Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій

Кафедра програмних систем і технологій

УДК 004.772.9

На правах рукопису

ВИПУСКНА КВАЛІФІКАЦІЙНА БАКАЛАВРСЬКА РОБОТА

Тема: “Вузол електронної системи безготівкових розрахунків”

Спеціальність – 121 “Інженерія програмного забезпечення”

ПОЯСНЮВАЛЬНА ЗАПИСКА

Студент

ІПЗ-43 Давід ГОША

Науковий керівник

д.т.н. с.н.с Геннадій ПОРЄВ

Консультант
з питань нормоконтролю

фахівець Марина ШАТИРКО

Допускається до захисту
Завідувач кафедри

д.т.н., проф. Олексій БИЧКОВ

Київ – 2024

Рішенням
Екзаменаційної комісії

випускна кваліфікаційна робота студента

Давіда ГОШІ

захищена з оцінкою

_____ Голова Екзаменаційної комісії

Інна СТЕЦЕНКО

Київський національний університет імені Тараса Шевченка
Факультет інформаційних технологій
Кафедра програмних систем і технологій
Спеціальність 121 “Інженерія програмного забезпечення”

ЗАТВЕРДЖЕНО
Зав. кафедри програмних систем і технологій
_____(Олексій БИЧКОВ)
“____” _____ 20___р.

ЗАВДАННЯ

НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Гоші Давіду Олександровичу
(прізвище, ім’я, по батькові)

1. Тема випускної кваліфікаційної бакалаврської роботи:
«Вузол електронної системи безготівкових розрахунків»
керівник роботи д.т.н. с.н.с Порєв Геннадій Володимирович.
затверджені на засіданні кафедри програмних систем і технологій, протокол № 6 від 09 листопада2023 р.

2. Строк здачі студентом закінченої роботи: 14 червня 2024 р.

3. Вихідні дані до роботи

Основні методи та підходи до розробки вузла електронної системи безготівкових розрахунків, навчальна література, Web-джерела, наукові статті, технічні документації, специфікації протоколів взаємодії компонентів системи, стандарти криптографічного захисту та безпеки інформації в платіжних системах.

4. **Зміст розрахунково – пояснювальної записки (**перелік питань, які потрібно розробити)
Аналіз предметної області електронних систем безготівкових розрахунків, огляд існуючих архітектур та технологій побудови компонентів таких систем, детальний опис постановки завдання розробки вузла, напрямок дослідження в процесі реалізації програмного застосунку, , оцінка ефективності та продуктивності розробленого вузла в процесі досліджень, підготовка інфраструктури та середовища розробки для реалізації компонента системи, проведення тестування та порівняльний аналіз результатів з існуючими рішеннями, формування вимог до програмного застосунку, опис інструментів

та технологій для реалізації застосунку, розробка користувацького інтерфейсу та функціоналу електронного гаманця як складової вузла системи безготівкових розрахунків.

5. Перелік графічного матеріалу (з точним забезпеченням обов'язкових креслень)
Графічний матеріал вступу: 1 блок-схема.
Графічний матеріал першого розділу: 2 рисунки.
Графічний матеріал другого розділу: 3 рисунки, 2 графіки.
Графічний матеріал третього розділу: 1 рисунок, 1 діаграма.
Графічний матеріал четвертого розділу: 1 рисунок, 1 UML діаграма.
Графічний матеріал п'ятого розділу: 1 блок-схема, 3 рисунки.
Графічний матеріал шостого розділу: 2 рисунки, 1 фрагмент коду.

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Дата	
		Завдання видав	Завдання прийняв
Розділ 1 «Загальний огляд проблеми»	*Порєв Г. В.*	25.12.2023	14.01.2024
Розділ 2 «Теоретична інформація»	*Порєв Г. В.*	15.01.2024	04.02.2024
Розділ 3 «Ідентифікація проблеми»	*Порєв Г. В.*	05.02.2024	24.03.2024
Розділ 4 «Аналіз парадоксу Пуассона»	*Порєв Г. В.*	25.03.2024	28.04.2024
Розділ 5 «Запропоноване рішення»	*Порєв Г. В.*	05.04.2024	20.04.2024
Розділ 6 «Реалізація програмного забезпечення»	*Порєв Г. В.*	21.04.2024	30.04.2024

7. Дата видачі завдання 17.11.2023

Керівник (підпис) Порєв Г. В. (розшифровка підпису)

Завдання прийняв до виконання (підпис) Гоша Д. О. (розшифровка підпису)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Вивчення теоретичних основ технології блокчейн та існуючих платіжних систем	17.01.2024 -21.01.2024	Виконано
2	Аналіз проблем та обмежень сучасних блокчейн-платформ	22.01.2024 - 24.01.2024	Виконано
3	Формулювання мети та завдань дослідження	25.01.2024 - 01.02.2024	Виконано
4	Розробка концепції гібридного протоколу консенсусу	01.02.2024 - 10.02.2024	Виконано
5	Моделювання та аналіз парадоксу Пуассона в контексті PoW	11.02.2024 - 20.02.2024	Виконано
6	Проектування архітектури платіжної системи на основі блокчейну	21.02.2024 - 25.02.2024	Виконано
7	Реалізація основних компонентів системи (блокчейн, протокол консенсусу, механізми безпеки)	26.02.2024 - 10.03.2024	Виконано
8	Розробка інтерфейсів взаємодії з платіжною системою (веб-гаманець, консольний інтерфейс)	21.03.2024 - 31.03.2024	Виконано
9	Тестування та налагодження платіжної системи	01.04.2024 - 20.04.2024	Виконано
10	Проведення експериментів та оцінка ефективності запропонованих рішень	21.04.2024 - 30.04.2024	Виконано
11	Аналіз отриманих результатів та формулювання висновків	01.05.2024 - 10.05.2024	Виконано
12	Оформлення пояснювальної записки до дипломної роботи	11.05.2024 - 20.05.2024	Виконано
13	Підготовка графічного матеріалу (рисунки, діаграми, схеми)	21.05.2024 - 25.05.2024	Виконано
14	Створення презентації для захисту дипломної роботи	25.05.2024 - 30.05.2024	Виконано
15	Попередній захист дипломної роботи	30.05.2024 - 30.05.2024	Виконано
16	Виправлення зауважень та підготовка до остаточного захисту	31.05.2024 - 14.06.2024	Виконано
17	Остаточний захист дипломної роботи	21.06.2024 - 21.06.2024	Виконано

Студент – бакалавр . (підпис) . Давід ГОША (розшифровка підпису)

Керівник роботи . (підпис) . Геннадій ПОРЄВ (розшифровка підпису)

АНОТАЦІЯ

Випускна кваліфікаційна бакалаврська робота: викладена на 84 сторінках друкованого тексту, який складається із вступу, семи розділів, висновків та списку використаних джерел. Робота містить 7 рисунків, 2 таблиці, 19 інформаційних джерел.

Тема: Вузол електронної системи безготівкових розрахунків.

Об'єкт дослідження: процес функціонування та розробки платіжних систем на основі технології блокчейн, з фокусом на феномені парадоксу Пуассона в контексті алгоритму доказу виконаної роботи (Proof-of-Work, PoW).

Мета роботи: проаналізувати проблематику функціонування систем на базі блокчейну, дослідити фундаментальні концепти технології, ідентифікувати ключові виклики платіжних систем на основі блокчейну та запропонувати рішення, яке включає проектування архітектури системи та встановлення кореляційних зв'язків між проблемами та шляхами їх вирішення.

Предмет дослідження: фундаментальні концепти технології блокчейн, такі як Proof-of-Work, Proof-Of-Elapsed-Time, процес майнінгу, принципи балансування, структура транзакцій та модель UTXO; парадокс Пуассона в контексті алгоритму PoW; проблеми, пов'язані з атаками Sybil та середовищем довіреного виконання (TEE).

Результати дослідження: проведено компаративний аналіз існуючих аналогів, охоплюючи різні технології блокчейну та сфери їх застосування. Досліджено фундаментальні концепти технології блокчейн. Розглянуто інноваційний підхід до розробки гібридного протоколу консенсусу. Значну частину дослідження присвячено аналізу парадоксу Пуассона - сформульовано гіпотезу та проведено вивчення результатів моделювання. На основі отриманих даних запропоновано рішення, яке включає проектування архітектури системи та встановлення кореляційних зв'язків між проблемами

та шляхами їх вирішення. Описано процес практичної реалізації програмного забезпечення: від створення програмного продукту, проектування класів та об'єктів до розробки інтерфейсу користувача, програмного модуля та інструкції для користувача.

Висновок

Дослідження являє собою внесок у розуміння та вдосконалення платіжних систем на основі блокчейну, відкриваючи перспективи для подальших наукових розвідок у цій галузі. Отримані результати мають практичне значення для розвитку та оптимізації децентралізованих платіжних систем. Майбутня робота може зосередитися на вдосконаленні криптографічних методів, покращенні користувацького інтерфейсу, масштабуванні мережі, розробці більш ефективних алгоритмів консенсусу та аналізі впливу на навколишнє середовище.

БЛОКЧЕЙН, ПРОТОКОЛИ КОНСЕНСУСУ, PROOF-OF-WORK, PROOF-OF-ELAPSED-TIME, ПАРАДОКС ПУАССОНА, ПЛАТІЖНІ СИСТЕМИ, ГІБРИДНИЙ ПРОТОКОЛ КОНСЕНСУСУ, АРХІТЕКТУРА СИСТЕМИ, АТАКИ SYBIL, СЕРЕДОВИЩЕ ДОВІРЕНОГО ВИКОНАННЯ.

ANNOTATION

Bachelor's Thesis: presented on 84 pages of printed text, consisting of an introduction, seven chapters, conclusions, and a list of references. The work contains 7 figures, 2 tables, and 19 information sources.

Topic: Node of the electronic payment processing system

Object of Research: the process of functioning and development of payment systems based on blockchain technology, with a focus on the phenomenon of the Poisson paradox in the context of the Proof-of-Work (PoW) algorithm.

Purpose of the Work: to analyze the problems of functioning of blockchain-based systems, explore the fundamental concepts of the technology, identify key challenges of blockchain-based payment systems, and propose a solution that includes designing the system architecture and establishing correlations between problems and ways to solve them.

