адміністраторів або досвідчених користувачів. Це можуть бути функції керування вузлами, детальна мережева статистика тощо.
- **Сервер пулу і сервер часу**: Ці сервери полегшують спільну розробку стратегій майнінгу і синхронізують час на всіх вузлах, сприяючи підвищенню надійності і точності системи.
- **Гібридна мережа**: Архітектура цієї системи є гібридом як клієнт-серверної, так і однорангової мережі, що дозволяє використовувати сильні сторони обох типів мереж. Така структура є особливо стійкою, оскільки поєднує в собі надійність однорангової мережі зі стабільністю та ефективністю клієнт-серверної мережі.

Архітектура цієї системи розроблена з акцентом на децентралізацію, безпеку та масштабованість. Завдяки поєднанню гібридного консенсусу, гібридного

налаштування мережі та ефективного процесу перевірки транзакцій, ця система вирішує проблеми високої затримки транзакцій, централізації та парадоксу Пуассона, які часто зустрічаються в традиційних блокчейн-платформах.

5.2 Кореляція між проблемами та рішеннями

Цей розділ співвідносить виявлені проблеми з відповідними рішеннями, які пропонує наша блокчейн-система.

- **Затримка транзакцій**: Традиційні технології блокчейн часто страждають від високої затримки транзакцій, що робить їх непридатними для транзакцій в режимі реального часу. Запропонована нами блокчейн-система вирішує цю проблему шляхом впровадження гібридного механізму консенсусу, що складається з Proof of Elapsed Time (PoET) і Proof of Work (PoW). PoET в основному займається перевіркою транзакцій, використовуючи притаманну йому ефективність, в той час як PoW займається вирішенням потенційних розгалужень, які можуть виникнути, таким чином, ефективно підтримуючи баланс між швидкістю і безпекою.
- **Масштабованість**: Наша система вирішує проблему масштабованості, загальну для сучасних блокчейн-технологій, шляхом розгортання гібридної однорангової мережі. Такий дизайн мережі дозволяє нашій системі підтримувати високий рівень продуктивності при масштабуванні, оскільки кожен додатковий вузол збільшує загальну пропускну здатність мережі для обробки транзакцій.
- **Ризики централізації**: Ризики централізації - ще одна проблема, яку вирішує наша блокчейн-система. Гібридний механізм консенсусу PoET/PoW відіграє тут вирішальну роль. На відміну від чистих PoW-систем, де потужність майнінгу може бути потенційно централізована в руках невеликої кількості потужних суб'єктів, компонент PoET в нашій системі забезпечує більш справедливий розподіл потужності майнінгу. Ця рівновага зберігає ідею

децентралізації блокчейну, яка є основним принципом запропонованої нами системи.

- **Споживання енергії**: Ми розробили нашу систему, щоб вона була екологічно чистою. Завдяки використанню гібридного механізму консенсусу PoET і PoW, наша система значно знижує споживання енергії в порівнянні зі звичайними блокчейнами PoW. Враховуючи, що PoET менш енергоємний, ніж PoW, він обробляє більшість підтверджень транзакцій, що призводить до зниження загального енергоспоживання.
- **Високі комісійні витрати**: Проблема високих комісій за транзакції, особливо під час високих перевантажень мережі, є ще однією проблемою, яку вирішує наша система. Завдяки ефективному дизайну мережі та механізму обробки транзакцій наша система забезпечує високу продуктивність мережі навіть під час пікових обсягів транзакцій. Як результат, вона може підтримувати низьку комісію за транзакції, що робить її економічно доцільною і зручною для повсякденних транзакцій.

Застосовуючи такий комплексний підхід до вирішення проблем, ми впевнені, що наша блокчейн-система є найкращою альтернативою в індустрії блокчейн-технологій. Вона об'єднує сильні сторони механізмів консенсусу PoET і PoW, що дозволяє створити швидку, масштабовану та енергоефективну систему. Крім того, гібридна структура мережі забезпечує високий рівень децентралізації, підтримуючи таким чином демократичний дух блокчейну.

6. Реалізація програмного забезпечення

У цьому розділі було обговорено дизайн, архітектуру та реалізацію нашого програмного забезпечення на основі блокчейну.

