Національний технічний університет України
«Київський політехнічний інститут
імені Ігоря Сікорського»
Фізико-технічний інститут
Кафедра математичних методів захисту інформації

Рівень вищої освіти: другий (магістерський) за освітньо–професійною програмою

Спеціальність: 113 «Прикладна математика»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедрою
_____ М.М.Савчук
(підпис) (ініціали, прізвище)
«___»_____2019 р.

ЗАВДАННЯ
на магістерську дисертацію студента

Богуцького Олександра Миколайовича

1. Тема дисертації: «Оцінки ефективності протоколів узгодження Proof-of-Activity та Proof-of-Burn для блокчейнів», науковий керівник дисертації: д.т.н., с.н.с. Кудін А. М., затверджені наказом по університету від _____р. № _____

2. Термін подання студентом дисертації _____

3. Об’єкт дослідження: процес досягнення узгодження в розподілених криптовалютних системах Proof of Activity та Proof of Burn.

4. Предмет дослідження: визначення значень показників ефективності побудови блоку у протоколах Proof-of-Activity та Proof-of-Burn.

5. Перелік завдань, які потрібно розробити:
- аналіз протоколів узгодження Proof-of-Activity та Proof-of-Burn для блокчейну та опис їхньої математичної моделі;
- визначення поняття «ефективності» для даних протоколів, обчислення ймовірностей генерування наступного блоку блокчейну у кожній із систем та порівняння отриманих результатів із іншими поширеними протоколами.

6. Орієнтовний перелік ілюстративного матеріалу:

- Ілюстрації до структур і процесів в технологіях, розглянутих в даній роботі;
- Таблиця порівнянь властивостей обраних криптовалют, приватних блокчейнів та нових підходів.

7. Орієнтовний перелік публікацій відсутній.

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	д.т.н., с.н.с. Кудін А. М.		
2	д.т.н., с.н.с. Кудін А. М.		
3	д.т.н., с.н.с. Кудін А. М.		

9. Дата видачі завдання

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Ознайомлення з джерелами інформації та літературою у визначеній області наукової роботи	листопад 2017 р.	Виконано
2	Аналіз та визначення проблемних аспектів обраної області дослідження	січень 2018 р.	Виконано
3	Визначення теми наукової магістерської дисертації	березень 2018 р.	Виконано
4	Постановка задач дослідницької роботи та переліку потенційних методів для їх виконання	вересень 2018 р.	Виконано
5	Опис Proof of Work та Proof of Stake протоколів, що лежать в основі Proof of Activity та Proof of Burn алгоритмів, які аналізуються у науковій роботі	грудень 2018 р.	Виконано

6	Детальний аналіз протоколів, опис їхньої математичної моделі, обчислення ймовірностей ефективності реалізації цих систем.	лютий 2019 р.	Виконано
7	Оформлення висновків шляхом порівняння отриманих у науковій роботі результатів із найпоширенішими протоколами. Пропонування подальших досліджень на основі отриманих результатів.	квітень 2019 р.	Виконано

Студент _____ Богуцький О. М.
(підпис)

Науковий керівник дисертації _____ Кудін А. М.
(підпис)

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ ЗАХИСТУ ІНФОРМАЦІЇ

«На правах рукопису»
УДК _____

«До захисту допущено»
В.о. завідувача кафедрою
_____ М.М.Савчук
(підпис) (ініціали, прізвище)
“___” _____ 2019 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 113 «Прикладна математика»

на тему: «Оцінки ефективності протоколів узгодження Proof-of-Activity та Proof-of-Burn для блокчейнів»

Виконав: студент 6 курсу, групи ФІ-73мн
Богуцький Олександр Миколайович
(підпис)

Керівник професор, д.т.н., с.н.с. Кудін А. М. А. М.
(підпис)

Рецензент к.т.н. Проскуровський Р. В.
(підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2019 року

РЕФЕРАТ

Роботу виконано на 56 аркушах, вона містить перелік посилань на використані джерела з 28 найменувань. У роботі наведено 6 рисунків.

Метою даної дипломної роботи є оцінка ефективності (за критерієм децентралізації) *Proof of Activity* та *Proof of Burn* протоколів консенсусу блокчейну.

Об'єктом дослідження є процес досягнення узгодження в розподілених криптовалютних системах *Proof of Activity* та *Proof of Burn*.

Предметом дослідження є визначення значень показників ефективності побудови блоку у протоколах *Proof of Activity* та *Proof of Burn*.

БЛОКЧЕЙН, ПРОТОКОЛИ КОНСЕНСУСУ, PROOF OF ACTIVITY, PROOF OF BURN

ABSTRACT

The thesis is presented in 56 pages. It contains bibliography of 28 references. 6 figures are given in the thesis.

The object is is a process of consensus achievement in distributed cryptocurrency systems *Proof of Activity* та *Proof of Burn*.

The subject is to determine the effectiveness values indicators for block mining in the protocols *Proof of Activity* та *Proof of Burn*.

BLOCKCHAIN, CONSENCUS PROTOCOLS, PROOF OF ACTIVITY, PROOF OF BURN

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

PoW — протокол Proof-of-Work

PoB — протокол Proof-of-Burn

PoS — протокол Proof-of-Stake

PoA — протокол Proof-of-Activity

СМО — Системи масового обслогувування

ВСТУП

Актуальність дослідження. Актуальність даного дослідження полягає у тому, що без нього ви не одержите диплом про вищу освіту. Відповідно, ви повинні оформити результати вашого дослідження належним чином.

Оскільки дані протоколи є новими та малодослідженими, постає питання у дослідженні оцінки майнингу блоків у цих системах. Зокрема, цікавим є визначення цієї оцінки та порівняння із іншими протоколами. Також важливе значення відіграє визначення потенційно нових параметрів цих систем.

Метою дослідження є оцінка ефективності (за критерієм децентралізації) *Proof of Activity* та *Proof of Burn* протоколів консенсусу блокчейну.

Для досягнення поставленої мети необхідно розв'язати **задачу дослідження**, що полягає у вирішенні таких завданнь:

1) провести огляд опублікованих та інтернет джерел за тематикою даного наукового дослідження;

2) провести детальний аналіз протоколів консенсусу *Proof of Activity* та *Proof of Burn*, а також протоколів *Proof of Work* та *Proof of Stake*, що лежать у їхній основі.

3) побудувати математичні моделі систем *Proof of Activity* та *Proof of Burn* із необхідними математичними обчисленнями.

4) обчислити ймовірності генерації наступного блоку випадковим користувачем для протоколів *Proof of Activity* та *Proof of Burn*.

5) порівняти отримані результати із більш поширеними протоколами консенсусу.

Об'єктом дослідження є процес досягнення узгодження в розподілених криптовалютних системах *Proof of Activity* та *Proof of Burn*.

Предметом дослідження є визначення значень показників

ефективності побудови блоку у протоколах *Proof of Activity* та *Proof of Burn*.

При розв'язанні поставлених завдань використовувались такі *методи дослідження*: і тут коротенький перелік (наприклад, але не обмежуючись: методи лінійної та абстрактної алгебри, теорії імовірностей, математичної статистики, комбінаторного аналізу, теорії кодування, теорії складності алгоритмів, методи комп'ютерного та статистичного моделювання)

Наукова новизна отриманих результатів полягає у розробці критеріїв ефективності та визначенні числових характеристик цих показників ефективності для протоколів консенсусу *Proof of Activity* та *Proof of Burn*.

Практичне значення результатів полягає у визначенні кола та правил застосування блокчейну на протоколах *Proof of Activity* та *Proof of Burn*.

1 КОНЦЕПЦІЯ РОЗПОДІЛЕНИХ КРИПТОВАЛЮТНИХ СИСТЕМ

Сьогодні криптовалюти (англ. *Cryptocurrency*) стали глобальним явищем, відомим більшості людей. В той час, як вони все ще залишаються якимись прискіпливими і не зрозумілими багатьом людям, банки, уряди і багато компаній усвідомлюють його важливість та впроваджують у свої системи та застосунки.

Починаючи з 2016 року, великі банки, великі бухгалтерські фірми, відомі компанії з розробки та впровадження програмного забезпечення та уряди почали активне дослідження криптовалют, публікуючи про них статті або розпочинаючи власний так званий блокчейн-проект.

Але поза межами всього шуму і прес-релізів переважна більшість людей – навіть банкіри, консультанти, вчені та розробники – мають дуже обмежені знання про криптовалюти. Вони часто не розуміють основних понять.