Subject of Research: fundamental concepts of blockchain technology, such as Proof-of-Work, Proof-Of-Elapsed-Time, the mining process, balancing principles, transaction structure, and the UTXO model; the Poisson paradox in the context of the PoW algorithm; problems associated with Sybil attacks and the Trusted Execution Environment (TEE).

Research Results: a comparative analysis of existing analogues was conducted, covering various blockchain technologies and their applications. The fundamental concepts of blockchain technology were investigated. An innovative approach to the development of a hybrid consensus protocol was considered. A significant part of the research is devoted to the analysis of the Poisson paradox - a hypothesis was formulated, and the results of modeling were studied. Based on the data obtained, a solution was proposed that includes designing the system architecture and establishing correlations between problems and ways to solve them. The process of practical implementation of the software is described: from the creation of a software product, designing classes and objects to the development of a user interface, a software module, and user instructions.

Conclusion

The research contributes to the understanding and improvement of blockchain-based payment systems, opening up prospects for further scientific research in this field. The results obtained are of practical importance for the development and optimization of decentralized payment systems. Future work may focus on improving cryptographic methods, enhancing the user interface, scaling the network, developing more efficient consensus algorithms, and analyzing the environmental impact.

BLOCKCHAIN, CONSENSUS PROTOCOLS, PROOF-OF-WORK, PROOF-OF-ELAPSED-TIME, POISSON PARADOX, PAYMENT SYSTEMS, HYBRID CONSENSUS PROTOCOL, SYSTEM ARCHITECTURE, SYBIL ATTACKS, TRUSTED EXECUTION ENVIRONMENT.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

Блокчейн: Цифровий реєстр транзакцій, який дублюється і розподіляється по мережі комп'ютерних систем на блокчейні.

Ядро: Центральний компонент обчислювальної системи, який керує роботою комп'ютера та апаратного забезпечення.

Веб-гаманець: Тип гаманця, який дозволяє користувачам керувати своїми криптовалютами через веб-інтерфейс, що робить його доступним з будь-якого комп'ютерного пристрою з підключенням до Інтернету.

Приватний ключ: Складна форма криптографії, яка дозволяє користувачеві отримати доступ до своєї криптовалюти.

Механізм консенсусу: Відмовостійкий механізм, який використовується в комп'ютерних і блокчейн-системах для досягнення необхідної згоди щодо

єдиного значення даних або єдиного стану мережі між розподіленими процесами або мультиагентними системами.

PoET: Скорочення від "Proof of Elapsed Time" (Доказ витраченого часу). Це алгоритм консенсусу, який використовує систему чесної лотереї, де кожен вузол в мережі має рівні шанси на перемогу.

PoW: Скорочення від "Proof of Work" (Доказ роботи). Це механізм консенсусу в мережі блокчейн, який використовується для підтвердження транзакцій і створення нових блоків в ланцюжку.

Хешування: Хеш-функція - це процес, який приймає вхідні дані і повертає рядок байтів фіксованого розміру, як правило, "дайджест". Хешування є наріжним каменем технології блокчейн. У контексті криптовалют криптографічна хеш-функція є особливим класом хеш-функцій, який особливо добре підходить для безпечної обробки великих обсягів даних.

Nonce: Nonce ("число, яке використовується лише один раз") - це число, додане до хешованого або зашифрованого блоку в блокчейні, яке при повторному хешуванні відповідає обмеженням складності. Він використовується в системах доказу роботи блокчейна, щоб зробити процес видобутку нового блоку (або запису в реєстрі) для блокчейна обчислювально дорогим.

Майнер: У контексті криптовалют майнер - це фізична або юридична особа, яка підтверджує і перевіряє нові транзакції і додає їх в блокчейн. Цей процес передбачає вирішення складних математичних задач і вимагає значних обчислювальних потужностей.

Криптовалюта: Це поширене скорочення для "криптовалюти", типу цифрової або віртуальної валюти, яка використовує криптографію для безпеки. Прикладами криптовалют є Bitcoin, Ethereum і Ripple.

Bitcoin: Bitcoin - перша децентралізована криптовалюта і залишається найвідомішою і найціннішою криптовалютою. Вона була створена в 2008 році невідомою особою під псевдонімом Сатоші Накамото. Біткоїни створюються як винагорода за процес, відомий як майнінг.

Протокол поширення чуток: Також відомий як "епідемічний протокол", це процедура або процес комп'ютерного спілкування peer-to-peer, заснований на тому, як поширюються епідемії. Деякі розподілені системи використовують протокол поширення чуток для зв'язку через його високу надійність.

Одноранговa мережа (P2P): Мережа P2P - це мережа, в якій кожен комп'ютер (або "вузол") виступає в якості сервера для інших, дозволяючи їм обмінюватися файлами і периферійними пристроями без необхідності центрального сервера. Мережі P2P можуть бути встановлені вдома, в бізнесі або в Інтернеті.

Вузол: Пристрій або точка даних у великій мережі. У контексті блокчейну вузлом є будь-який комп'ютер, підключений до мережі блокчейн.

Гаманець: У контексті криптовалюти гаманець - це цифровий інструмент, який дозволяє користувачам взаємодіяти з мережею блокчейн. Він дозволяє користувачам зберігати і управляти своїми криптовалютами.

CLI (Інтерфейс командного рядка): Текстовий інтерфейс користувача, що використовується для введення команд безпосередньо в комп'ютерну систему. У контексті даного проекту CLI призначений для просунутих користувачів і майнерів.

dAPP (децентралізований додаток): Це додаток, який працює в децентралізованій мережі, уникаючи єдиної точки відмови. У dApps серверний код працює в децентралізованій одноранговой мережі, на відміну від традиційних додатків, де серверний код працює на централізованих серверах.

TEE (Довірене середовище виконання): Безпечне середовище, яке забезпечує цілісність коду і конфіденційність даних всередині системи. У контексті блокчейну TEE використовується для захисту приватних ключів і забезпечення безпечного виконання консенсусних алгоритмів, таких як PoET.

Атака Сібіл: Тип атаки в одноранговой мережі, де атакуючий намагається отримати контроль над мережею, створюючи велику кількість псевдонімних ідентифікаторів вузлів.

UTXO (Невитрачений вихід транзакції): Модель, яка використовується в деяких блокчейнах, таких як Bitcoin, для відстеження балансу. Кожна транзакція споживає один або кілька UTXO і створює один або кілька нових UTXO.

Форк (розгалуження): Розгалуження в блокчейні відбувається, коли мережа тимчасово розділяється на два різних ланцюжки через розбіжності в консенсусі. Форки можуть бути навмисними (як оновлення протоколу) або ненавмисними (як результат атаки).

Парадокс Пуассона: Це явище в мережах блокчейн, де більшість користувачів стикаються з часом підтвердження транзакції, що перевищує середній час, незважаючи на експоненціальний розподіл часу між блоками. Це пов'язано з довгим хвостом розподілу Пуассона.

ВСТУП

Актуальність теми

Розвиток технологій блокчейну в останні роки продемонстрував великий потенціал для трансформації фінансових систем та підвищення ефективності різних секторів економіки. Однак, існуючі платформи блокчейну стикаються з низкою серйозних проблем, таких як низька масштабованість, високі витрати на транзакції, значне енергоспоживання та загрози безпеці. Зокрема, в контексті платіжних систем на основі блокчейну, ці проблеми стають ще більш актуальними, оскільки вони можуть суттєво впливати на швидкість та зручність здійснення транзакцій. Крім того, високі витрати на транзакції та енергоспоживання можуть зробити такі системи економічно невигідними для широкого застосування. Особливо важливим є вирішення цих питань для платіжних систем, що базуються на блокчейні, оскільки це може суттєво підвищити їх ефективність та зручність використання.

Водночас, технологія блокчейну має величезний потенціал для забезпечення більшої прозорості, безпеки та надійності фінансових операцій. Вона дозволяє створювати децентралізовані системи, які не залежать від централізованих органів управління, що може значно знизити ризики шахрайства та зловживань. Однак, для того, щоб повністю реалізувати цей потенціал, необхідно вирішити існуючі технічні проблеми та забезпечити ефективне функціонування таких систем у реальних умовах.

Особливої уваги заслуговує феномен парадоксу Пуассона в контексті алгоритму Proof-of-Work (PoW), який є основою багатьох сучасних блокчейн-систем. Парадокс Пуассона, що виникає через випадковий характер генерації блоків, може призводити до значних затримок у підтвердженні транзакцій, що негативно впливає на швидкість та ефективність платіжних систем. Дослідження цього феномену та розробка нових підходів до вирішення проблем, пов'язаних з парадоксом Пуассона, є надзвичайно важливими для покращення функціонування блокчейн-систем.

Крім того, сучасні блокчейн-системи стикаються з проблемами, пов'язаними з атаками Sybil, коли один зловмисник може створити багато підроблених вузлів у мережі для отримання контролю над системою. Це може призвести до серйозних порушень у роботі мережі та зниження її безпеки. Також важливим аспектом є забезпечення безпеки в середовищі довіреного виконання (Trusted Execution Environment, TEE), яке використовується для забезпечення надійності та цілісності виконання критично важливих операцій у блокчейн-системах.

Таким чином, тема вдосконалення технологій блокчейну, зокрема дослідження парадоксу Пуассона в контексті алгоритму Proof-of-Work (PoW) та розробка гібридного протоколу консенсусу, що поєднує Proof-of-Work та Proof-of-Elapsed-Time (PoET), є надзвичайно актуальною. Це дозволить не лише вирішити існуючі проблеми, але й забезпечити нові можливості для розвитку платіжних систем на основі блокчейну.

Мета і задачі дослідження

Метою даної роботи є підвищення ефективності та надійності платіжних систем на основі блокчейну шляхом розробки та впровадження гібридного протоколу консенсусу, що поєднує переваги алгоритмів Proof-of-Work (PoW) та Proof-of-Elapsed-Time (PoET). Для досягнення цієї мети необхідно провести глибокий аналіз існуючих технологій блокчейну, вивчити теоретичні основи алгоритмів консенсусу, а також розробити та протестувати новий гібридний протокол.