6.1 Створення програмного продукту

Програмний продукт було створено з використанням мови програмування Go, завдяки її простоті, високій продуктивності та потужній підтримці паралельних процесів, що є критично важливим для роботи мережі блокчейн.

Ми також використали різні пакети Go, такі як bufio, fmt, os, strconv, strings, json, blockchain і network, щоб абстрагуватися і спростити деякі завдання, пов'язані з мережею, криптографією, взаємодією користувачів і управлінням даними.

6.2 Класи та об'єкти

Програмне забезпечення використовує декілька класів та об'єктів, визначених у пакетах блокчейну та мережі. Короткий опис кожного з них наведено нижче:

- ‘User’: Клас User представляє користувача в мережі блокчейн. Він надає методи для доступу до специфічної для користувача інформації, такої як адреса та гаманець користувача. Об'єкти цього класу створюються методами userNew або userLoad, в залежності від наданих аргументів.
- ‘Transaction’: Клас Transaction представляє транзакцію в блокчейні. Він інкапсулює всі необхідні деталі транзакції, такі як відправник, одержувач і сума переказу.
- ‘Package’: Клас Package з мережевого пакету представляє пакет даних, який надсилається через мережу. Він включає опцію (операцію, яку потрібно виконати) і дані (пов'язані з нею дані).

6.3 Реалізація інтерфейсу користувача

Інтерфейс користувача для програмного забезпечення - це інтерфейс командного рядка (CLI). Таке рішення було прийнято для того, щоб зробити додаток легким і незалежним від платформи.

CLI надає користувачеві команди для взаємодії з мережею блокчейн, такі як створення нового користувача, завантаження існуючого користувача, створення транзакцій і запит балансів. Кожна команда має структуру /command arg1 arg2 ..., де command - ім'я команди, а arg1, arg2, ... - аргументи команди.

Ось приклад коду функції розбору та обробки команд:

```
func handleClientInput() {
    // ...
    for {
        message := inputString("> ")
        splitted := strings.Split(message, " ")

        switch splitted[0] {
        case "/exit":
            os.Exit(0)
        case "/user":
            handleUserCommand(splitted)
        case "/chain":
            handleChainCommand(splitted)
        default:
            fmt.Println("Undefined command")
        }
    }
}
```

6.4 Реалізація програмного модуля

Основна логіка програми інкапсульована в головній функції handleClientInput. Ця функція обробляє команди користувача, викликає відповідні функції на основі команд і обробляє будь-які помилки, які можуть виникнути під час цього процесу.

Програма використовує функцію init для розбору аргументів командного рядка та

ініціалізації глобальних змінних Address і User. Потім функція main викликає функцію handleClientInput для обробки команд клієнта.

6.5 Інструкція для користувача програми

Щоб скористатися програмою, виконайте наступні кроки:

- Запустіть виконуваний файл з відповідними аргументами командного рядка: -loadaddr:<шлях до файлу адреси>, -newuser:<ім'я користувача> або -loaduser:<ім'я користувача>.
- Після запуску програми ви побачите запрошення >. Тут ви можете ввести наступні команди:
    - /адреса користувача: Вивести адресу поточного користувача.
    - /гаманець користувача: Надрукувати гаманець поточного користувача.
    - /user balance: Роздрукувати баланс поточного користувача.
    - /chain print: Вивести поточний стан блокчейну.
    - /chain tx <отримувач> <сума>: Створити транзакцію для відправки монет на вказану суму одержувачу.
    - /chain balance <user>: Вивести баланс користувача.
    - /exit: Вийти з програми.
- Після завершення роботи використовуйте команду /exit для безпечного виходу з програми.

6.6 Детальний опис класів та об'єктів

Давайте розширимо наше дослідження класів та об'єктів, реалізованих у цьому програмному забезпеченні.

6.6.1 Класи Block і BlockChain

Клас Block представляє окремий блок у блокчейні. Кожен блок включає в себе CurrHash, карту Mapping (яка відображає стан кожної адреси в блоці), Miner (хто видобув цей блок) і TimeStamp (коли цей блок був створений).

Клас BlockChain представляє весь блокчейн і включає в себе БД для підключення до бази даних та індекс для відстеження довжини блокчейну. Він надає різні методи для маніпуляцій та запитів до блокчейну, такі як LastHash, Balance, Size, AddBlock та HeadBlock. Ці методи дозволяють нам взаємодіяти з блокчейном та його окремими блоками.