Мало хто знає, але криптовалюти з'явилися як побічний продукт іншого винаходу. Сатоші Накамото (*Satoshi Nakamoto*), невідомий винахідник *Bitcoin*, першої і все ще дуже важливої криптовалюти, ніколи не мав наміру винаходити її. У своєму анонсі в кінці 2008 року, Сатоші розповів, що він розробив «*Peer-to-Peer* електронну систему грошових коштів» – *Bitcoin*[1].

Найбільш важливою частиною винаходу Сатоші є те, що він знайшов спосіб побудувати децентралізовану систему цифрових грошей. У дев'яностих роках було зроблено багато спроб створення цифрових грошей, але всі вони провалилися[1].

Після того, як всі централізовані спроби провалилися, Сатоші спробував побудувати цифрову систему готівки без центрального органу. Подібно мережі однорангових вузлів для спільного використання

файлів[1].

Це рішення стало народженням криптовалют. Вони виконують роль відсутнього елемента, який Сатоші знайшов для реалізації цифрових грошей.

Для реалізації цифрових коштів потрібна платіжна мережа з рахунками, залишками та транзакціями. Однією з головних проблем, яку має вирішувати кожна платіжна мережа, є запобігання так званому повторним витратам: запобігти тому, щоб одна організація витрачала одну суму двічі. Зазвичай це робиться центральним сервером, який веде облік залишків та усіх транзакцій.

Головною особливістю децентралізованої мережі є те, що у неї немає такого сервера. Таким чином, вам потрібна кожна окрема сутність мережі для виконання цієї роботи. Кожен учасник мережі повинен мати список з усіма транзакціями, щоб перевірити, чи дійсні майбутні транзакції.

Але як ці суб'єкти можуть досягнути згоди щодо цих записів?

Якщо учасники мережі не згодні лише з одним, незначним балансом, все порушено. Вони потребують абсолютного консенсусу. Зазвичай, ви знову приймаєте центральний орган, який оголошує правильний стан балансів. Але як можна досягти консенсусу без центрального органу?

Відповідь на це питання свого часу ніхто не знав та й насправді ніхто не вірив, що це можливо, доки Сатоші не з'явився із своєю ідеєю.

Його головним нововведенням було досягнення консенсусу без центральної влади. Криптовалюти є частиною цього рішення – тією частиною, яка зробила рішення цієї проблеми і допомогла змінити світ.

Власне криптовалюти це лише обмежені записи в базі даних, які ніхто не може змінити без виконання певних умов.

Як майнери створюють чи «добувають» монети і підтверджують операції у мережі? Давайте подивимося на механізм управління базами даних криптовалют. Криптовалюта, як Bitcoin, складається з мережі учасників. Кожен учасник має запис про всю історію всіх транзакцій і,

таким чином, баланс кожного рахунку. Власне у Bitcoin стан мережі і визначається балансом кожного користувача. Транзакція – це файл, який говорить: «Боб дає Алісі X Bitcoin» і підписаний приватним ключем Боба. Після підписання транзакція транслюється в мережі, відправляється з одного вузла до кожного іншого вузла. Це основна p2p-технологія[1].

1.1 Блокчейн у криптовалютах

Транзакція відома майже відразу всією мережею. Але тільки через певний проміжок часу це підтверджується. Підтвердження є критичним поняттям. Неформально можна визначити, що основним питанням криптовалют з точки зору реалізації є визначення алгоритму підтвердження транзацій[1].

Поки операція не підтверджена, вона очікується і може бути підробленою. Коли транзакція підтрверджується, вона додається до загального блоку транзакцій. Після цього, вона вже не може бути спростована, вона не може бути скасована, вона стає частиною незмінного запису історичних транзакцій: так званого блокчейна[1, 4].

Тільки майнери можуть підтверджувати операції. Це частина їхньої роботи в криптовалютній мережі. Вони беруть транзакції, помічають їх як істинні і поширюють у мережі. Після підтвердження транзакції майнером кожен вузол повинен додати його до своєї бази даних, щоб вона стала частиною загального ланцюжка блоків – блокчейна[1].

За цю роботу майнери отримують у нагороду токени криптовалюти. Оскільки діяльність майнерів є найважливішою частиною системи криптовалют, розглянемо її більш детально[1].

Важливим є те, що кожен може бути майнером. Оскільки децентралізована мережа не має повноважень для делегування цього завдання, криптовалюта потребує певного механізму, щоб запобігти

зловживанню однієї правлячої сторони. Уявіть, що хтось створює тисячі учасників і поширює підробні операції. Система почне перевантажуватись.

Тому Сатоші встановив правило, згідно з яким майнери повинні «інвестувати» певну роботу своїх комп'ютерів, щоб отримати право на це завдання. Фактично, вони повинні знайти хеш, який з'єднує новий блок зі своїм попередником. Це називається доказ виконання роботи - *Proof-of-Work*. У *Bitcoin*, він базується на алгоритмі *SHA-256 Hash*. Важливим є те, що обчислення цієї функції є своєрідною основою криптологічної головоломки, яку майнери намагаються обчислити. Знайшовши рішення, учасник може побудувати блок і додати його в блокчейн. Як стимул, він має право додати так звану транзакцію *coinbase*, що дає йому певну кількість біткоїнів. Це єдиний спосіб створення дійсних Bitcoin[1].

Біткоіни можуть бути створені лише тоді, коли майнери вирішують криптографічну головоломку. Оскільки складність цієї головоломки збільшує обсяг комп'ютерної потужності учасника, існує лише певна кількість криптовалютних токенів, які можуть бути створені за певний проміжок часу. Це є частиною консенсусу, який жоден з учасників мережі не може розірвати[1].

Один з найбільш поширених алгоритмів консенсусу включає доказ роботи (PoW). Чим більше майнер платить за обчислювальну техніку, необхідну для вирішення криптографічної головоломки, тим більше шансів, що він/вона отримує право на добування блоків. Проте, цей підхід заважає підвищенню енергоспоживання та необхідності дорогого добування апаратних пристроїв. Доказом частки (PoS) є інший алгоритм, який надає майнерам права на видобуток пропорційно до їх ставок у криптовалюті.

1.2 Proof of Burn

Доказ спалювання (*Proof-of-Burn*) є методом розподіленого консенсусу та альтернативою Доказу виконання роботи (*Proof-of-Work*) та Доказу володіння часткою (*Proof-of-Stake*). Він також може бути використаний для завантаження однієї криптовалюти з іншої[3].

Доказ спалювання є одним з декількох алгоритмів механізму консенсусу, реалізованих мережею блокчейн для забезпечення того, щоб всі учасники, що беруть участь, домовилися про справжній і дійсний стан мережі *blockchain*, тим самим уникнувши можливості подвійного витрачання криптомонет. *PoB* дотримуються принципу «спалювання» або «знищення» монет, що зберігаються майнерами.[3]

На ринку криптовалют протокол *Proof of Work* є найбільш часто використовуваним протоколом, який дозволяє користувачам майнити свою валюту. Він використовується цифровими валютами для досягнення згоди або, скоріше, децентралізованої угоди навколо певного блоку до блокчейна. З доказом роботи майнери конкурують один з одним для завершення транзакцій у мережі і отримують винагороду. Як вже зазначалось раніше, *PoW* використовує геш-функцію *Hashcash* (SHA-256) в якості поняття «Доказу роботи», яка використовується майнерами *Bitcoin* для вирішення складних математичних задач, що додають блоки у блокчейн.

У мережі користувачі відправляють один одному цифрові токени. Транзакції формують блоки. Ця відповідальність виконується окремими вузлами, відомими як «майнери», і процес називається «добуванням» або «майнингом». Доказ роботи є трудомістким і обчислювально дорогим, а процес дуже інтенсивний. У світі *Bitcoin*, *Proof of Work* – єдиний протокол, який буде використовуватися на основі поточної бази даних. Проте інші криптовалюти використовують комбінацію «Доказ виконання роботи» і «Доказ володіння часткою».

«Доказ володіння часткою» – це альтернатива для досягнення згоди або децентралізованого консенсусу.

У зв'язку з тим, що як показує час, занадто багато енергії, потужності та високої вартості вимагає протокол *Proof of Work*, було запропоновано використовувати *Proof of Stake*, оскільки майнери вважали, що видобуток одного блоку є марною витратою ресурсів. Цей протокол включає користувачів, які розміщують баланс гаманця криптовалюти. Не кожна криптовалюта підтримує Доказ Ставки, але вона поширена серед деяких із них. На відміну від Доказу роботи, Доказ Ставки є більш екологічним і забезпечує винагороду для власників.