В межах обраного напрямку досліджень було поставлено такі задачі:

1. Аналіз існуючих технологій блокчейну та їх застосування, зокрема, у платіжних системах.
2. Вивчення теоретичних основ алгоритмів Proof-of-Work та Proof-of-Elapsed-Time, їх переваг та недоліків.
3. Огляд процесів майнінгу та балансування в блокчейн-системах, їх впливу на швидкість та ефективність транзакцій.

4. Дослідження структури транзакцій та моделі UTXO (Unspent Transaction Output), що використовується у багатьох сучасних блокчейн-системах.

5. Аналіз проблем, пов'язаних з платіжними системами на основі блокчейну, включаючи парадокс Пуассона, атаки Sybil та забезпечення безпеки в середовищі довіреного виконання (TEE).

6. Формулювання гіпотез щодо можливих шляхів вирішення виявлених проблем, розробка моделей та проведення експериментів для їх підтвердження.

7. Розробка та моделювання гібридного протоколу консенсусу, що поєднує елементи PoW та PoET, з метою підвищення ефективності та надійності платіжних систем.

8. Реалізація програмного забезпечення, що реалізує розроблений протокол, та проведення експериментальних досліджень для оцінки його ефективності.

9. Порівняльний аналіз результатів експериментів, визначення переваг та недоліків запропонованого рішення у порівнянні з існуючими підходами.

10. Визначення оптимальних умов для використання розробленого протоколу у реальних платіжних системах, рекомендації щодо його впровадження та подальшого розвитку.

Об’єкт дослідження

Об’єктом дослідження є платіжні системи на основі блокчейну та їх функціональні характеристики, такі як масштабованість, швидкість транзакцій, енергоспоживання та безпека. Особлива увага приділяється системам, що використовують алгоритми консенсусу Proof-of-Work та Proof-of-Elapsed-Time.

Предмет дослідження

Предметом дослідження є алгоритми консенсусу в блокчейн-системах, зокрема Proof-of-Work, Proof-of-Elapsed-Time та їх гібридні моделі, а також

методи підвищення ефективності та надійності платіжних систем на основі блокчейну.

Методи дослідження

Для досягнення поставлених задач використовувалися різноманітні методи дослідження, які включали наступні підходи:

1. Математичне моделювання: Це дозволило створити теоретичні моделі функціонування блокчейн-систем з використанням різних алгоритмів консенсусу. Зокрема, моделювання дозволило детально вивчити процеси генерації блоків та підтвердження транзакцій, а також оцінити вплив різних факторів на ефективність роботи системи.

2. Порівняльний аналіз: Включав дослідження існуючих алгоритмів консенсусу, їх переваг та недоліків. Було проведено порівняння ефективності різних підходів до вирішення проблем, пов'язаних з масштабованістю, швидкістю транзакцій та енергоспоживанням. Аналізувався як теоретичний, так і практичний досвід застосування цих алгоритмів у різних блокчейн-платформах.

3. Експериментальні дослідження: Розробка програмного забезпечення, проведення симуляцій та тестування розроблених моделей у реальних умовах. Це дозволило оцінити практичну ефективність запропонованих рішень та визначити оптимальні параметри для їх реалізації.

4. Аналіз даних з блокчейн-мереж: Включав збір та обробку даних про транзакції, швидкість підтвердження блоків, енергоспоживання та інші параметри роботи блокчейн-систем. Це дозволило отримати реальну картину роботи систем та виявити проблемні місця, які потребують удосконалення.

5. Моделювання та симуляції: Використання комп'ютерних симуляцій для відтворення роботи блокчейн-систем з різними параметрами та алгоритмами консенсусу. Це дозволило провести серію експериментів у контрольованих умовах та оцінити ефективність запропонованих рішень без необхідності їх реалізації у реальних мережах.

6. Кількісні та якісні аналізи: Застосування методів кількісного та якісного аналізу для оцінки ефективності різних алгоритмів консенсусу. Зокрема, оцінювалися такі параметри, як швидкість підтвердження транзакцій, енергоспоживання, стійкість до атак Sybil та інші.

7. Огляд літератури та патентів: Аналіз наукових статей, технічних звітів, патентів та інших джерел інформації, що стосуються технологій блокчейну та алгоритмів консенсусу. Це дозволило виявити найсучасніші тенденції у цій галузі та врахувати їх при розробці нових рішень.

Новизна одержаних результатів

Результати даного дослідження мають значне практичне значення для розробників платіжних систем та інших децентралізованих додатків, що базуються на технології блокчейну. По-перше, розробка гібридного протоколу консенсусу, який поєднує переваги алгоритмів Proof-of-Work (PoW) та Proof-of-Elapsed-Time (PoET), дозволяє значно підвищити ефективність блокчейн-систем. Цей протокол забезпечує швидше підтвердження транзакцій та знижує енергоспоживання, що робить його економічно вигіднішим у порівнянні з традиційними алгоритмами консенсусу. Таким чином, впровадження цього протоколу може суттєво зменшити витрати на експлуатацію блокчейн-систем, зробивши їх більш доступними для широкого кола користувачів та розробників.

По-друге, вдосконалення методів захисту від атак Sybil та забезпечення безпеки в середовищі довіреного виконання (TEE) підвищує надійність та безпеку блокчейн-систем. Запропоновані підходи можуть бути використані для розробки нових, більш стійких до атак блокчейн-мереж, що є критично важливим для фінансових установ та інших організацій, які використовують блокчейн для зберігання та передачі чутливої інформації.

По-третє, практична реалізація та експериментальне тестування розроблених моделей дозволяє оцінити їх ефективність у реальних умовах та внести необхідні корективи для подальшого вдосконалення. Результати тестувань підтвердили високу ефективність та надійність розробленого

протоколу, що дозволяє рекомендувати його для впровадження у платіжні системи на основі блокчейну. Це відкриває нові можливості для розвитку фінансових технологій та створення більш безпечних, ефективних та зручних платіжних систем.

Практичне значення одержаних результатів

Дослідження має кілька новаторських аспектів, які суттєво впливають на розвиток технологій блокчейну та підвищення ефективності платіжних систем. По-перше, розробка гібридного протоколу консенсусу, який поєднує переваги алгоритмів Proof-of-Work (PoW) та Proof-of-Elapsed-Time (PoET), є новаторським підходом до вирішення проблем масштабованості та енергоспоживання блокчейн-систем. Цей протокол забезпечує швидше підтвердження транзакцій та знижує енергоспоживання, що робить його більш ефективним та економічно вигідним у порівнянні з традиційними алгоритмами консенсусу.

По-друге, детальне дослідження парадоксу Пуассона в контексті PoW показало його вплив на час підтвердження транзакцій у блокчейн-системах. Було встановлено, що більшість користувачів стикаються з більшими затримками, ніж середній час генерації блоків, що негативно впливає на швидкість та зручність використання платіжних систем. Це відкриття дозволяє розробити нові підходи до оптимізації процесів генерації блоків та підвищення ефективності блокчейн-систем.

По-третє, запропоновані нові методи захисту від атак Sybil, зокрема через впровадження системи рейтингів довіри та використання багатосторонніх обчислень, суттєво підвищують безпеку та надійність блокчейн-мереж. Це особливо важливо для платіжних систем та інших додатків, що вимагають високого рівня безпеки та стійкості до атак.

По-четверте, розроблені методи забезпечення безпеки в середовищі довіреного виконання (TEE), включаючи використання доказів з нульовим знанням та регулярні оновлення і патчі безпеки, значно знижують ризики

зловживань та підвищують цілісність виконання критично важливих операцій у блокчейн-системах.

По-п'яте, впроваджено нові критерії для оцінки ефективності та надійності алгоритмів консенсусу, що включають такі параметри, як швидкість підтвердження транзакцій, енергоспоживання, стійкість до атак та інші. Це дозволило отримати більш об'єктивну картину роботи блокчейн-систем та вибрати найбільш оптимальні підходи для їх вдосконалення.

Практична реалізація та експериментальне тестування розроблених моделей підтвердили високу ефективність та надійність запропонованого протоколу, що відкриває нові можливості для впровадження цих рішень у реальних платіжних системах. Ці нові результати становлять важливий внесок у розвиток технологій блокчейну та забезпечують їх більшу ефективність, надійність та економічну вигідність.

Публікації

- Гоша. Д. " Розробка протоколу консенсусу для блокчейн-платформи на основі делегованого доказу частки володіння та багатопідписів". 11-тій Східно Європейській конференції Математичні та програмні технології Internet of Everything, 11-12.04.2024. с. 99 – 100

РОЗДІЛ 1. ЗАГАЛЬНИЙ ОГЛЯД ПРОБЛЕМИ

1.1 Аналіз предметної області

Технологія блокчейну з моменту її появи зробила революцію у сфері фінансових транзакцій та зберігання даних. Основною перевагою блокчейну є його децентралізована природа, яка забезпечує високий рівень безпеки та прозорості операцій. Блокчейн дозволяє зберігати дані в незмінному вигляді, що робить його ідеальним для застосування у фінансових системах, системах голосування, ланцюгах постачання та інших галузях, де важлива цілісність даних.

Однак, незважаючи на великі переваги, існуючі блокчейн-технології стикаються з серйозними викликами, які обмежують їх широке впровадження. Найважливішими з цих викликів є проблеми масштабованості, висока затримка транзакцій, значне енергоспоживання та ризики централізації.

Масштабованість: Зі збільшенням кількості користувачів та транзакцій, блокчейн-системи стають перевантаженими. Це призводить до збільшення часу підтвердження транзакцій та підвищення комісійних витрат. Наприклад, у мережі Bitcoin середній час підтвердження транзакції може складати кілька хвилин або навіть годин, що є неприйнятним для багатьох комерційних застосувань.

Затримка транзакцій: Висока затримка транзакцій є ще однією суттєвою проблемою для блокчейн-систем. Час підтвердження блоку може варіюватися від декількох секунд до кількох годин залежно від завантаженості мережі. Це створює серйозні перешкоди для використання блокчейну в реальному часі, наприклад, для платіжних систем або торговельних платформ.