Ось приклад реалізації класу BlockChain:

```
type BlockChain struct {
    DB *sql.DB
    index uint64
}

func (chain *BlockChain) AddBlock(block *Block) error {
    chain.index++
    _, err := chain.DB.Exec("INSERT INTO BlockChain (Hash, Block) VALUES (?, ?)",
    Base64Encode(block.CurrHash),
    SerializeBlock(block))
    return err
}
```

6.6.2 Клас Transaction

Клас Transaction, який представляє транзакцію в блокчейні, є критично важливим для функціонування нашого блокчейн-додатку. Він включає різні поля, такі як заголовок (який містить мета-інформацію про транзакцію) та входи і виходи (які містять фактичні дані транзакції).

Ось спрощений огляд класу Transaction:

```
type Transaction struct {
    Header  TransactionHeader
    Inputs  []TxInput
    Outputs []TxOutput
}
```

6.6.3 Реалізація мережевого пакету

Мережевий компонент нашого програмного забезпечення реалізований за допомогою мережевого пакету. Цей пакет реалізує мережевий зв'язок між вузлами в мережі блокчейн.

Мережевий пакет включає клас Package, який інкапсулює дані, що надсилаються мережею, та функцію Send, яка надсилає пакет на вказану адресу.

Ось спрощений огляд класу Package та функції Send:

```
type Package struct {
    Option  byte
    Data    string
}

func Send(address string, pkg *Package) *Package {
    // implementation omitted for brevity
}
```

Команди запитів та маніпуляцій з блокчейном

Наше програмне забезпечення надає різні команди для взаємодії з блокчейном, такі як /chain tx для створення транзакції та /chain balance для запиту балансу користувача. Ці команди взаємодіють з блокчейном шляхом виклику методів в об'єкті BlockChain.

Наприклад, ось як реалізована команда /chain balance:

```
func chainBalance(splited []string) {
    if len(splited) != 2 {
        fmt.Println("len(splited) != 2\n")
        return
    }
    printBalance(splited[1])
}

func printBalance(useraddr string) {
    for _, addr := range Address {
        res := nt.Send(addr, &nt.Package{
            Option: GET_BLNCE,
            Data:   useraddr,
```

```
        })
        if res == nil {
            continue
        }
        fmt.Printf("Balance (%s): %s coins\n", addr, res.Data)
    }
    fmt.Println()
}
```

У цій реалізації printBalance зв'язується з кожним вузлом мережі блокчейн (представленим зрізом Address), щоб запитати баланс за вказаною адресою. Це робиться шляхом надсилання пакету GET_BLNCE до кожного вузла і роздруківки відповідей.

7. Висновки

7.1 Результати та висновки

7.1.1 Короткий огляд досягнень

Наша розробка криптовалютної системи на основі блокчейну є значним досягненням у цій галузі. Ми успішно розробили систему, яка включає в себе основні функції криптовалютної мережі, такі як обробка транзакцій, майнінг блоків і ведення розподіленої книги. Наш підхід до впровадження двох типів гаманців - веб-гаманця для новачків та інтерфейсу командного рядка для досвідчених користувачів, ефективно задовольнив широку базу користувачів.

Гібридна архітектура системи, що є поєднанням однорангової та клієнт-серверної мережевих моделей, довела свою ефективність у підтримці надійного та ефективного зв'язку між різними вузлами мережі. Дотримання правил консенсусу Біткоїна, таких як правило найдовшого ланцюжка для вирішення конфліктів і підтвердження роботи для майнінгу блоків, а також безпечне зберігання ключів користувачів у зашифрованій базі даних, підкреслює надійність системи.

7.2 Практичне значення та подальша робота

7.2.1 Практичне значення

Практичне значення цього проекту є багатогранним і поширюється на різні сфери цифрової економіки. Зі все більшим впровадженням технології блокчейн у різних галузях ця криптовалютна система має значний практичний вплив.