У протоколі «Доказ ставки» має значення кількість монет у гаманці користувача. Користувачі, які володіють значним відсотком від загальної кількості монет, зароблятимуть більше ставок. Залежно від криптовалюти, що використовується, винагорода «Доказ ставки» може викликати інфляцію.

Proof of Burn відрізняється від *Proof of Work* та *Proof of Stake* в тому сенсі, що *PoB* стосується енергетичної проблеми протоколу *PoW*, а монети надсилаються адресу, що не підлягає перевірці, де монети спалюються, а вартість монет збільшується [14].

Ідея полягає в тому, щоб майнери показували докази того, що вони спалили деякі монети – тобто відправили їх на адресу, яка підлягає перевірці. Це дорого з їхньої особистої точки зору, так само, як доказ виконання роботи (PoW); але він не споживає жодних ресурсів, крім спаленого базового активу. До теперішнього часу всі докази криптовалют записують за допомогою видобутку криптовалют, які видобуваються на роботі, тому остаточним джерелом дефіциту залишається доказове «видобуток»[2].

Ключова ідея доведення спалювання полягає в тому, що при виборі об'єкту, який повинен кваліфікуватися як «складність», тобто вимагати від майнерів довести, що вони зробили «щось, що складно зробити», важливим є лише те, щоб окремі майнери вважали завдання дорогим. Крім того, інші

учасники повинні мати мегкий спосіб перевірки того, що майнер справді виконав складну роботу[2].

Доказ спалювання (*PoB*) – це альтернативний алгоритм консенсусу, який намагається вирішити проблему споживання енергії *PoW*. *PoB* часто називають *PoW* без витрат енергії. Він працює за принципом, що дозволяє майнерам «спалювати» або «знищувати» символи віртуальної валюти, що дає їм право писати блоки пропорційно монетам.

Ієн Стюарт (*Iain Stewart*), винахідник алгоритму *PoB*, наводить аналогію – згорілі монети – це добувні установки. По суті, майнер спалює свої монети, щоб купити віртуальну видобувну установку, яка надає йому/їй потужність для майнінгу блоків. Чим більше майнери спалюють монет, тим більша віртуальна «видобувна установка»[3].

Щоб спалити монету, майнери відправляють їх на перевірочну «невитратну» адресу (*unspendable address* або *eater address*). Цей процес не споживає багато інших ресурсів, крім спалених монет, і гарантує, що мережа залишається активною та гнучкою. Залежно від реалізації, майнерам дозволено спалювати національну валюту або валюту альтернативного ланцюга, наприклад Bitcoin. В обмін вони отримують винагороду в символічній копії блокчейна[14, 3]. Точні технічні деталі процесу спалювання, тобто відправлення на адресу, цього протоколу можуть варіюватись у залежності від криптовалюти та її реалізації. Наприклад, мережа *Slimcoin*, віртуальна валюта, яка використовує *PoB*, дозволяє майнерам спалювати монети, що не тільки дають йому/їй право конкурувати за наступний блок, але й дає йому/їй можливість отримувати блоки протягом більш тривалого періоду часу – не менше року. По суті, реалізація *PoB* у *Slimcoin* поєднує три алгоритми – *PoW*, *PoS* та основну концепцію *PoB*. Процес спалювання монет у цій криптовалюті включає в себе наступний принцип – чим більше монет спалюється, тим більше права матиме майнер на видобуток, що забезпечує *PoS* складова[3].

Перевірочна адреса не має приватного ключа, а це означає, що в той

час, коли хтось може переглядати загальну кількість монет і транзакцій за цією адресою, ніхто не може отримати до неї доступ, щоб розблокувати кошти. Важливо забезпечити такий вид прозорості, щоб учасники мережі могли перевірити, чи були фактично спалені монети [13]. Існує декілька способів реалізації доказів спалювання, які ми розглянемо в наступному розділі. Ось приклад такої адреси:

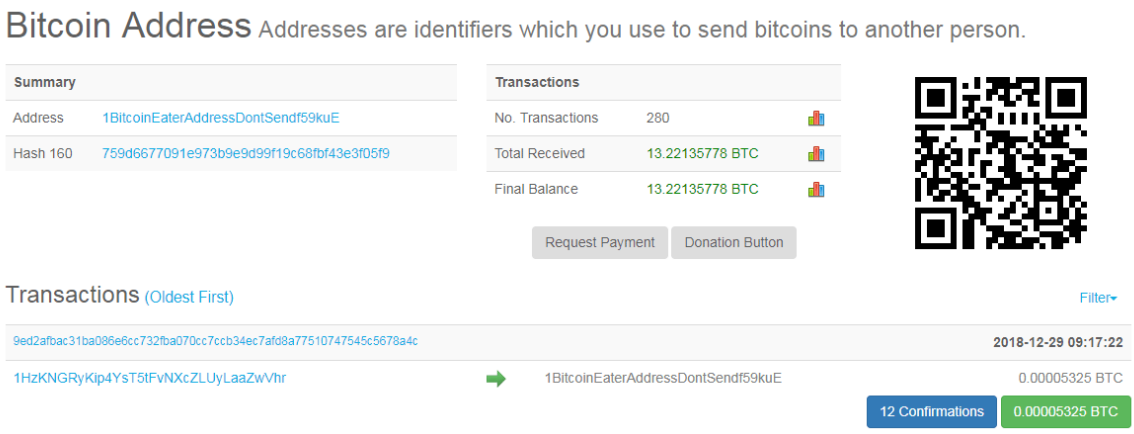

Рисунок 1.1 – Бітокін *Eater* адреса, на рахунку якої міститься близько 13 біткоїнів, які ніколи не зможуть бути відновлені[13].

1.3 Proof of Activity

Іддо Бентов (*Iddo Bentov*), Чарльз Лі (*Charles Lee*), Алекс Мізрахі (*Alex Mizrahi*) та Мені Розенфельд (*Meni Rosenfeld*) запропонували новий протокол для криптовалюти, який базується на протоколі Bitcoin за допомогою комбінування компонент «Доказ виконання роботи» (PoW) з системою «Підтвердження частки» (PoS). Протокол підтвердження активності (*Proof-of-activity* - PoA) пропонує хорошу безпеку проти можливих майбутніх атак на Bitcoin, і має відносно низьку штрафну санкцію з точки зору мережевого зв'язку та простору зберігання. Розробники дослідили різні сценарії можливих атак та запропонували

засоби захисту потенційних вразливостей протоколу *PoA*, а також оцінили продуктивність її основної підпрограми[8].

Протокол доведення активності є розширенням протоколу *Bitcoin*. Мережеві вузли в *PoA* повинні робити більш складні перевірки порівняно з роботою, яку виконують вузли мереж Bitcoin, і основна ідея полягає в тому, що ця додаткова робота приносить певні переваги.

Первинну підпрограму, яку об'єднує *PoA*, називають *follow-the-satoshi*, за допомогою чого деяка псевдовипадкова величина перетворюється в сатоші (*satoshi*)(найменшу одиницю криптовалюти), яка рівномірно підібрана між усіма сатошами, що були добуті до цих пір. Це робиться шляхом вибору псевдовипадкового індексу між нулем і загальною кількістю існуючих блоків, перевіряючи кожен із них, в якому був добутий цей сатоші, і після кожної транзакції, яка передала цю сатоші на наступну адресу, до досягнення адреси що в даний час контролює даний сатоші. Зауважимо, що цей процес можна розглядати як вибір псевдовипадкової зацікавленої сторони в єдиній формі, наприклад, якщо Аліса має 2 монети і Боб має 6 монет, то Аліса в 3 рази менше може бути вибраною в порівнянні із Бобом [8].

Далі наведемо алгоритм генерації блоків у мережі *PoA*.

1.3.1 Генерація блоків у протоколі PoA

1) Кожен майнер намагається згенерувати порожній заголовок блоку, тобто дані заголовка, що складаються з хешу попереднього блоку, загальнодоступної адреси майнера, індексу блоку в блокчейні. Цей заголовок не посилається на будь-які транзакції.

2) Коли майнер успішно генерує порожній заголовок блоку, що означає, що хеш дані заголовка блоку менші, ніж поточна «ціль» складності, він передає свій заголовок блоку в мережу.

3) Всі вузли мережі розглядають хеш цього заголовка блоку як

дані, які детерміновано виводять N псевдовипадкових зацікавлених сторін. Виведення здійснюється шляхом конкатенації цього хешу з хешем попереднього блоку і з N фіксованими значеннями суфіксів, потім хешування кожної комбінації, а потім викликом follow-the-satoshi з кожним з N хешей в якості вхідних даних.