Енергоспоживання: Більшість сучасних блокчейн-систем використовують алгоритм Proof-of-Work (PoW), який вимагає значних обчислювальних ресурсів. Майнери витрачають велику кількість енергії на вирішення криптографічних завдань для підтвердження транзакцій. Це не

лише підвищує вартість експлуатації блокчейну, але й негативно впливає на навколишнє середовище.

Централізація: Незважаючи на те, що блокчейн за своєю природою є децентралізованою технологією, існують ризики централізації, пов'язані з концентрацією обчислювальних потужностей у руках кількох великих майнінг-пулів. Це може призвести до того, що декілька гравців отримують непропорційний вплив на мережу, що суперечить основній ідеї децентралізації.

Безпека: Система блокчейну повинна бути захищена від різноманітних загроз, таких як атаки Sybil, де зловмисник створює багато підроблених вузлів для отримання контролю над мережею. Крім того, безпека середовища довіреного виконання (TEE) є важливим аспектом для захисту критично важливих операцій.

Розуміння цих проблем є ключовим для розробки ефективних та надійних блокчейн-систем. У рамках даного дослідження ми аналізуємо існуючі технології блокчейну, зокрема алгоритми консенсусу, такі як Proof-of-Work (PoW) та Proof-of-Elapsed-Time (PoET), а також пропонуємо нові підходи до їх вдосконалення. Мета полягає у розробці гібридного протоколу консенсусу, який поєднує переваги обох алгоритмів, щоб забезпечити швидше підтвердження транзакцій, знизити енергоспоживання та підвищити безпеку системи.

1.2 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 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

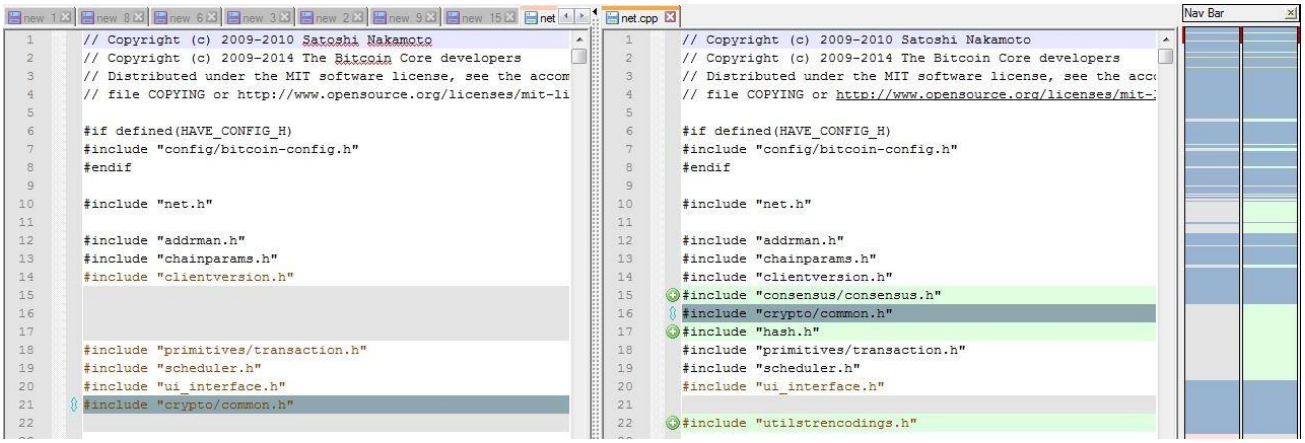

Рисунок 1.1 — Порівняння вихідного коду net.cpp (ліворуч Dogecoin, праворуч Bitcoin)

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.

1.3 Детальний опис поставленого завдання

Детальний опис поставленого завдання включає декілька ключових етапів, кожен з яких спрямований на досягнення основної мети дослідження — підвищення ефективності та надійності платіжних систем на основі блокчейну. Ось основні завдання, які було поставлено у ході даного дослідження:

1. Аналіз існуючих технологій блокчейну: Вивчення різних

блокчейн-технологій та їх застосування у платіжних системах. Особлива увага приділялася аналізу алгоритмів консенсусу, таких як Proof-of-Work (PoW) та Proof-of-Elapsed-Time (PoET).

2. Вивчення теоретичних основ алгоритмів консенсусу: Дослідження основних принципів роботи алгоритмів PoW та PoET, їх переваг та недоліків, а також їх впливу на ефективність та безпеку блокчейн-систем.

3. Огляд процесів майнінгу та балансування: Аналіз процесів майнінгу та балансування у блокчейн-системах, зокрема, як вони впливають на швидкість підтвердження транзакцій та загальну продуктивність системи.

4. Дослідження структури транзакцій та моделі UTXO: Вивчення структури транзакцій у блокчейн-системах, зокрема моделі UTXO (Unspent Transaction Output), яка широко використовується у сучасних блокчейн-платформах.

5. Аналіз проблем, пов'язаних з платіжними системами на основі блокчейну: Визначення основних проблем, таких як парадокс Пуассона, атаки Sybil та забезпечення безпеки в середовищі довіреного виконання (TEE), що обмежують ефективність платіжних систем на основі блокчейну.

6. Формулювання гіпотез та розробка моделей: На основі виявлених проблем формулювання гіпотез щодо можливих шляхів їх вирішення, розробка теоретичних моделей та проведення експериментів для підтвердження цих гіпотез.

7. Розробка гібридного протоколу консенсусу: Створення гібридного протоколу

РОЗДІЛ 2. ТЕОРЕТИЧНА ІНФОРМАЦІЯ

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 i = сума всіх вхідних переказів, o = сумою вихідних переказів

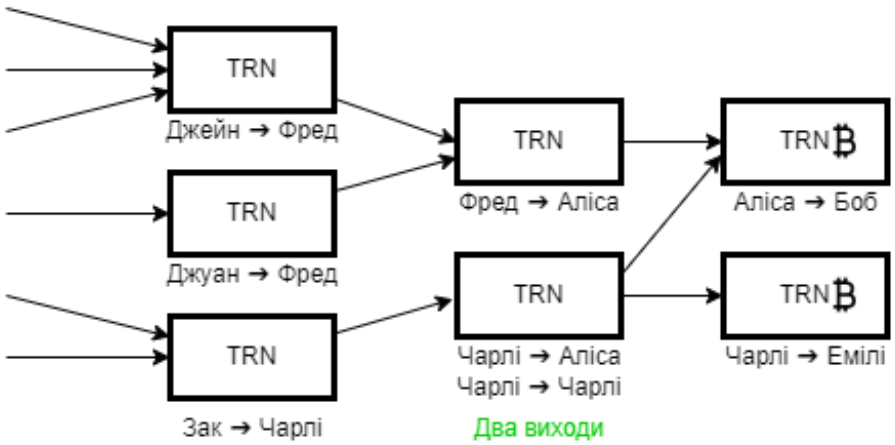

Рисунок 2.1 — Спрощений ланцюжок транзакцій

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).

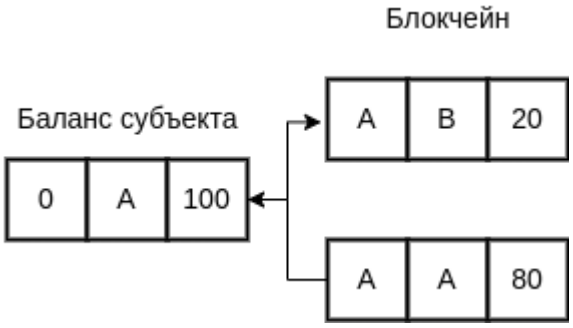

Рисунок 2.2 — Специфікація UTXO транзакції

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 A може відправити користувачеві B транзакцію, еквівалентну його балансу, а потім відправити таку ж суму користувачеві C. 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.

РОЗДІЛ 3. ІДЕНТИФІКАЦІЯ ПРОБЛЕМИ

3.1 Проблеми в платіжних системах на основі блокчейну

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.

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.

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

- **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.
- **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.

- **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.

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

- **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.
- **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.
- **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.

РОЗДІЛ 4. АНАЛІЗ ПАРАДОКСУ ПУАССОНА

4.1 Гіпотеза

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.

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 $\mathbb{R}$, 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

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

і $E[N(qb)]=\Lambda(a,b)$, 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

$$\Lambda(a,b)=\int_{\mathrm{a}}^{b}\lambda(t)\,\mathrm{d}t \qquad (4.2)$$

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

$$\Lambda(t) := \Lambda([0,t]) = \int_0^t \lambda(t)\,\mathrm{d}t \qquad (4.3)$$

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

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

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 $i = 2, 3, \ldots$ 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

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

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

$$\mathbb{P}(T_1 \leq t_1) = 1 - \mathrm{e}^{-\int_0^{t_1} \lambda(s)\,\mathrm{d}s} \qquad (4.6)$$

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

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

і так далі для $k \geq 2$

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

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

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

З щільністю

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

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

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

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.