По-перше, система забезпечує надійний, безпечний та енергоефективний засіб для проведення транзакцій, усуваючи ключові обмеження в існуючих системах блокчейн. Висока затримка транзакцій та споживання енергії - суттєві проблеми

традиційних блокчейн-систем, таких як Біткоїн, - ефективно вирішуються в нашому рішенні. Це робить систему більш придатною для транзакцій в режимі реального часу, відкриваючи двері для її більш широкого використання в повсякденній комерційній діяльності.

По-друге, подвійний підхід до інтерфейсу користувача, через веб-гаманці та інтерфейс командного рядка, забезпечує доступність для широкого кола користувачів, що є значною практичною перевагою.

Нарешті, архітектура цієї системи також є рішенням проблеми масштабованості, яка присутня в існуючих блокчейн-системах. Оскільки наша система може обробляти більшу кількість транзакцій без перевантаження, вона вирішує проблему перевантаження мережі і, як наслідок, зростання комісій за транзакції.

7.2.2 Майбутня робота

Хоча наша система успішно досягає початкових цілей, завжди є місце для вдосконалення і розвитку. Нижче наведені потенційні напрямки майбутньої роботи:

- **Вдосконалення криптографічних методів**: Ми прагнемо впровадити більш досконалі криптографічні методи для подальшого посилення безпеки системи.
- **Покращений користувацький інтерфейс**: Користувацький досвід інтерфейсу веб-гаманця може бути покращений за допомогою більш зручних функцій.
- **Масштабування мережі**: Зі збільшенням кількості користувачів мережа може бути масштабована для підтримки більшої кількості транзакцій та користувачів.
- **Ефективні алгоритми консенсусу**: Дослідження і впровадження більш ефективних алгоритмів консенсусу може знизити обчислювальну

потужність, необхідну для майнінгу, що призведе до більш енергоефективної системи.

- **Розробка API**: API для взаємодії сторонніх додатків з системою розширить її потенційне застосування і призведе до розвитку додаткових сервісів навколо криптовалютної системи.
- **Аналіз впливу на навколишнє середовище**: Оскільки занепокоєння щодо впливу технології блокчейн на навколишнє середовище зростає, майбутня робота може також зосередитися на вивченні методів мінімізації енергоспоживання та впливу нашої системи на навколишнє середовище.

Загалом, наша криптовалютна система на основі блокчейну має значне практичне застосування і є багатообіцяючою основою для подальших розробок і вдосконалень.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Накамото, С. (2008). Біткойн: пірингова система електронних грошей. [Електронний ресурс] Режим доступу: https://bitcoin.org/bitcoin.pdf (Дата звернення: 28.05.2023).
2. Вуд, Г. (2014). Ethereum: Безпечний децентралізований узагальнений реєстр транзакцій. [Електронний ресурс] Режим доступу: https://ethereum.github.io/yellowpaper/paper.pdf (дата звернення: 28.05.2023).
3. Антонопулос, А. М. (2014). Опановуємо біткоїн: розблокування цифрових криптовалют. O'Reilly Media, Inc.
4. Мова програмування Go. (2023). [Електронний ресурс] Режим доступу: https://golang.org/doc/ (Дата звернення: 28.05.2023).
5. Бутерін, В. (2013). Смарт-контракт нового покоління та платформа децентралізованих додатків. [Електронний ресурс] Режим доступу: https://ethereum.org/en/whitepaper/ (дата звернення: 28.05.2023).
6. IEEE. (2016). Стандарт верифікації та валідації систем та програмного забезпечення - IEEE Std 1012-2016. Стандарти IEEE.
7. GSTC. (2018). GSTC R 34.10-2018: Інформаційні технології - Криптографічний захист інформації - Процеси підписання та перевірки електронного цифрового підпису. Національний стандарт України.
8. GSTC. (2012). GSTC R 34.11-2012: Інформаційні технології. Криптографічний захист інформації. Національний стандарт України.
9. ISO/IEC. (2016). ISO/IEC 27001-2016: Інформаційні технології - Методи забезпечення безпеки - Системи управління інформаційною безпекою - Вимоги. Національний стандарт України.
10. Монтрезор, А., Джеласіті, М. (2013). PeerSim: Масштабований симулятор P2P. У матеріалах дев'ятої міжнародної конференції IEEE з пірингових обчислень (P2P'09). [Електронний ресурс] Режим доступу: https://www.gsd.inesc-id.pt/~ler/docencia/rcs1314/papers/P2P2013_041.pdf (дата звернення: 28.05.2023).