4) Кожна із зацікавлених сторін, яка перебуває в режимі онлайн, перевіряє, чи є порожній заголовок блоку, який переданий майнером, правильним, що означає, що він містить хеш попереднього блоку і відповідає поточній складності. Після перевірки, зацікавлена сторона перевіряє, чи є вона однією з N щасливих учасників цього блоку. Перші N *- 1* щасливі учасники підписують хеш цього порожнього заголовка блоку приватним ключем, який керує їхніми сатоші, і транслюють свою сигнатуру в мережу. Коли N-ий учасник виявляє себе у блоці, він створює загорнутий блок, який розширює пустий заголовок блоку, включаючи стільки транзакцій, скільки він захоче включити, N *- 1* підписів інших зацікавлених сторін, а також свій власний підпис для загального хешу всього цього блоку.

5) N-ий учасник передає загорнутий блок в мережу, і коли інші вузли бачать, що цей обгорнутий блок є дійсним згідно з вищезазначеними пунктам, вони вважають його істинним продовженням та новою ланкою блокчейну. Вузли намагаються розширити найдовшу гілку блокчейна, про яку вони знають, де «найдовша» вимірюється у складності по вигляду, як у *Bitcoin*[8].

Слід відзначити, що якщо деякі з N вибраних зацікавлених сторін перебували в автономному режимі, то інші майнери також зможуть вирішить блок і тим самим виведять інші псевдовипадкові зацікавлені сторони, так що загальна складність буде відкоригована як за загальною потужністю, так і за часткою всіх зацікавлених сторонін у мережі.

Процес створення блоку у протоколі *PoA* здійснюється в двох раундах спілкування, на відміну від одного раунду комунікації у Bitcoin. Фактично, якщо ми встановимо $N = 1$ замість того, щоб підсилювати

владу зацікавлених сторін, вибираючи $N > 1$ переможців, *PoA* також вимагатиме лише один раунд зв'язку. У будь-якому випадку, процес створення блоку *PoA* значно більше задіяний, ніж процес створення блоку у *Bitcoin*[8].

Якщо деякі з вибраних валідаторів недоступні для завершення блоку, вибирається наступний виграшний блок, вибирається нова група валідаторів тощо, поки блок не отримає правильний обсяг підписів. Тарифи розподіляються між шахтарем і валідаторами, які підписалися на блоці[27].

Компонент *PoW* протоколу відіграє важливе значення для регулювання спроб при виборі зацікавлених сторін, оскільки в іншому випадку виникнення відбуватиметься з неймовірною швидкістю. Одна з основних проблем полягає в тому, що без компонента *PoW* не існує чіткого способу зробити систему конвергентною, тобто мати таке правило протоколу, яке визначає вагу дійсної гілки таким чином, щоб зберегти децентралізацію і змусити мережу сходитися до переможної гілки без запровадження можливих векторів атаки. Інша проблема полягає в тому, що раціональні зацікавлені сторони можуть максимізувати свою очікувану винагороду, підписуючи кожну гілку, яку вони бачать, якщо *PoW* не вимагає розширення гілки [4].

Протокол може дозволити N псевдовипадковим зацікавленим сторонам перемістити свої монети на нову *P2SH* адресу [5] замість відправки лише допоміжного підпису, щоб забезпечити 160-бітний захист від атаки другого прообразу. Клієнт зацікавленої сторони може вирішити додати додаткову 160-бітну адресу лише в тому випадку, якщо кількість монет на виході перевищує певне порогове значення. При цьому кожен блок буде містити до $\approx N \cdot 160$ додаткових бітів.

Відзначимо, що існує схожість між рівномірним вибором псевдовипадкової зацікавленої сторони та наданням права голосу групам зацікавлених сторін. Наприклад, якщо існують дві групи зацікавлених сторін, одна чесна група, що контролює 80% акцій, та зловмисна група,

що контролює 20% акцій, то вибір псевдовипадкової зацікавленої сторони рівномірно передбачає, що зловмисна група вибирається 20% часу, і тому вона фактично має 20% голосів. У порівнянні з іншими схемами доведення частки, де багато зацікавлених сторони будуть використовувати свої права голосу, щоб закріпити свої підписи до певних блоків контрольних точок, щоб зміцнити блокчейн, протокол *PoA* має значно менше накладних витрат з погляду мережевої комунікації та розширенням блокчейна[8].

Слово «активність» у назві протоколу «доказ активності» підкреслює той факт, що тільки активні зацікавлені сторони, які підтримують повний онлайн-вузол, отримують винагороду в обмін на життєво важливі послуги, які вони надають для мережі. У цьому й полягає осоновна відмінність від протоколу *Proof of Stake*, у яких користувачі, що знаходяться в автономному режимі, можуть накопичувати вагу з плином часу і в кінцевому підсумку можуть бути використані в атаках з подвійним витрачанням коштів (англ. *double-spending attacks*)[8].

Раніше припускалось, що транзакції є стандартними, тобто що кожен вихід підписується одним приватним ключем, який відповідає вказаній адресі, тому зацікавлена сторона просто надає додатковий підпис, який доводить, що вона може витратити виграшний сатоші. Протокол Bitcoin включає в себе мову сценаріїв, що дозволяє формувати більш складні нестандартні транзакції, які можуть бути проведені за допомогою декількох підписів [6] або взагалі без підписів [7]. Отже, існує протиріччя між *Proof of Stake* та нестандартними сценаріями, оскільки зацікавлена сторона, яка може викупити нестандартний результат, може не надати підпис, який доводить, що вона контролює цей результат. На щастя, багато випадків використання нестандартних транзакцій передбачають швидкі операції, а не довгострокові. Розробники протоколу запропонували, щоб нестандартні сценарії мали жорстку форму, щоб зацікавлені сторони мали право на лотерею, тобто підлягали погашенню

шляхом підписання приватним ключем sk_0, що відповідає адресі A_0. Нестандартні сценарії, сумісні з цією формою, можуть бути корисними, наприклад, (A_1 or (A_2 and A_3)), де sk_1 керується захищеним апаратним гаманцем, sk_2 знаходиться на смартфоні, а sk_3 – на ноутбуці[8].

Будь-який конкретний вибір N для числа зацікавлених сторін може бути підданий критиці як «магічне число» [8], що подібне 10-хвилинному часу генерування блоку у протоколі Bitcoin. Вибір на користь більшого числа означає, що влада зацікавлених сторін по відношенню до майнера посилюється, хоча це також передбачає підвищену складність зв'язку і роздування даних. Питання про те, чи може це число бути динамічним, а не постійним, є законним, хоча далеко не ясно, чи є динамічний варіант вигідним в цілому. Наявність динамічного числа похідних зацікавлених сторін може бути можливо шляхом використання аналогічних ідей, тобто протокол може визначити рівень участі зацікавлених сторін і вказати, що число похідних зацікавлених сторін зменшується, коли рівень участі занадто низький, і навпаки.Недоліками динамічного варіанту є більш складні правила протоколу, і, що більш важливо, загрози безпеці, які виникають, коли зловмисник зможе скористатися цими правилами протоколу, маючи менше необхідності використовувати частку в гілці, яка генерується в таємниці. Аналогічно, заміна постійної 10-хвилинної цілі часу знаходження блоку на динамічний час також може призвести до уразливості криптовалюти [9, 10, 11]. В цілому, розглядається константа $N = 3$ для числа похідних зацікавлених сторін[8].

Протокол *PoA* винагороджує зацікавлені сторони, які беруть участь і підтримують мережу, а не карає зацікавлені сторони, які не беруть участі. Можливо, існують потенційні переваги протоколу *Proof of Stake*, що карає пасивних незацікавлених учасників, а саме – більше заохочення зацікавлених сторін спонукається до участі в акції та більш справедливого розподілу винагороди. Ця остання турбота натякає на ефект «багаті стають багатшими», тобто сторони, у яких багато монет, буде легше отримати ще більше монет. Подібність *PoA* із протоколом

PoW існує у тому, що (як згадувалося вище) очікувана винагорода, яка є результатом наступної сатоші, є лінійною у відношенні до частки, а очікувана винагорода від видобутку *PoW* є лінійною в апаратних витратах, тобто, наприклад, очікується, що майнер, який керує двома ідентичними машинами, заробить удвічі більше, ніж з однією з машин. При цьому протокол *PoA* може бути доповненим механізмом покарання шляхом отримання додаткових псевдовипадкових зацікавлених сторін, які повинні надати свої підписи в одному з декількох наступних блоків, або ж бути покарані конфіскацією монет[8].