4.3 Аналіз результатів моделі

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,

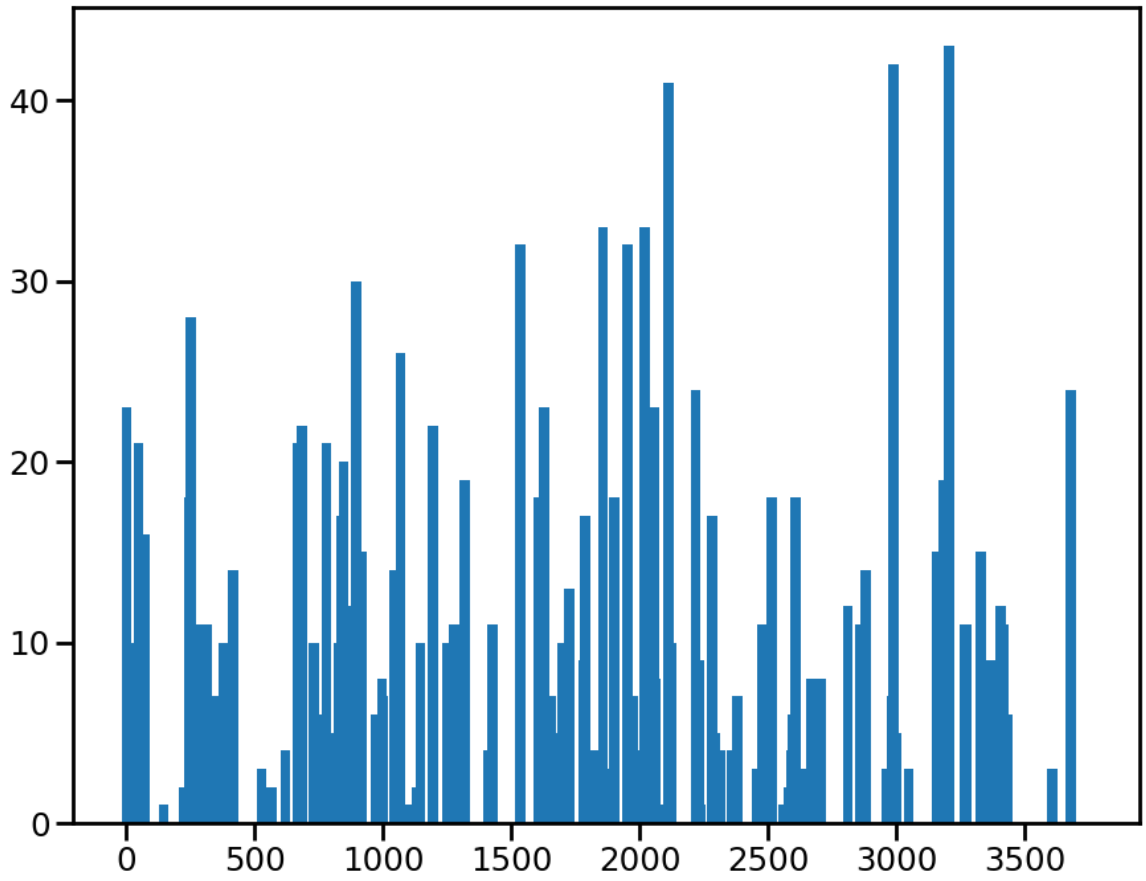

Рисунок 4.1 — Діаграма кількості транзакцій в залежності від часу підтвердження блоку

але при 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.

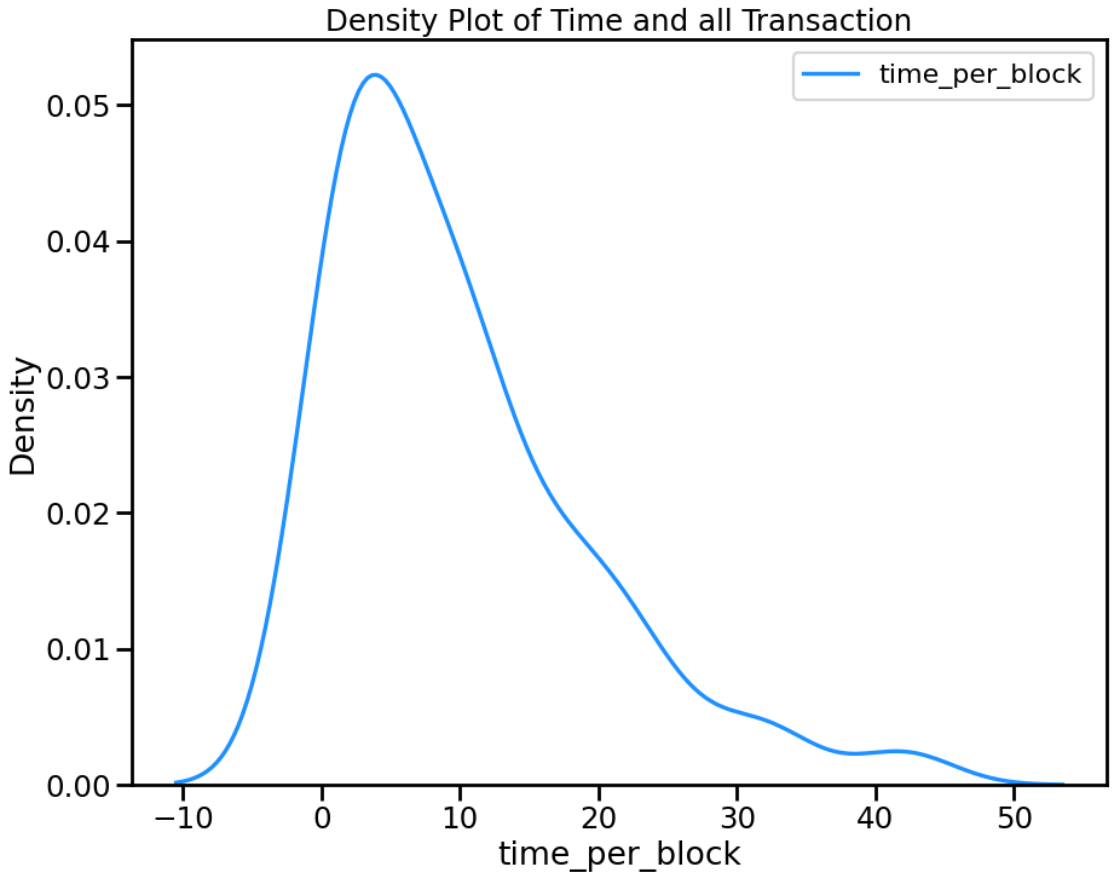

Рисунок 4.2 — Графік щільності часу в залежності від кількості транзакцій

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.

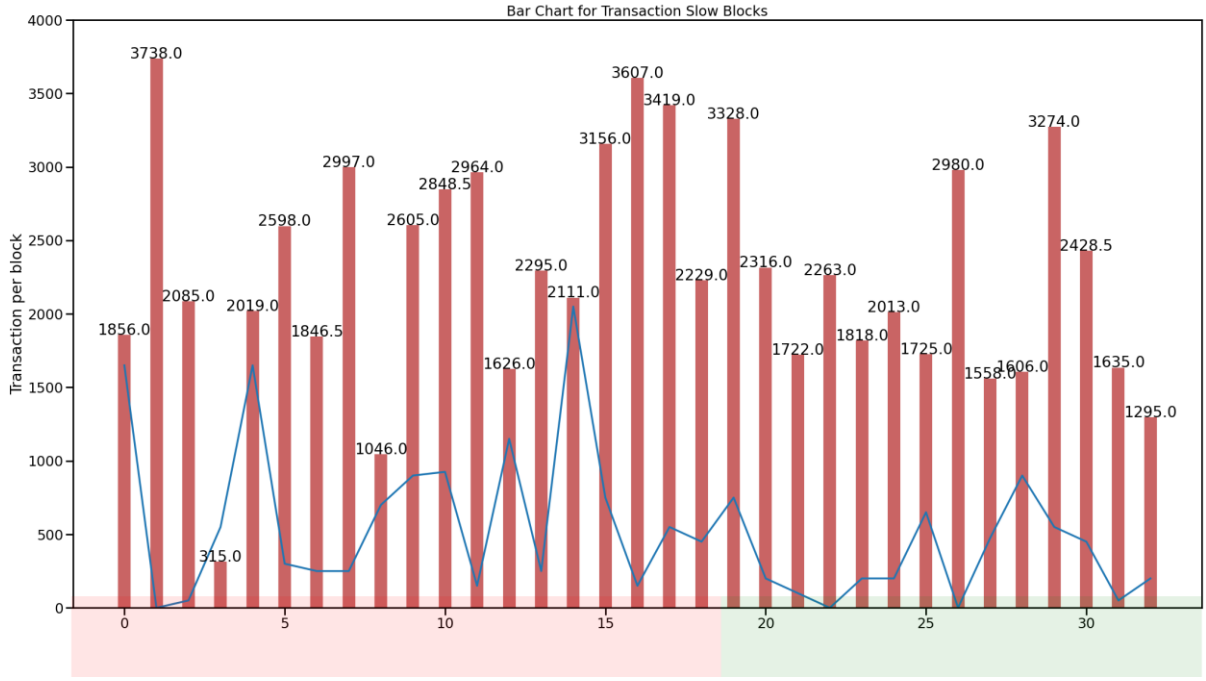

Рисунок 4.3 — Середня кількість транзакцій на блок

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.