11. SAFE Network. (2020). Еволюція термінології з розвитком технологій: Децентралізована проти розподіленої. [Електронний ресурс] Режим доступу: https://medium.com/safenetwork/evolving-terminology-with-evolved-technology-decentralized-versus-distributed-7f8b4c9eacb (Дата звернення: 28.05.2023).
12. Мова програмування Go. (2023). Go Programming Language Playlist. [Електронний ресурс] Режим доступу: https://www.youtube.com/playlist?list=PL4_hYwCyhAvZmzpIjwewZOdBMFJooHIHx (Дата звернення: 28.05.2023).
13. Стандартний макет проекту Go. (2023). Стандартний макет проекту Go. [Електронний ресурс] Режим доступу: https://github.com/golang-standards/project-layout (Дата звернення: 28.05.2023).
14. Walter, K. (2023). Огляд алгоритмів консенсусу в блокчейні. [Електронний ресурс] Режим доступу: https://github.com/cedricwalter/blockchain-consensus (Дата звернення: 28.05.2023).

15. Антонопулос, А. М., Діллон, В. (2017). Опановуємо біткоїн: програмування відкритого блокчейну. [Електронний ресурс] Режим доступу: https://github.com/bitcoinbook/bitcoinbook (Дата звернення: 28.05.2023).

16. Superstas. Реалізація Gcoin Mempool [Електронний ресурс]. - Режим доступу : https://github.com/superstas/gcoin/blob/master/gcoin/mempool/mempool.go. - Назва з екрану. - (2023).

17. Kiayias, A., Russell, A., David, B., Oliynykov, R. Ouroboros: Надійний захищений протокол блокчейну з доказом частки [Електронний ресурс]. - Режим доступу: https://pdfs.semanticscholar.org/7dce/801b2b13001d0d3b0319c550ee1977e456df.pdf. - Назва з екрану. - (2017).

18. Ляо К., Кац Я., Зікас В. Протоколи BFT під вогнем [Електронний ресурс]. - Режим доступу: https://arxiv.org/pdf/1801.07447.pdf. - Назва з екрану. - (2018).

19. Беріні М. Розробка та впровадження додатку біткоїн-гаманця [Електронний ресурс]. - Режим доступу: https://openaccess.uoc.edu/bitstream/10609/45861/6/mberiniTFM1215memoria.pdf. - Назва з екрану. - (2015).

ДОДАТКИ

2. General description

2.1 Product perspective

The blockchain application we're developing is a standalone system designed to provide a secure, decentralized platform for the peer-to-peer transfer of tokens. It comprises three main components: the blockchain core, a web wallet, and a console interface.

The blockchain core is the foundation of the system, managing transactions, blocks, and consensus algorithms.

The web wallet provides a user-friendly interface, facilitating the secure storage and management of tokens.

The console interface is a command line interface (CLI) that simplifies user interaction with the blockchain, enabling token transfer and balance checking.

The system employs a hybrid consensus mechanism combining Proof of Elapsed Time (PoET) and Proof of Work (PoW), ensuring secure, efficient transaction validation. This system is intended for a wide range of users and has no direct analogues, being a unique blend of blockchain core, web wallet, and console interface functionalities.

2.2 Product features

Key features of the system include:

- **Blockchain core**: This component manages the fundamental functionalities of the blockchain, ensuring the integrity and security of the decentralized ledger.
- **Web wallet**: A secure environment for token storage and management. Users can check balance, transfer tokens, and upload private keys for enhanced security.
- **Console interface**: This CLI allows users to interact easily with the blockchain system, facilitating operations like transaction creation and balance checking.
- **Hybrid PoET and PoW consensus**: This unique approach ensures fair and efficient transaction validation, enhancing system security, performance, and scalability.
- **Security features**: Compliant with GTSU R standards, the system prioritizes security to protect user tokens and transactions.
- **Scalability and performance**: The system, developed with Go, is capable of handling a significant number of transactions and users while maintaining fast processing speed.
- **Customizability**: The system enables users to create their own tokens, broadening its potential use cases.