Як і у випадку з мережею Bitcoin, вузли в Мережі *PoA* можуть мати небажання повторно передавати дані для блоків, які створюються. Щоб максимізувати очікувану винагороду, зацікавлені сторони можуть утриматися від повторної передачі транзакцій (якщо $N = 1$), а майнери і зацікавлені сторони можуть утриматися від повторної передачі порожніх заголовків блоків *PoW*. Коли кількість онлайн-вузлів у результаті пулів невелика, це занепокоєння є досить м'яким, оскільки користувачі криптовалюти можуть передавати дані безпосередньо в пули і, отже, гарантувати швидкі підтвердження своїх транзакцій. Однак, однією з цілей *PoA* є наявність багатьох зацікавлених сторін. Коли кількість вузлів зацікавлених сторін в мережі велика, кожен такий вузол отримує винагороду нечасто, і тому очікує, що гранична втрата винагороди при повторній передачі даних буде невеликою. Таким чином, дані будуть передаватися альтруїстичними зацікавленими сторонами, які дотримуються протоколу, і деякими раціональними зацікавленими сторонами відповідно до того, як вони оцінюють невелику очікувану втрату винагороди по відношенню до вартості своєї частки, як це диктується загальним станом мережі. Крім того, протокол *PoA* може бути доповнений методами, які заохочують поширення даних [12], шляхом винагородження вузлів, які повторно передають транзакцію з частиною плати за транзакцію.

Причина конкатенації хешу попереднього обгорненого блоку, коли

отримується N зацікавлених сторін, замість того, щоб використовувати тільки хеш поточного порожнього блоку заголовка, полягає в тому, що ентропія хешу заголовка блоку зменшується по мірі підвищення рівня складності *PoW*. Крім того, рівень складності *PoA* нижчий, ніж у біткоїна[8].

Протокол *PoA* прагне децентралізувати потужність, яка синхронізує транзакції досить вираженим способом. Щоб монополізувати процес створення блоків, зловмисник повинен контролювати значну частину загальної кількості монет, які були створені до цих пір. Стверджується, що в ймовірних сценаріях вартість атаки буде набагато вище з протоколом *PoA*, ніж з протоколом *PoW* біткоїна. Крім того, протокол *PoA* виконує інші корисні властивості, а саме поліпшену топологію мережі, стимули для підтримки повних мережних онлайн-вузлів, низькі транзакційні збори та більш ефективне використання енергії[8].

Slimcoin, альтернативна криптовалюта, заснована на *Peercoin*, що використовує *Proof of Burn* як частину свого алгоритму консенсусу та альтернативного методу видобутку блоків[24].

Учасники з повними вузлами *Slimcoin* (*SLM*) можуть заробити монети, знаходячи блоки. Імовірність того, що учасник (ідентифікований за його адресою або відкритим ключем) знайде блок, визначається оцінкою, яка називається ефективними спаленими монетами (англ. *Effective Burnt Coins*) на основі кількості спалених монет з його адреси. Спалені монети розпадаються з плином часу: оцінка ефективних спалених монет зменшується на кожному блоці *Proo of Work*, опускаючись до нуля через кілька років[24].

На відміну від криптовалюти *Slimcoin*, *Counterparty* (*XCP*) використовує *Proof of Burn* для видачі своїх токенів. Учасники відправляють біткоїни на перевірочну «невитратну» адресу *Bitcoin* і отримують у результаті токени криптовалюти[25].

1.4 Висновки

У даному розділі наукової роботи було проаналізовано основні поняття та складові елементи криптовалютних систем. Зокрема, аналіз було зосередженно на протоколах узгодження та двох з найпоширеніших протоколів, що активно використовуються сьогодні у криптовалютах – *Proof of Burn* та *Proof of Activity*. Крім того, у ході дослідження було розглянуто протоколи, що служать основою для вищенаведених алгоритмів. Надалі у роботі буде детально проаналізовано математичну складову цих систем та обчислено оцінки успіху користувачів у залежності від протоколу та ефективність кожного із них. Під ефективністю буде розглядатись ймовірність генерації наступного блоку випадковим користувачем.

2 ОЦІНКИ СИСТЕМ PROOF OF BURN ТА PROOF OF ACTIVITY

Перш ніж приступати до оцінювання даних протоколів, спершу необхідно описати їхні математичні моделі.

2.1 Системи масового обслуговування та їхні параметри

Теорія систем масового обслуговування є важливим розділом економіко-математичного моделювання і являє собою теоретичні основи ефективного конструювання і експлуатації систем масового обслуговування. Системи масового обслуговування (англ. *queuing system*) зустрічаються в багатьох галузях економіки і призначені для багаторазового використання при виконанні подібних завдань[15]. У багатьох областях фінансів, економіки, виробництва та побуту важливу роль відіграють системи спеціального виду, що реалізують багаторазове виконання однотипних завдань. Такі системи як комп'ютерні мережі, системи збору, зберігання і обробки інформації, транспортні системи, різні військові системи, зокрема системи протиповітряної або протиракетної оборони, також можуть розглядатися як своєрідні системи масового обслуговування.

СМО не є чимось необхідним один або два рази. Щоб ці моделі приносили результати, вони повинні бути представленні безперервним процесом. Протягом своєї історії, СМО пройшли довгий шлях. Вони перейшли від простих фізичних бар'єрів до найсучасніших цифрових додатків. Існує багато видів рішень для систем масового обслуговування, що вибираються, але їх найпростіші випадки є найменш ефективними на практиці[20].

Як бачимо англійською мовою теорія масового обслуговування звучить як «*Queuing theory*», що у дослівному перекладі означає «теорія черг». Справді, теорія масового обслуговування в значній мірі присвячена вивченню черг, що виникають в різних системах. Наприклад – черга в банку. При занадто довгому очікуванні в черзі до каси чи менеджера відділення або при відмові в обслуговуванні клієнт залишиться незадоволеним, і малоймовірно, що він повернеться сюди знову. А використання теорії масового обслуговування дозволяє обчислити середню інтенсивність клієнтів в різний час робочого дня і виявити оптимальну пропускну здатність системи, що дозволить уникнути різних проблем[15].

Проте СМО не завжди може бути представленою чергою. Деколи модель системи обслуговування представляється тільки надходженням клієнтів. У таких випадках вони називаються моделями чистого народження, або ж тільки виходом клієнтів з системи – моделлю чистої загибелі. Прикладом першої моделі є процес оформлення свідоцтва про народження дітей, прикладом другої моделі може бути вилучення запасів, що зберігаються на складі. Обидві ці моделі мають досить складні формули, які тягнуть за собою громіздкі обчислення. Тому для оптимізації розрахунків доцільно застосовувати програми, що спрощують і автоматизують дані обчислення[15].

Кожна СМО включає в себе певну кількість обслуговуючих пристроїв, які називають каналами обслуговування. Роль каналів можуть відігравати різні прилади або особи, які виконують ті чи інші операції.

Розглянемо основні поняття систем масового обслуговування[15].

– Заявка (вимога) – запит на обслуговування.

– Вхідний потік заяв – сукупність вимог, що надходять у СМО.

– Черга.

– Час обслуговування – період часу, протягом якого обслуговується вимога.

– Канал або канали обслуговування.

– Математична модель СМО – це сукупність математичних виразів, що описують вхідний потік вимог, процес обслуговування та їх взаємозв'язок.

– Вихідний потік обслужених заявок.

Система масового обслуговування призначена для виконання деякого потоку заявок, що надходять на вхід системи здебільшого не регулярно, а в випадкові моменти часу. Крім того процес обслуговування заявок також триває не протягом постійного, заздалегідь відомого, а випадкового проміжку часу, який залижеться від багатьох факторів. Досить часто ці фактори можуть бути невідомими системі. Враховуючи те, що характер потоку заявок та час їхнього обслуговування несе випадковий характер, це призводить до нерівномірного навантаження системи масового обслуговування. Як наслідок, можуть виникати проміжки часу у які на вході системи можуть накопичуватись необслужені заявки, що призводить до перенавантаження СМО. З іншої сторони можливі абсолютно протилежні стани коли на вході немає жодної заявки. Це у свою чергу веде до простоювання каналів. Закон, який визначає порядок обслуговування вхідних заявок, називається дисципліною черги[15].

Як бачимо, кожна система масового обслуговування має свої параметри:

1) характер потоку заявок;

2) число каналів обслуговування;

3) продуктивність каналів;

4) правила організації роботи.