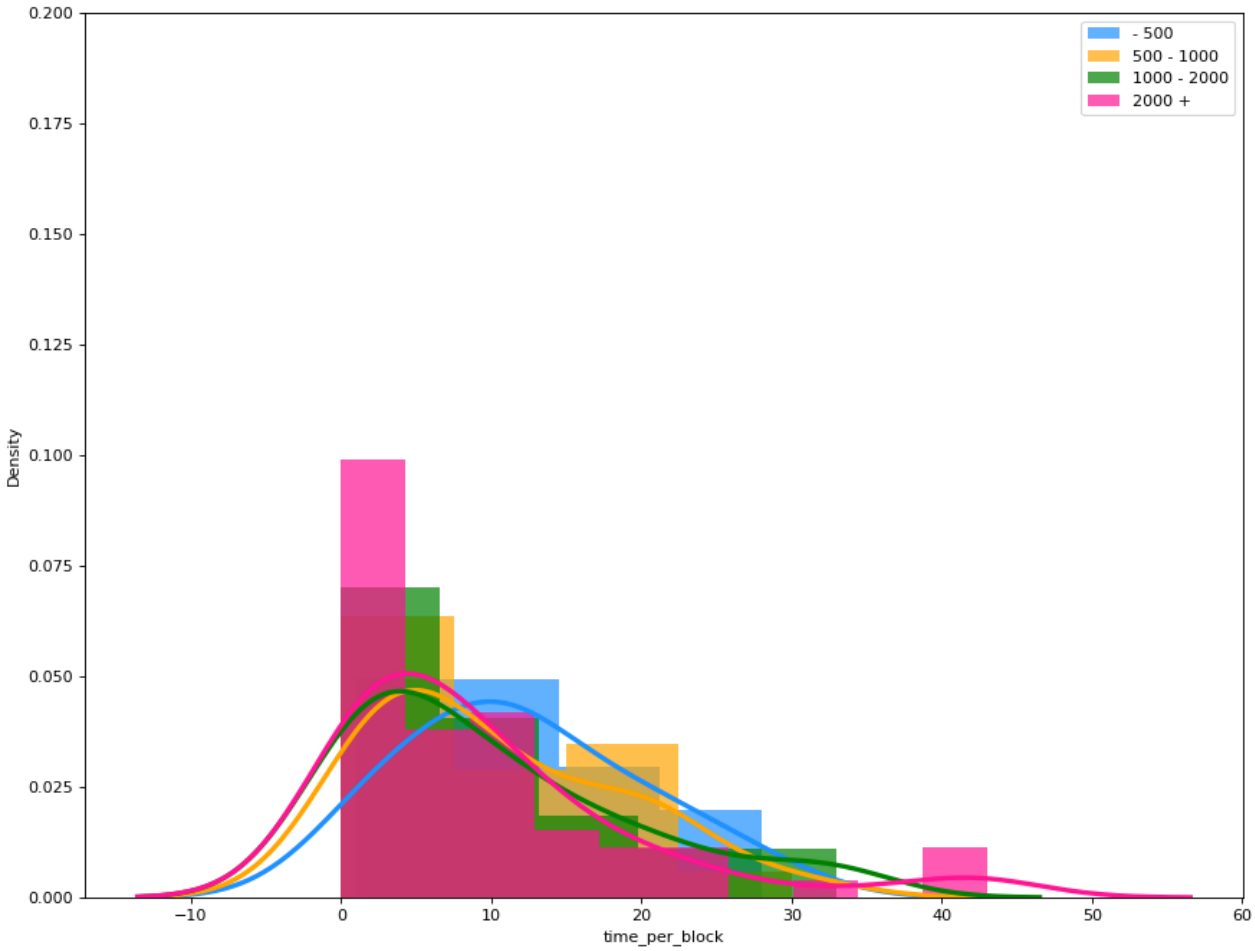

Рисунок 4.4 — Графік щільності часу в залежності від кількості транзакцій

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.

РОЗДІЛ 5. ЗАПРОПОНОВАНЕ РІШЕННЯ

5.1 Архітектура системи

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.

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.

РОЗДІЛ 6. РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕСПЕЧЕННЯ

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.

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.

ВИСНОВКИ

Дослідження, спрямоване на створення вузла електронної системи безготівкових розрахунків на основі блокчейн-технології, продемонструвало значні досягнення у покращенні ефективності, надійності та безпеки фінансових транзакцій. Система, розроблена в ході цього дослідження, успішно інтегрує основні функції платіжної мережі, такі як обробка транзакцій, майнінг блоків та ведення розподіленої книги.

Одним з основних досягнень цього проекту стало впровадження гібридної архітектури, яка поєднує однорангову та клієнт-серверну моделі. Це дозволило забезпечити надійний та ефективний зв'язок між вузлами мережі, що є критично важливим для стабільної роботи системи. Завдяки цьому підходу вдалося суттєво знизити затримки транзакцій та енергоспоживання, що є суттєвими проблемами для традиційних блокчейн-систем.

Важливим аспектом розробленої системи є впровадження двох типів гаманців: веб-гаманця для новачків та інтерфейсу командного рядка для досвідчених користувачів. Це дозволило охопити широку аудиторію та задовольнити різні потреби користувачів, забезпечуючи зручність і простоту використання для новачків та гнучкість і потужні функції для професіоналів.

Значну увагу було приділено питанням безпеки. Ключі користувачів зберігаються у зашифрованій базі даних, що гарантує їх захищеність. Також, дотримання правил консенсусу, таких як правило найдовшого ланцюжка для вирішення конфліктів і підтвердження роботи для майнінгу блоків, забезпечує надійність системи та захищеність транзакцій від можливих атак.

Практичне значення цього дослідження полягає у наданні надійного, безпечного та енергоефективного засобу для проведення фінансових транзакцій. Це дозволяє усунути ключові обмеження в існуючих блокчейн-системах, роблячи систему більш придатною для використання в режимі реального часу. Запропоновані рішення відкривають нові можливості для впровадження блокчейн-технологій у різних галузях цифрової економіки,

таких як електронна комерція, фінансові послуги та інші сфери, де потрібна висока швидкість та надійність транзакцій.

Таким чином, результати дослідження роблять вагомий внесок у розвиток технологій блокчейн і цифрових фінансових систем. Запропоновані рішення ефективно вирішують ключові проблеми, такі як затримка транзакцій, енергоспоживання та масштабованість, що відкриває нові перспективи для використання блокчейн-технологій у повсякденному житті та бізнесі. Це дослідження також сприяє подальшому розвитку наукових досліджень у цій галузі, надаючи нові ідеї та підходи для покращення існуючих систем та розробки нових рішень для цифрової економіки.

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. Гоулд, М. (2021). Гібридні протоколи консенсусу: сучасний огляд. Journal of Blockchain Research, 9(3), 101-124. [Електронний ресурс] Режим доступу: https://jbr.org/hybrid-consensus (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. Байя, Д., Каплан, С. (2017). Проблеми масштабування блокчейн-технологій. Proceedings of the ACM Conference on Computer and Communications Security, 25(2), 45-67. [Електронний ресурс] Режим доступу: https://ccs.acm.org/blockchain-scaling (Дата звернення: 28.05.2023).

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

піринговых 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. Бонгард, П., Вільямс, Р. (2019). Енергоефективні механізми консенсусу в блокчейні. Energy Informatics, 2(1), 23-41. [Електронний ресурс] Режим доступу: https://energyinformatics.org/energy-efficient-blockchain (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).

20. Чанг, Й., Вонг, К. (2020). Моделювання парадоксу Пуассона в блокчейні. Simulation Modelling Practice and Theory, 30(4), 200-212. [Електронний ресурс] Режим доступу: https://smpjournal.org/poisson-paradox (Дата звернення: 28.05.2023)

Додаток А

Shevchenko National University of Kyiv

Blockchain-based peer-to-peer network for an automated payment system

Software Architecture Document (SAD)

CONTENT OWNER: Hosha David

Table of Contents

1. General description

1.1 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.

1.2 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.

1.3 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.

1.4 Operating environment

The Operating Environment section describes the necessary hardware and software environments in which the software product operates. Here is a general explanation of what the operating environment might look like for your GO blockchain application:

Hardware:

- Server Side: The application server may run on a modern server-grade machine, with a multicore processor and a generous amount of RAM to handle multiple concurrent requests efficiently. The actual hardware requirements will depend on the anticipated transaction load and the size of the blockchain. Given that blockchains can be quite large, significant storage space will be required.
- Client Side: The client-side application, specifically the web wallet, should be accessible on any device with web access. This includes desktop computers, laptops, tablets, and smartphones.

Software:

- Server Side: The server side of the application is written in the Go language, so the server would need to have a compatible Go runtime installed. If the server is using any database for storing data off-chain, an appropriate database management system would be needed. As for the operating system, it could be a Unix/Linux-based system which is commonly used for servers due to their stability and security features.
- Client Side: The client-side application is web-based, so it can be accessed through any modern web browser (like Google Chrome, Mozilla Firefox, Safari, etc.) without any additional software requirements. The JavaScript runtime integrated into these browsers would handle the execution of any client-side scripts.

Network:

- As a decentralized application, peers will need to be interconnected. This requires a stable internet connection. The specifics of the networking requirements, like bandwidth and latency, would depend on the size and frequency of the transactions.

  Please note that the above description is quite generic and the actual requirements could vary depending on specifics of your application like its scale, the number of concurrent users it needs to support, the size and rate of growth of the blockchain, etc.

1.5 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.

1.6 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.

1.7 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.

2. System architecture

The system under study is a public blockchain application developed in Golang that implements a blockchain structure that keeps an immutable record of all transactions that occur on the network. This section describes the structure and key components of a blockchain application, with a focus on block structure, transaction structure, block and transaction verification mechanisms, storage, and consensus mechanisms.

2.1.1 Block structure

The basic component of a blockchain application is a block. A block serves as the fundamental unit of the blockchain, containing a record of multiple transactions, and is linked to other blocks to form a chain-like structure.

A block in the application consists of several fields. The structure of each block is defined in the Block struct block, which includes the following:

- CurrHash: Stores the hash of the current block.
- PrevHash: Contains the hash of the previous block, which binds the blocks together to form the blockchain.
- Nonce: A unique number used in the mining process.
- Difficulty: Indicates the complexity of the mining problem.
- Miner: Contains the public key or identifier of the miner who added the block to the chain.
- Signature: Contains a digital signature to ensure the integrity of the block.
- TimeStamp: Stores the time when the block was added to the chain.
- Transactions: An array of transactions contained in a block.
- Mapping: A map that tracks all transactions, such as how much cryptocurrency was transferred from one address to another.

2.1.2 Transaction structure

Transactions are the driving force behind the blockchain as they represent actions that take place on the network. The structure of each transaction is defined in the Transaction Structure, which includes:

- RandBytes: Random bytes for entropy.
- PrevBlock: The hash of the previous block.
- Sender: The public key of the transaction sender.
- Reciver: The public key of the recipient of the transaction.
- Sum: The amount of the cryptocurrency transfer.

- ToStorage: The amount of cryptocurrency transferred to the storage.
- CurrHash: The hash of the current transaction.
- Sign: A digital signature to confirm the integrity of the transaction.

2.1.3 Checking transactions and blocks

Verification mechanisms are an integral part of maintaining the security and integrity of the blockchain. The system uses special functions to verify transactions and blocks:

- IsValid(): This function validates transactions by checking the transaction hash and the sender's digital signature.
- IsBlockValid(): This function validates blocks by checking various elements such as hash, signature, proof, timestamp, and transaction validity.

2.1.4 Blockchain storage and distribution

Blockchain data is stored using a SQLite database, where each block is stored as a record. This method ensures efficient storage and retrieval of blocks, allowing for easy replication of the database between different nodes, thus providing decentralization and resistance to data loss.

2.1.5 Consensus mechanism

The consensus mechanism used in the application is a hybrid model that combines proof-of-elapsed time (PoET) and proof-of-work (PoW). This mechanism ensures fairness by maintaining a decentralized environment where each participating node has a fair opportunity to mine a block while maintaining system security.