The system, therefore, provides a comprehensive, secure, and user-friendly platform for peer-to-peer token storage and transfer.

2.3 User classes and characteristics

The Go app is designed for two main user categories:

1. End users: Ranging from beginners to advanced users, these individuals interact with the blockchain primarily via the web wallet and console interface.

2. Developers and administrators: Interacting on a technical level with the blockchain core and source code, these users have in-depth knowledge of blockchain technology.
3. Miners: These users provide computational power to verify and add transactions to the blockchain, contributing to the security and reliability of the network.
4. Novice users: Majority of end users, they interact primarily with the web wallet to manage their cryptocurrency assets.

2.4 Operating environment

The blockchain core, web wallet, console interface, and mining nodes operate on modest hardware requirements. The network is designed to handle approximately two transactions every two minutes to balance user needs and avoid network overload.

2.5 Design and implementation constraints

Constraints include programming language limitations, compliance with legal and regulatory requirements, platform restrictions, encryption standards, and implementation of industry best practices.

2.6 User documentation

User documentation is divided into end user and developer/administrator documentation. For end users, a comprehensive User Manual, an online help system, a FAQ section, and video tutorials are provided. For developers and administrators, a detailed Developer Guide, API documentation, and Administrator's Guide are available. All documentation is updated regularly to match system updates.

2.7 Assumptions and dependencies

Assumptions:

1. User knowledge: Users have basic web application skills, and miners possess technical blockchain knowledge.
2. Internet access: Users have reliable, high-speed internet for real-time updates of transactions and blocks.
3. Regulatory environment: The application adheres to Ukrainian laws and regulations related to blockchain and cryptocurrencies.
4. Maintenance and support: Continuous maintenance and support for the application are expected.

Dependencies:

1. Go programming language: The application's functionality and development depend on Go's continued support.
2. Web technologies: HTML, SASS, and JavaScript updates can impact the web wallet.
3. PoET consensus mechanism: Changes to PoET may affect the operation of the blockchain.
4. TEE (Trusted Execution Environment): Application's performance is tied to TEE technology for secure transaction processing.
5. Network infrastructure: Reliable network infrastructure is crucial for connecting miners and nodes for transaction verification and block creation.

3. System architecture

This public blockchain application developed in Golang maintains an immutable transaction record on the network. This section highlights the main components like block structure, transaction structure, verification mechanisms, storage, and consensus mechanisms.

3.1.1 Block Structure

Each block includes fields like CurrHash (hash of current block), PrevHash (hash of previous block), Nonce (unique number for mining), Difficulty (mining complexity), Miner (public key of miner), Signature (digital signature for block integrity), TimeStamp (time of block addition), Transactions (array of transactions), and Mapping (tracks all transactions).

3.1.2 Transaction Structure

Transactions, which drive blockchain actions, include RandBytes (random bytes for entropy), PrevBlock (hash of previous block), Sender (public key of sender), Reciver (public key of recipient), Sum (amount of transferred cryptocurrency), ToStorage (amount transferred to storage), CurrHash (hash of current transaction), and Sign (digital signature for transaction integrity).

3.1.3 Transaction and Block Verification

Security and integrity are maintained through the IsValid() function (validates transactions) and the IsBlockValid() function (validates blocks).

3.1.4 Blockchain Storage and Distribution

The blockchain data is stored via a SQLite database, each block serving as a record, enabling efficient storage, retrieval, and database replication between nodes for decentralization.

3.1.5 Consensus Mechanism

A hybrid model combining proof-of-elapsed time (PoET) and proof-of-work (PoW) ensures fairness in a decentralized environment and maintains system security.

Overall, the blockchain application structure ensures transaction security and integrity, fitting various applications like cryptocurrencies and decentralized applications (dApps).

3.2 User Interface and Experience

The blockchain application has both a GUI and CLI. The GUI, based on web technologies, offers an intuitive, user-friendly interface with a web wallet for browsing the blockchain, initiating transactions, and viewing transaction history. Key features include a Home page, Login/Registration, Wallet page, Explorer page, and secure Logout. For advanced users, a CLI offers additional control.

Feedback is given through notifications, clear error messages when problems occur, and robust security measures protect users' private keys and safely terminate user sessions. The interface balances usability and functionality for all user levels.