І в залежності від цих параметрів визначається пропускна здатність (ефективність функціонування) СМО, що дозволяє їй успішно справлятися з потоком заявок. У випадку виникнення помилок чи проблем зі обробленням заявок при навантаженні системи, слід аналізувати існуючу систему, правила обробки заявок та пошуку кращого оптимальнішого рішення.

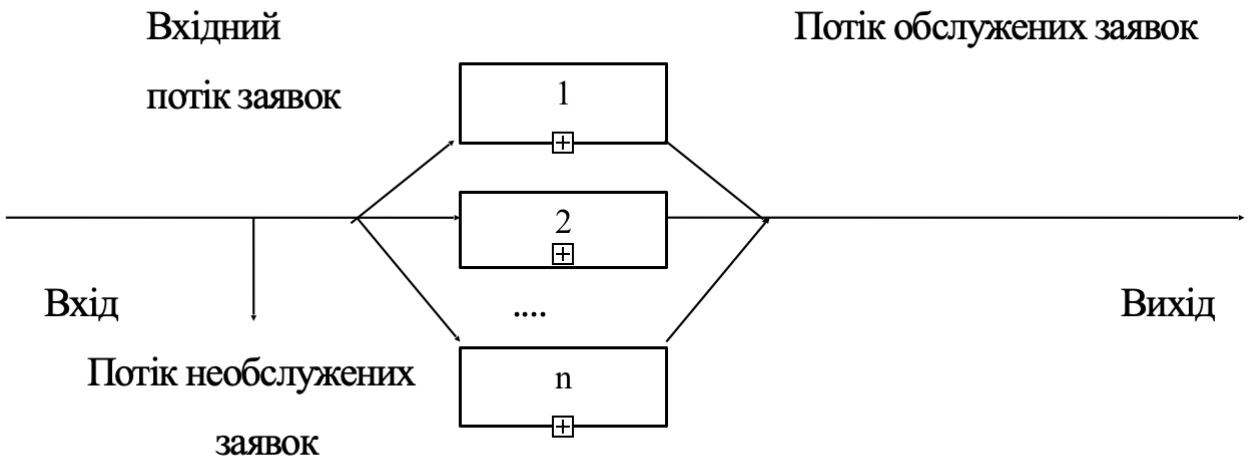

Рисунок 2.1 – Схема систем масового обслуговування у загальному вигляді[15]

СМО є предметом теорії масового обслуговування. Мета теорії масового обслуговування – вироблення рекомендацій щодо раціональної побудови СМО, раціональної організації їх роботи та регулювання потоку заявок для забезпечення високої ефективності роботи СМО. Для досягнення цієї мети ставляться завдання теорії масового обслуговування, що складаються у встановленні залежностей ефективності функціонування СМО від її параметрів.

Головними елементами моделі масового обслуговування є клієнт (заявка на обслуговування) і сервіс (обслуговуючий пристрій). Клієнти, вступивши до сервісу, можуть відразу ж потрапити на обслуговування або стати в чергу, якщо сервіс зайнятий.

Неменш важливим фактором при аналізі та побудові системи масового обслуговування є вибір принципу побудови черги заявок, згідно з яким обиратимуться клієнти з черги для подальшого обслуговування. Найвідомішими правилами вибору клієнтів із черги є «першим прийшов – першим обслужився» та «першим прийшов – останнім обслужився». Перше правило зазвичай позначається абрівіатурою *FIFO* (англ. *First-In-First-Out*). Друге – *LIFO* від англ. *Last-In-First-Out*. Також

можливі модифікації наведених правил. Наприклад, клієнти можуть бути обрані з черги згідно заданого пріоритету. Прикладом може служити виробничий цех, у якому термінові роботи виконуються раніше звичайних. Або приклад із обслуговування клієнтів у банку. Власники «ґолд» карти обслуговуються першими.

Також важливим фактором при аналізі систем з чергами є поведінка клієнта, що потребує обслуговування. Клієнти при наявності паралельного обслуговування можуть перейти з однієї черги до іншої метою скоротити тривалість очікування, або ж залишити чергу, простоявши в ній якийсь час і дійшовши до висновку, що і так вже занадто багато часу втрачено. Саме тому відстеження ефективності роботи системи масового обслуговування є дуже важливим та, навіть, критичним. У залежності від поточної ситуації у системі та на основі прогнозування подальшої кількості появи клієнтів у черзі, слід приймати рішення про збільшуення або зменшення кількості сервісів в залежності від інтенсивності надходження замовлень або збільшення працездатності сервісів при необхідності.

У загальному випадку в якості показників ефективності функціонування системи масового обслуговування можна обираються три основні групи показників:

1) Показники ефективності використання СМО:

– Абсолютна пропускна спроможність СМО – середнє число заявок, яке зможе обслужити система обслуговування за одиницю часу.

– Відносна пропускна спроможність СМО – відношення середнього числа заявок, що обслуговуються СМО в одиницю часу, до середнього числа заявок, що надійшли за цей момент часу.

– Середня тривалість періоду зайнятості СМО.

– Коефіцієнт використання СМО – середня частка часу, протягом якого система обслуговування зайнята обслуговуванням клієнтів.

2) Показники якості обслуговування заявок:

– Середній час очікування заявки у черзі.

– Середній час перебування заявки в системі масового обслуговування.

– Імовірність відмови заявці в обслуговуванні без очікування.

– Ймовірність того, що заявка, що надійшла, буде негайно прийнята до обслуговування.

– Закон розподілу часу очікування заявки в черзі.

– Закон розподілу часу перебування заявки в СМО.

– Середнє число заявок, що знаходяться в черзі.

– Середнє число заявок, що знаходяться в СМО.

3) Показники ефективності функціонування пари «СМО - споживач», де під «споживачем» розглядається сукупність певних заявок або деяке їхнє джерело.

Остання група показників є найбільш ефективною та корисною при аналізі та оптимізації системи обслуговування у тих випадках, коли дохід, який отримується в наслідок обслуговування заявок, і витрати на це обслуговування вимірюються в одинх і тих же одиницях. Як правило, ці показники носять конкретний характер і визначаються специфікою обслуговування заявок СМО.

За дисциплінами обслуговування СМО поділяють на три класи:

1) СМО з відмовами (нульовим очікуванням). У таких системах заявка, що надійшла на вхід СМО у момент, коли всі канали були зайняті, отримує «відмову» і залишає систему обслуговування. Щоб ця заявка все ж таки була обслуженою, вона повинна знову поступити на вхід цієї СМО та розглядатися при цьому як нова заявка, тобто така, що надійшла вперше. Прикладом СМО з відмовами може служити робота автоматична телефонна станція (АТС). Згідно її роботи, якщо набраний телефонний номер зайнятий, то заявка отримує відмову. Тому для того щоб додзвонитися за цим номером, необхідно його набрати ще раз. У такому випадку повторна заявка повинна надходити на вхід як нова.

2) СМО з очікуванням. У таких системах заявка, що надійшла в момент зайнятості всіх каналів, стає в чергу і чекає звільнення каналу,

який прийме її до обслуговування. Кожна заявка, що надійшла на вхід, в решті решт буде обслужена. Такі системи обслуговування часто зустрічаються в сфері торгівлі, побутового і медичного обслуговування, а також на підприємствах. Прикладом може служити черга в перукарню.

3) СМО змішаного типу (обмеженим очікуванням). Це такі системи, в яких на перебування заявки в черзі накладаються деякі обмеження, в більшості випадків обмеження накладаються на довжину черги, тобто максимально можливе число заявок, які одночасно можуть перебувати в черзі. Як приклад такої системи можна привести майстерню по ремонту автомобілів, що має обмежену за розмірами стоянку для несправних машин, які очікують ремонту.

2.2 Прокотол Proof of Burn як система масового обслуговування

Опишемо математичну модель протоколу *Proof of Burn* з точки зору системи масового обслуговування та оцінимо ймовірність успіху користувача, тобто участі у роздачі винагороди, у залежності від кількості спалених монет цим користувачем.

2.2.1 Прокотол Proof of Burn – модель народження-гибелі

Прокотол *Proof of Burn* у свою чергу описується дещо складнішою моделюю народження-смерті або народження-гибелі.

Описується дана модель наступним чином: система перебуває в заданому стані і для випадкового часу, що слідує за експоненціальним розподілом з параметром $\lambda_i + \mu_i$. При виході зі стану i процес переходить у стан $(i + 1)$ або стан $(i - 1)$. Рух аналогічний руху випадкового ходу, за

винятком того, що переходи відбуваються у випадкові моменти часу, а не у фіксовані періоди часу[19].