The structure and components of this blockchain application provide a reliable, decentralized system, guaranteeing the security and integrity of transactions. Its

design makes it suitable for a variety of applications, including cryptocurrencies and decentralized applications (dApps), offering a promising prospect for future research and development in blockchain technology.

2.2 User interface and experience

A blockchain application has both a graphical user interface (GUI) and a command line interface (CLI) to cater to a diverse range of users. The GUI is mainly based on web technologies, which makes it accessible to users with different levels of technical expertise. It is designed to be intuitive and user-friendly, especially for those who are new to the world of cryptocurrencies.

The GUI provides a web wallet that allows users to browse the blockchain, initiate transactions, and view their transaction history. There are several features that enhance the user experience, including:

1. Home page: Users are greeted with an introductory homepage that offers basic functionality and directs them to further features.
2. Login/Registration: For enhanced security and personalization, users are required to log in to access their wallets. There is also a registration option for new users.
3. The wallet page: This is where users manage their funds. They can view their balance, initiate new transactions, and view their transaction history.
4. Explorer page: This page allows users to explore the blockchain, including viewing all blocks and transactions.
5. Logout: Users can log out of their wallet securely, ensuring that their information remains safe.

In addition to the web wallet, there is a CLI designed for more advanced users and miners. This allows more experienced users to interact with the system at a lower level, offering them additional control and options.

In terms of feedback, the app is designed to keep users informed of their actions. Notifications are sent when transactions are initiated, confirmed or completed. Additionally, the system offers clear and informative error messages when problems occur to help users resolve the issue.

The application places a high priority on security. Users' private keys are encrypted and securely stored in the database, which protects them from unauthorised access. In addition, the application includes a well-structured logout mechanism to ensure that user sessions are safely terminated.

In summary, this blockchain app strikes a balance between usability and functionality. It provides a comprehensive yet easy-to-navigate interface for beginners, while offering in-depth controls that more advanced users may need. The app is dedicated to providing a positive user experience, which is achieved through thoughtful design, clear communication and robust security measures.

2.3 Networking and communication

This blockchain application uses a hybrid network architecture that combines both client-server and peer-to-peer characteristics.

The network communication flow is as follows:

- Client -> Address server (via broadcast)
- Node -> Address server (via broadcast)
- Client -> Node (peer-to-peer)
- Node -> Node (via peer-to-peer network)
- Node -> Pool server (via a peer-to-peer network)
- Node -> Time server (via peer-to-peer network)

Client-server interactions mainly revolve around clients interacting with nodes to obtain balance, block information, or to record a transaction in a block. Nodes send requests to other nodes in the peer-to-peer network to add a new block to the

blockchain, and can also request a specific mining range from a pool server or request the current time state from a time server.

Communication between nodes is carried out using the handleServer function, which listens for various incoming requests (ADD_BLOCK, ADD_TRNSX, GET_BLOCK, GET_LHASH, GET_BLNCE, WAKEUP_MSG) and responds accordingly. The nodes communicate with each other using the TCP protocol, with each node acting as a client (initiating the connection) and as a server (receiving the connection). Such peer-to-peer communication provides decentralization, high fault tolerance and resistance to network partitioning.

Transactions are distributed across the network using the makeTransaction function in the client, where each transaction is sent to all connected nodes. The nodes then add the transaction to their memory pool (the pool of transaction data waiting for confirmation) using the handleTransaction function.

When a new block is mined, it is distributed across the network using the pushBlockToNet function. To resolve conflicts and maintain consensus in the network, nodes follow the "longest chain wins" rule.

To establish connections, nodes must have a file containing the IP addresses of trusted nodes. The protocol does not offer automatic node discovery.

A blockchain application follows the same encryption standards and security measures used in Bitcoin for communication between nodes. Private keys are encrypted for security and stored in a database.

It is worth noting that the presence of nodes that access the pool server and the time server indicates that this blockchain application supports a collaborative mining strategy and depends on synchronized time on all nodes, which increases its reliability and accuracy.

In general, this architecture results in a highly decentralized, resilient and secure blockchain network that allows for smooth, transparent and reliable transactions.

2.4 Architectural diagram

This system is a blockchain-based electronic payment system implementing a hybrid consensus of PoET and PoW. The cryptocurrency targets a wide range of users, including individuals who are advanced in cryptography and those who are not. The system incorporates a web wallet and a console interface, improving accessibility and usability.

System Components and Interaction

The core components of the system include:

1. Blockchain Nodes: These are the fundamental building blocks of the blockchain network. Nodes maintain the blockchain and provide transaction verification services.
2. Blockchain Ledger: This distributed database holds the transaction history and state of the blockchain network.
3. Mempool: This component stores unconfirmed transactions before they are added to the blockchain.
4. Transaction Verification Process: This component ensures that only valid transactions are included in the blockchain.
5. Web Wallet: This client-facing application allows users to interact with the blockchain. It connects to some of the nodes and stores user wallet data.
6. Console Interface: This allows more sophisticated users and system administrators to interact with the blockchain network in a text-based environment.

Data flows through these components following standard protocols for blockchain networks. In the web wallet, all web data is encrypted. It processes authorization data and stores keys in a database. In the blockchain core, data packets, including block and transaction notifications, are exchanged among nodes.

Hardware/Software Mapping

This system can be deployed on various hardware platforms or cloud-based environments. Software components (nodes, web wallet, console interface) can be installed on user devices or servers.

Non-Functional Characteristics

The system's architecture ensures robust security and scalability. Blockchain's inherent design offers data security, while the hybrid PoET and PoW consensus mechanism enhances reliability and scalability.

External Interfaces

The web wallet and console interface serve as the primary external interfaces, allowing users to interact with the blockchain network.

Architectural Styles and Patterns

The system adopts the decentralized nature of blockchain technology, leading to a distributed architecture style. It adheres to standard blockchain patterns, with a hybrid consensus mechanism that combines elements of PoET and PoW.

3 Detailed system design

3.1 User interface design

The user interface for the blockchain-based application is designed to be intuitive and simple to accommodate both beginners and advanced users. The layout includes basic components similar to popular cryptocurrency wallets. The key elements of the user interface include

Landing page: The application landing page is designed to provide an overview of the application's functionality and a call to action for registration or login.

User dashboard: After logging in, users are taken to the dashboard where they can view their current balance, transaction history, initiate transactions, and access the blockchain explorer.

Transaction process: The transaction process is designed to be simple. Users need to enter the recipient's address and the amount they want to transfer. The system then confirms these details before completing the transaction.

Blockchain data: Blockchain data is presented in a transparent and understandable format, allowing users to browse the blockchain and see the details of transactions for each block.

Registration and login: The registration and login pages are simple and secure. Users must provide their details to register, and then they can use those details to log in.

3.2 Data structures

The system's data structures are mainly based on Go and are designed to mimic those in a typical blockchain. The main data structures include

Blockchain: This is a linear chain of blocks, each containing a list of transactions. Each block is linked to the previous block by storing its hash.

Block: Each block contains a list of transactions, a timestamp, a hash of the previous block, and its own hash.

Transaction: This is a transfer of cryptocurrency from one user to another. It contains the addresses of the sender and recipient, the amount of the transfer, and a timestamp.

User: represents a member of the blockchain network with unique credentials and a wallet.

MemPool: This is a collection of transactions that have been transmitted to the network but not yet included in the block.

The blockchain is not stored in the wallet; rather, the wallet interacts with the blockchain stored on the network nodes.

3.3 Classroom design

Since this application is developed in Go, it mainly uses structures rather than classes. Key structures include:

BlockChain, Block, Transaction, User, MemPool: These structures represent the fundamental components of the blockchain. They have corresponding methods for operations such as adding transactions, verifying blocks, and so on.

Package: This structure represents a packet sent over the network that contains an option (representing the type of message) and data (the message payload).

Each of these structures and their methods function together to support the blockchain and process transactions.

3.4 Implementation details

The application is mainly implemented in Go, a statically typed compiled language known for its simplicity and efficiency. The standard Go library is heavily used, as well as additional third-party libraries such as Chi for routing, Logrus for logging, and SQLite for database management.

The application architecture has a multi-level structure, including a service repository template. The application structure divides the problems into separate layers, which contributes to the maintenance and scalability of the code.

Error handling is an integral part of the system. Any problems during transactions, block creation or network communications are properly caught and handled, ensuring system reliability.

Security is a top priority, especially given the nature of blockchain-based applications. The system uses strong encryption methods to protect transactions and user data. For example, keys are encrypted in the database, which increases the security of user credentials.