3.3 Networking and Communication

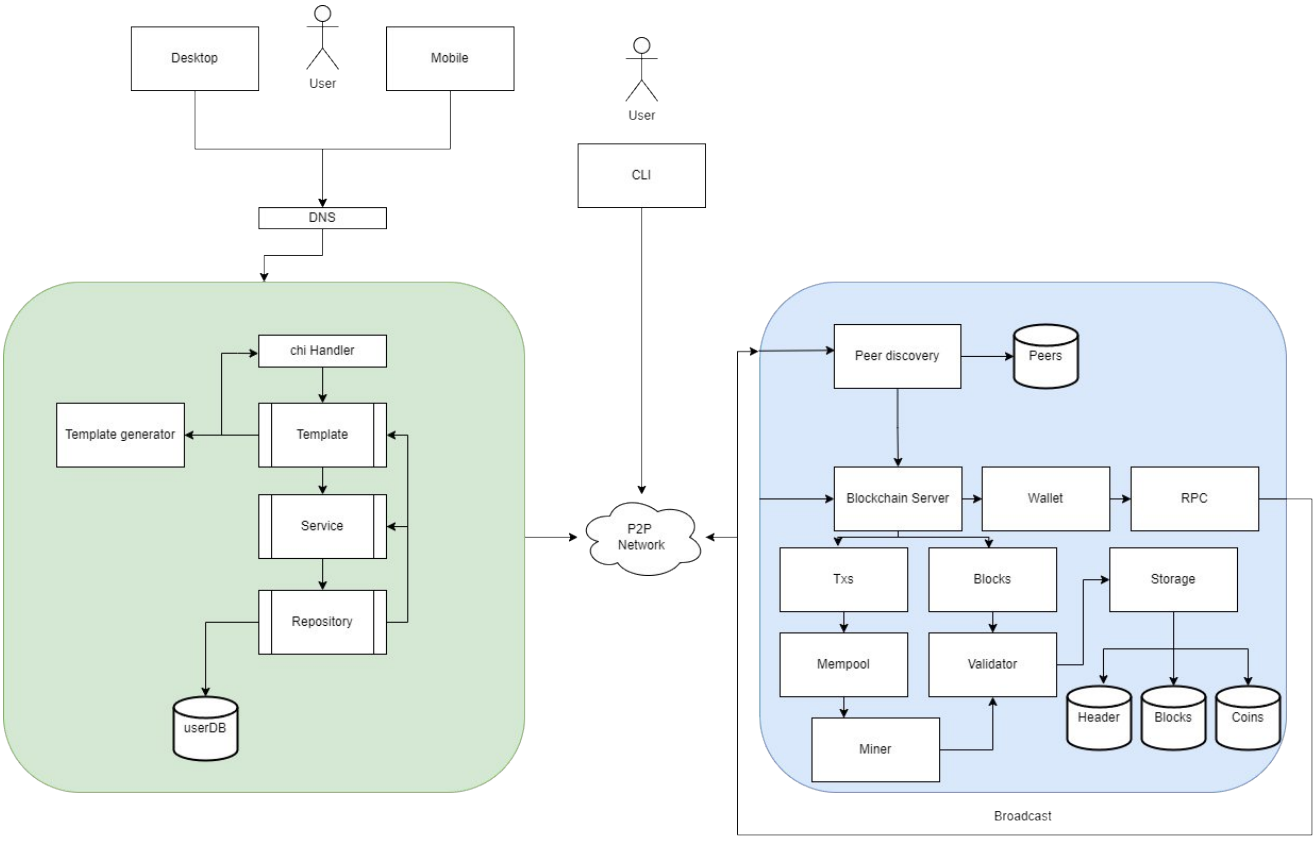
This blockchain application's hybrid network architecture combines client-server and peer-to-peer traits. The network communication flow involves clients and nodes

interacting for balance and block information, and to record transactions. Nodes request to add new blocks to the blockchain and request specific mining ranges or the current time state.