Модель народження-гибелі є важливим підкласом ланцюгів Маркова з безперервним часовим параметром. Ці процеси характеризуються властивістю, що при перехід відбувається до сусіднього стану.

Зміни у моделі чистого народження відбуваються наступним чином: $E_{n-1} \to E_n \to E_{n+1} \to ....$ Модель процесу народження-гибелі дозволяє перехід не лише станів $E_n \to E_{n+1}$, а й $E_n \to E_{n-1}$ при $n > 0$. При $n = 0$ дозволено лише перехід $E_0 \to E_1$.

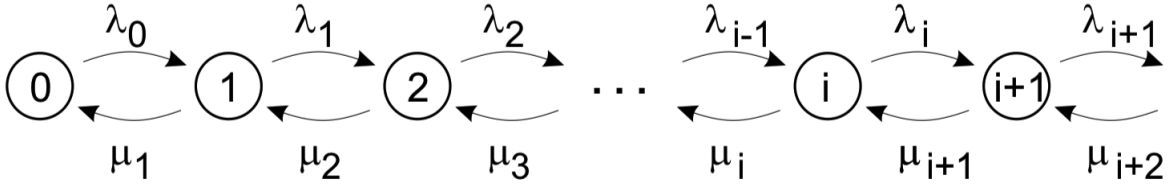

Рисунок 2.2 – Схема моделі народження-гибелі[17]

Описується така модель наступним чином. У момент часу t система знаходиться у стані E_n. Тоді протягом відрізку часу $(t, t + h)$:

– Транзакція $E_n \to E_{n+1}$ відбудеться з ймовірністю $\lambda nh + o(h)$.

– Транзакція $E_n \to E_{n-1}$ відбудеться з ймовірністю $\mu nh + o(h)$.

– Ймовірність, що відбудеться хоча б один перехід дорівнює $o(h)$.

Тоді

$$P_n(t + h) = P_n(t)(i - \lambda_n h - \mu_n h) + P_{n-1}(t)(\lambda_{n-1} h) + P_{n+1}(t)(\mu_{n+1} h) + o(h).$$

Час еволюції ймовірностей:

$$P'_n(t) = -(\lambda_n + \mu_n)P_n(t) + \lambda_{n-1} P_{n-1}(t) + \mu_{n+1} P_{n+1}(t).$$

При $n = 0$ закон описується наступним чином:

$$P_0(t+h) = P_0(t)(1-\lambda_0 h) + P_1(t)\mu_1 h o(h) \Rightarrow P_0'(t) = -\lambda_0 P_0(t) + \mu_1 P_1(t).$$

При $\lambda_0 = 0$ перехід $E_0 \to E_1$ є неможливим та стан E_0 є поглинаючим.

Також при $\lambda_0 = 0, P_0(t) = \mu_1 P_1(t) \geq 0$, і, отже, $P_0(t)$ монотонно зростає.

Розглянемо стаціонарний розподіл цієї моделі. При $t \to \infty, P_n(t) \to P_n(\infty)$. З цього слідує, що $P_0'(t) \to 0$ та $P_n'(t) \to 0$. Тому

$$0 = -\lambda_0 P_0 + \mu_1 P_1 \Rightarrow P_1 = \frac{\lambda_0}{\mu_1} P_0.$$

$$0 = -(\lambda_1 + \mu_1) P_1 + \lambda_0 P_0 + \mu_2 P_2 \Rightarrow P_2 = \frac{\lambda_0 \lambda_1}{\mu_1 \mu_2} P_0.$$

Аналогічно $P_3 = \frac{\lambda_0 \lambda_1 \lambda_2}{\mu_1 \mu_2 \mu_3} P_0$ і т. д.

Як бачимо, залежність від початкових умов зникає.

Після нормалізації, тобто $\sum\limits_{i=1}^{\infty} P_n = 1$, отримуємо

$$P_0 = \frac{1}{1 + \sum\limits_{n=1}^{\infty} \prod\limits_{i=0}^{n-1} \frac{\lambda_i}{\mu_{i+1}}}, P_n = \frac{\prod\limits_{i=0}^{n-1} \frac{\lambda_i}{\mu_{i+1}}}{1 + \sum\limits_{n=1}^{\infty} \prod\limits_{i=0}^{n-1} \frac{\lambda_i}{\mu_{i+1}}}, \ n \geq 1.$$

Стан ергодичності описується тим, що $P_n > 0$ для усіх $n \geq 0$. З цього випливає $\sum\limits_{n=1}^{\infty} \prod\limits_{i=0}^{n-1} \frac{\lambda_i}{\mu_{i+1}} < \infty$[18].

Лінійна модель народження-гибелі визначається наступними умовами[16]:

– $\lambda_n = n\lambda$.

– $\mu n = n\mu$.

$$\Rightarrow P_0'(t) = \mu P_1(t).$$

$$P_n'(t) = -(\lambda+\mu)nP_n(t) + \lambda(n-1)P_{n-1}(t) + \mu(n+1)P_{n+1}(t).$$

Стаціонарна поведінка характеризується:

$$\lim_{t\to\infty} P_0'(t) = 0 \Rightarrow P_1(\infty) = 0.$$

Аналогічно при $t \to \infty, P_n'(\infty) = 0$.

Можливі два випадки:

1) $P_0(\infty) = 1$. При цій умові ймовірність остаточного зникнення дорівнює 1.

2) Якщо $P_0(\infty) = P_0 < 1$, то відношення $P_1 = P_2 = P_3 = ... = 0$ реалізуються з ймовірністю $1 - P_0$. Тобто, популяція моделі може зростати безмежно.

Отже, популяція лінійної моделі народження-гибелі повинне або вимирати, або збільшуватися протягом невизначеного часу.

За означенням, $M(t) = \sum\limits_{n=1}^{\infty} nP_n(t)$.

Розглянемо $M'(t) = \sum\limits_{n=1}^{\infty} nP_n'(t)$. Тоді

$$M'(t) = -(\lambda+\mu)\sum_{n=1}^{\infty} n^2P_n(t) + \lambda\sum_{n=1}^{\infty}(n-1)nP_{n-1}(t) + \mu\sum_{n=1}^{\infty}(n+1)nP_{n+1}(t).$$

Перепишемо $(n-1)n = n^2 - n = n^2 - 2n + n + 1 - 1 = (n^2 - 2n + 1) + + (n-1) = (n-1)^2 + (n-1)$.

Аналогічно отримуємо $(n+1)n = (n+1)^2 - (n+1)$.

Підставимо ці перетворення у $M'(t)$:

$$M'(t) = -(\lambda+\mu)\sum_{n=1}^{\infty} n^2 P_n(t) + \lambda\sum_{n=1}^{\infty}(n-1)^2 P_{n-1}(t) +$$
$$+\mu[\sum_{n=1}^{\infty}(n+1)^2 P_{n+1}(t) + P_1(t)] + \lambda\sum_{n=1}^{\infty}(n-1)P_{n-1}(t) -$$
$$-\mu[\sum_{n=1}^{\infty}(n+1)P_{n+1}(t) + P_1(t)].$$

$$\Rightarrow M'_t = \lambda\sum_{n=1}^{\infty} nP_n(t) - \mu\sum_{n=1}^{\infty} nP_n(t) = (\lambda-\mu)\sum_{n=1}^{\infty} nP_n(t) = (\lambda-\mu)M(t).$$

Якщо $P_{n_0}(0) = 1$, то $M(t) = n_0 e^{(\lambda-\mu)t}$.

Згідно значення $\lambda - \mu$ можливі 2 випадки:

1) $\lambda - \mu > 0$, тобто $\lambda > \mu$. Отримуємо $M(t) \to \infty$.

2) Якщо ж $\lambda < \mu$, то $M(t) \to 0$.