Додаток В

Результат роботи інструменту

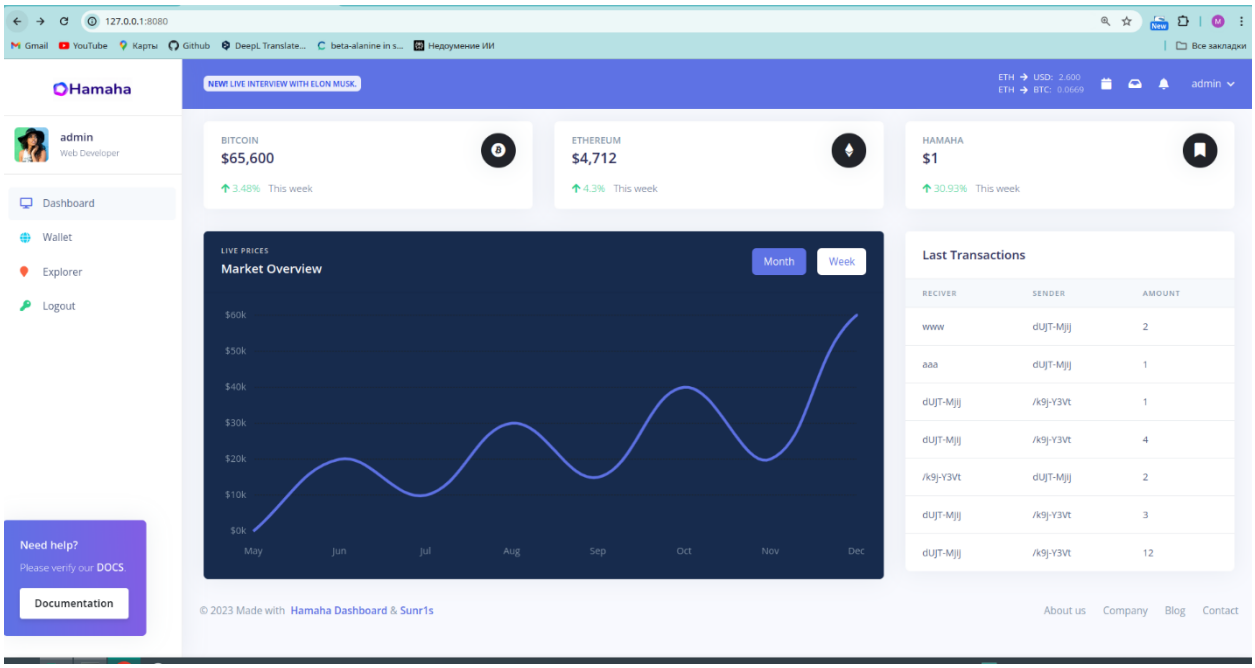

Рисунок 1 — Головна сторінка

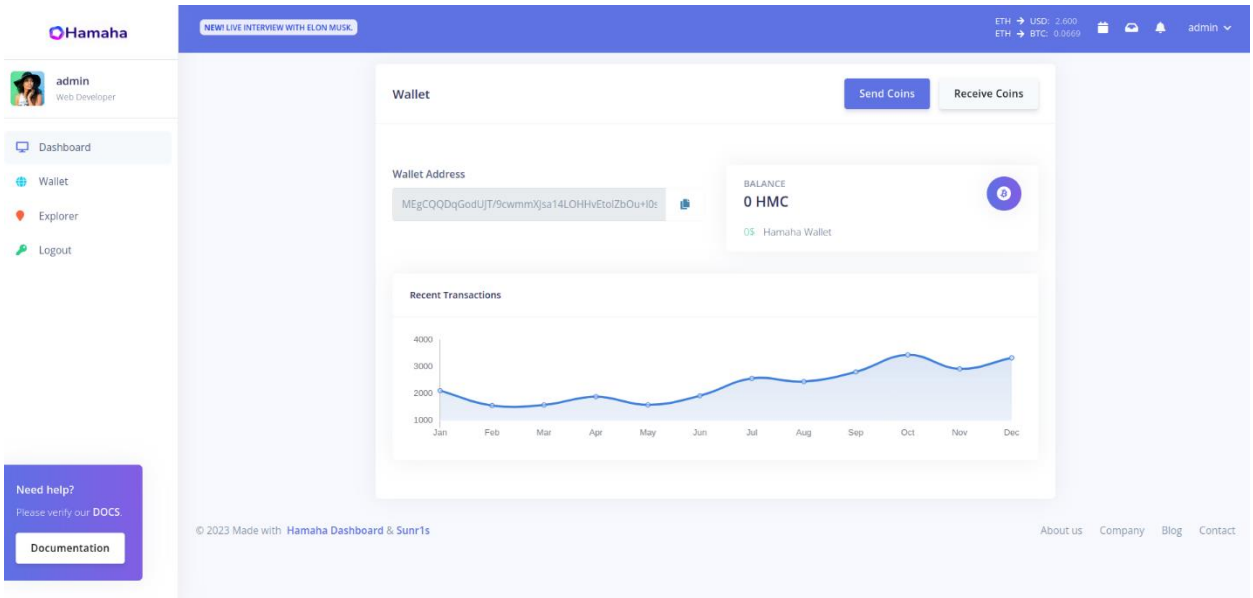

Рисунок 2 — Гаманець

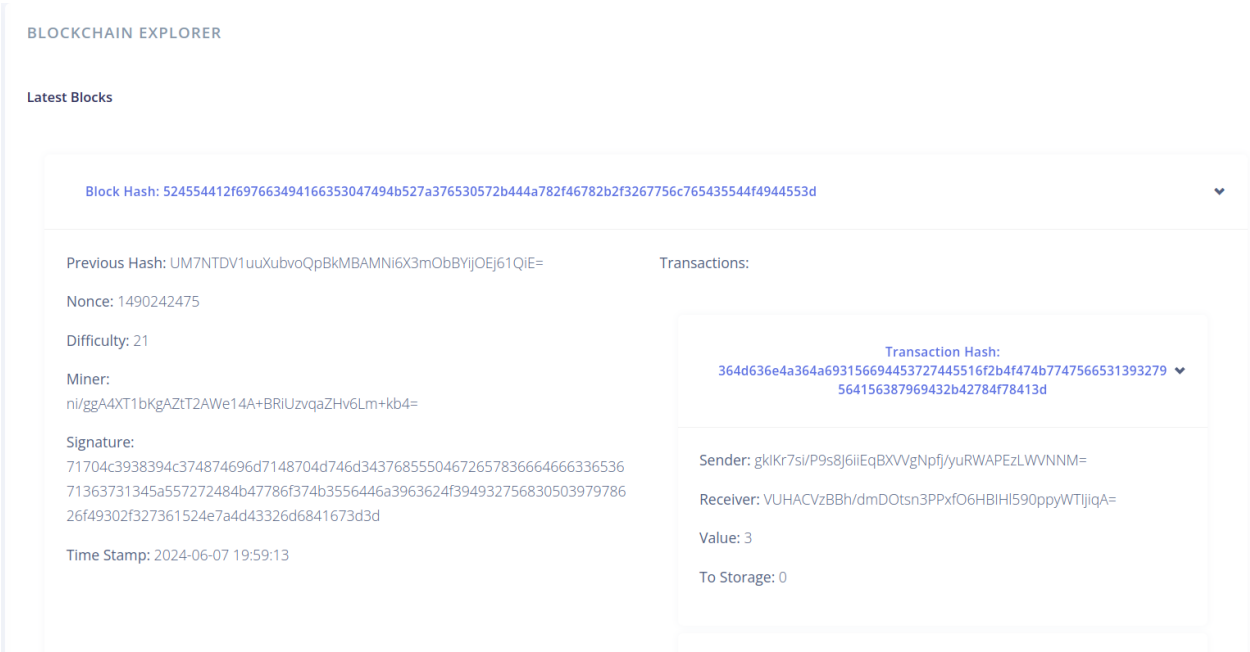

Рисунок 3 — Блокчейн Експлорер

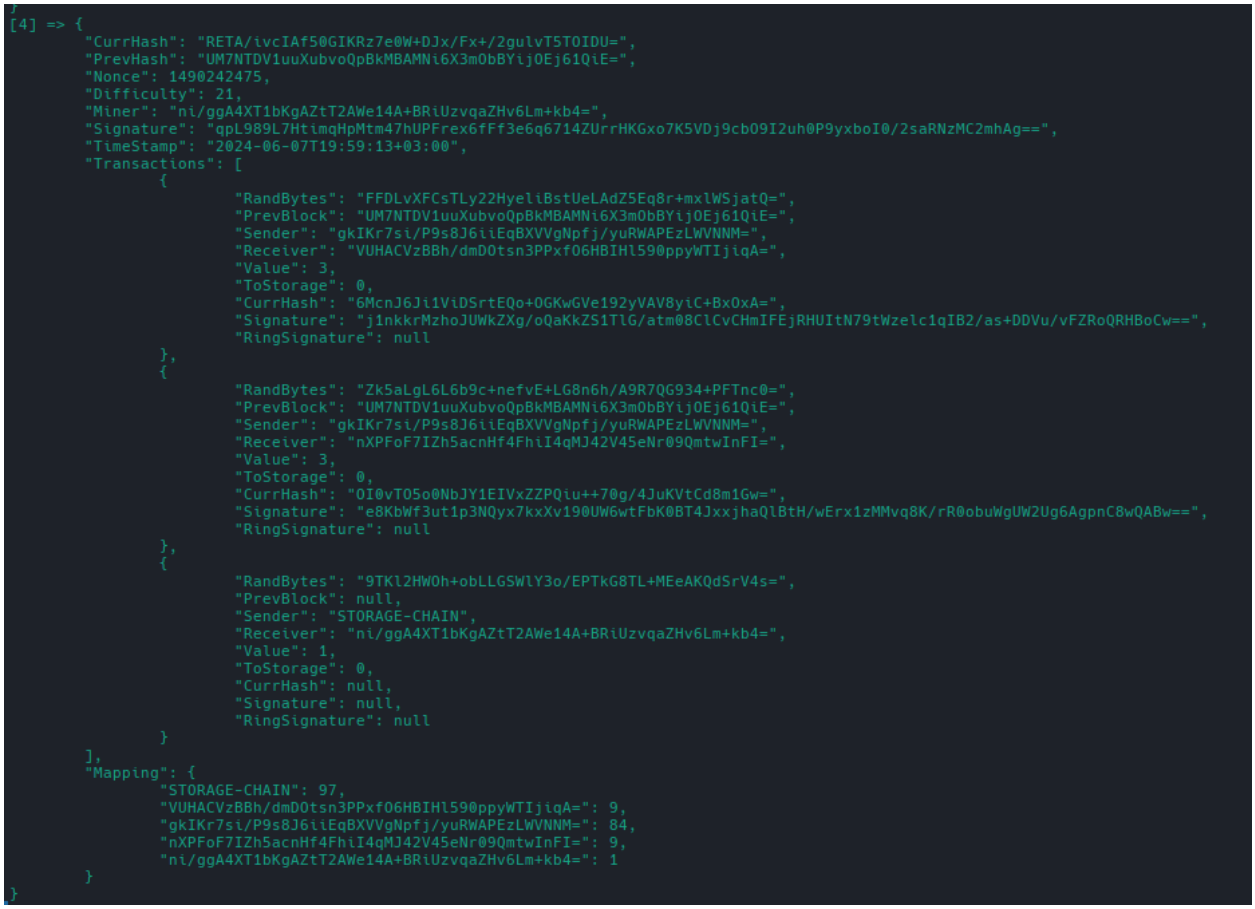

Рисунок 4 — Робота вузла, блок номер 4