The handleServer function allows communication between nodes using TCP protocol, offering decentralization, high fault tolerance, and resistance to network partitioning. Transactions are distributed across the network using the makeTransaction function, and new blocks are distributed with the pushBlockToNet function. The "longest chain wins" rule maintains consensus in the network

Nodes require a file with IP addresses of trusted nodes, and the encryption standards used ensure secure communication between nodes. The presence of nodes accessing pool and time servers suggest support for a collaborative mining strategy and synchronized time on all nodes, boosting reliability and accuracy. This architecture provides a decentralized, resilient, and secure blockchain network enabling smooth, reliable transactions.

3.4 Architectural diagram



Architectural diagram

4 Detailed system design

4.1 User Interface Design

The application's user interface is designed to be intuitive for all users. Key elements include:

1. Landing page: Provides an overview and prompts for registration or login.
2. User dashboard: Displays balance, transaction history, and allows for transaction initiation and blockchain exploration.
3. Transaction process: Users input the recipient's address and transfer amount, with the system confirming these details.
4. Blockchain data: Presented transparently, showing transaction details for each block.
5. Registration and login: Simple and secure pages where users provide details to register and login.

4.2 Data Structures

The system's data structures, based on Go, mimic typical blockchain structures:

1. Blockchain: A chain of blocks, each containing a list of transactions and linked by storing the previous block's hash.
2. Block: Contains a list of transactions, a timestamp, previous block's hash, and its own hash.
3. Transaction: A cryptocurrency transfer, containing sender and recipient addresses, transfer amount, and timestamp.
4. User: A network member with unique credentials and a wallet.
5. MemPool: A collection of transmitted, yet unconfirmed, transactions.

The blockchain is stored on network nodes, not in the wallet, with the wallet interacting with it.

4.2 Data Structures

The system's data structures, based in Go, include:

1. Blockchain: A chain of blocks, each housing a list of transactions. Blocks are linked by their hashes.
2. Block: Contains transactions, a timestamp, the previous block's hash, and its own hash.
3. Transaction: Represents a cryptocurrency transfer, containing sender and recipient addresses, transfer amount, and timestamp.
4. User: A network member with unique credentials and a wallet.
5. MemPool: A collection of transactions that are not yet included in the block.

The blockchain interacts with the wallet, but is stored on network nodes.

4.3 Structure Design

Key structures include BlockChain, Block, Transaction, User, MemPool, and Package. They represent the fundamental components of the blockchain and provide methods for operations such as adding transactions and verifying blocks.

4.4 Implementation Details

The application, implemented in Go, uses standard Go library along with third-party libraries like Chi, Logrus, and SQLite. It features a multi-level architecture that promotes maintainability and scalability. Error handling is in place to manage issues during transactions or network communications. Security is ensured through strong encryption methods.

5. Conclusions and further work

5.1 Summary of achievements

The development and completion of this blockchain-based cryptocurrency system is a significant achievement. We have successfully implemented a system that includes the basic functions of a cryptocurrency network, including the ability to conduct transactions, mine new blocks and maintain a distributed ledger. The hybrid architecture of the system, which combines peer-to-peer and client-server network models, ensures reliable and efficient communication between different network nodes.

In addition, a notable achievement was the creation of two types of wallets - a web wallet for new users and a command line interface (CLI) for advanced users and miners. This dual approach makes the system accessible to a wide range of users, while offering advanced tools for more experienced users.

5.2 Lessons learnt

During the project implementation, we faced numerous challenges and gained invaluable experience. Implementing a peer-to-peer network for blockchain distribution and transaction verification was a significant experience. In addition, securing the keys in the database and ensuring the security of transactions was a complex task that required strict attention to detail.

The development process also highlighted the importance of thorough testing and debugging. Given the complexity of blockchain technology, thorough testing was vital to identify and correct potential issues. If the project were to be started again, paying more attention to the architecture and design of the system would help to anticipate and avoid potential problems.

5.3 Future work

While the current state of the project is working and meets its original goals, there is still room for expansion and improvement. Future work may include integrating more advanced cryptographic methods to increase the security of the system and implementing more user-friendly features in the web wallet interface to improve the user experience.

In addition, scaling the network to support more transactions and users is a potential area of future work. Research into more efficient consensus algorithms could also be conducted to reduce the computing power required for mining and make the system more energy efficient.

In addition, creating an API for third-party applications to interact with the system could be a useful feature to consider in the future. This would allow for the development of additional services and applications around the cryptocurrency system, thereby expanding its potential uses.