Відповідно до виведення, що наведено вище, якщо $M_2(t) = \sum\limits_{n=1}^{\infty} n^2 P_n(t)$, можна показати, що

$$M'_2(t) = 2(\lambda-\mu)M_2(t) + (\lambda+\mu)M(t).$$

Коли $\lambda > \mu$, дисперсія процесу дорівнює

$$n_0 e^{2(\lambda-\mu)t}(1 - e^{(\mu-\lambda)t})\frac{\lambda+\mu}{\lambda-\mu}.$$

2.2.2 Оцінка ймовірності участі користувача у розіграші винагороди у залежності від кількості спалених монет

Алгоритм *Proof of Stake* є узагальненням алгоритму *Proof of Work*. У протоколі *Proof of Stake* вузли відомі як «валідатори», і замість видобутку блокчейна вони перевіряють транзакції, щоб заробити плату за

транзакцію. Не існує поняття майнингу як такого, оскільки всі монети існують з першого дня існування мережі. Простіше кажучи, вузли вибираються випадковим чином для перевірки блоків, і ймовірність цього випадкового вибору залежить від кількості проведеної ставки. Отже, наприклад, якщо вузол X володіє 2 монетами і вузол Y володіє 1 монетою, то вузол X вдвічі частіше буде викликаний для перевірки блоку транзакцій. Конкретна реалізація *Proof of Stake* може змінюватися залежно від випадку використання. Такими прикладами є *Proof of Burn* та *Proof of Deposit*. Алгоритм *PoS* заощаджує дорогі обчислювальні ресурси, які витрачаються на майнинг у режимі консенсусу[26].

Протокол *Proof of Work* як відомо відображає наступне твердження – чим більша обчислювальна потужність у користувача, тобто чим більше він може зробити математичних обчислень, тим вищий пріоритет у розподілі винагороди при добуванні наступного блоку. В теорії, якщо користувач буде володіти необмеженою обчислювальною потужністю, то крім нього ніхто «вигравати» не буде. Але як правило на практиці вводяться обмеження. Тобто кожному користувачеві виділяється певна підмножина вхідних значень для повного перебору.

Складова *Proof of Stake* полягає у своєрідній грі у рулетку, тобто відбувається процес ставок. Іншими словами, чим більше було потрачено монет на різні значення, тим більша ймовірність виграти.

У протоколі *Proof of Burn* розробники зробили акцент на дещо іншу стратегію. Вона полягає у наступному: кількість спалених монет визначає своєрідний рейтинг користувача, який відповідно можна залишати незмінним, або ж підвищувати спалюючи монети.

Також слід відзначити те, як саме здійснюються переходи у протоколі *Proof of Burn*. Як вже розглядалось раніше, цей алгоритм описується моделлю народження-гибелі, у якій можливий рух у обидві сторони, що суперечить природі блокчейну. Проте рух здійснюється не по блокчейні, а власне по станах мережі, які записуються у блоки транзакцій. Коли здійснюється перехід від стану E_n до стану E_{n-1}, то цей

перехід відбувається не як перехід до попереднього блоку у ланцюжку блокчейн, а відбувається перехід до стану системи з кількістю монет, що менша у деяких користувачів ніж була у попередньому стані системи.

Щодо розподілу ймовірностей, то схема досить проста. Як вже раніше зазначалось, чим більше користувач спалив монет, тим більша ймовірність того, що він отримає винагороду при її розподілі. Таким чином, ймовірність «виграшу» визначається біноміальним законом розподілу. Тобто ймовірність числюється наступним чином:

$$P(k) = \binom{k}{n} p^k q^{n-k} = \binom{k}{n} p^k (1-p)^{n-k}, k = 0, 1, 2...,$$

де $\binom{k}{n} = \frac{n!}{k!(n-k)!}$ – біноміальний коефіцієнт.

У даному протоколі число n дорівнює середній кількості спалених монет у мережі, а числу k відповідає кількості спалених монет конкретним користувачем.

2.3 Прокотол Proof of Activity як система масового обслуговування

Розглянемо математичну модель протоколу *Proof of Activity* та оцінимо шанси користувача поучастувати у підтвердженні наступного блоку блокчейна у залежності від проведеного часу у мережі, тобто від тривалості його сесії.

2.3.1 Прокотол Proof of Activity – модель чистого народження

Модель обслуговування системи, представлена тільки надходженням клієнтів називається моделлю чистого народження. Хоча слід зазначити, що модель народження є частковим випадком моделі народження-смерті.

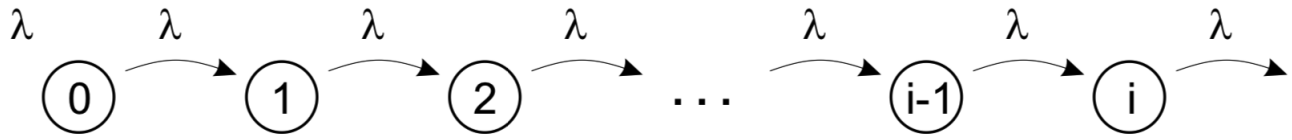

Рисунок 2.3 – Схема процесу Пуассона та моделі чистого народження систем масового обслуговування[17]

Нехай $p_0(t)$ – ймовірність відсутності подій (надходження клієнтів) за період часу t. За умови, що довжина інтервалу часу T між надходженнями клієнтів описується експоненціальним розподілом з інтенсивністю λ, будемо мати[20]:

$p_0(t) = P\{\text{інтервал часу } T \geq t\} = 1 - P\{\text{інтервал часу } T \leq t\} =$
$= 1 - (1 - e^{-\lambda t}) = e^{-\lambda t}$.

При досить малому інтервалі часу $t > 0$ отримуємо:

$$p_0(h) = e^{-\lambda h} = 1 - \lambda h + 2\frac{\lambda)}{2!} - ... = 1 - \lambda h + O(h^2)$$

Експоненційний розподіл базується на припущенні, що на досить малому часовому інтервалі $h > 0$ може наступити не більше однієї події, тобто надходження клієнта. Отже, при $h \to 0,\ p_1 = 1 - p_0(h) \approx \lambda h$[21].

Даний результат показує, що ймовірність надходження клієнта протягом інтервалу h прямо пропорційна h з коефіцієнтом пропорційності, що дорівнює значенню інтенсивності надходжень λ.

Процес Пуассона також можна визначити як процес з незалежними приростами і з однаковим експоненціальним розподілом часу між виникненням подій[23].

Для того, щоб отримати розподіл кількості клієнтів, що надійшли протягом деякого відрізку часу, позначимо ймовірність надходження n клієнтів протягом часу t через $p_n(t)$. При досить малому $h > 0$ маємо наступне:

$$p_n(t+h) = p_n(t-\lambda h) + p_{n-1}(t)\lambda h,\ n > 0.$$

$$p_0(t+h) = p_0(t-\lambda h),\ n = 0.$$

З першої рівності слідує, що надходження n клієнтів протягом часу $t+h$ можливо в двох випадках:

1) якщо є n надходжень протягом часу t і немає надходжень за час h;

2) або існує $n-1$ надходжень за час t і одне надходження за час h.

Будь-які інші комбінації є неможливими внаслідок того, що протягом малого періоду h можливе надходження тільки однієї події. У відповідності до розділу незалежності подій до правої частини рівняння застосуємо закон множення ймовірностей. У другому рівнянні відсутність надходжень клієнтів протягом інтервалу $t+h$ можлива лише тоді, коли немає надходжень клієнтів за час h.

Виконавши перегрупування членів та переходячи до границі при $h \to 0$, отримуємо наступне:

$$p_n'(t) = \lim_{x\to 0} \frac{p_n(t+h) - p_n(t)}{h} = -\lambda p_n(t) + \lambda p_{n-1}(t),\ n > 0;$$

$$p_0'(t) = \lim_{x\to 0} \frac{p_0(t+h) - p_0(t)}{h} = -\lambda p_0(t),\ n = 0.$$

$$\Rightarrow p_0(t) = e^{-\lambda t}.$$

$$\frac{d}{dt}[e^{\lambda t} p_i(t)] = \lambda p_{i-1}(t) e^{\lambda t} \Rightarrow p_i(t) = e^{\lambda t} \lambda \int_0^t p_{i-1}(t') e^{\lambda t'} dt'.$$

$$p_1(t) = e^{-\lambda t} \lambda \int_0^t e^{-\lambda t'} e^{\lambda t'} dt' = e^{-\lambda t}(\lambda t).$$

Отримуємо розв'язок наведених вище різницево-диференціальних рівнянь, що має наступний вигляд[18]:

$$p_n(t) = \frac{(\lambda t)^n e^{-\lambda t}}{n!},\ n = 0,\ 1,\ 2, ...$$

Кількість «народжень» чи подій на інтервалі часу $(0, t)$ розподіляється за законом Пуассона із параметром λt ($\sim Poi(\lambda t)$).

У цьому випадку ми отримали дискретну щільність ймовірності розподілу Пуассона з математичним очікуванням $M(n|t) = \lambda t$ надходжень за час t.

2.3.2 Оцінка ймовірності участі користувача у підтвердженні наступного блоку блокчейна у залежності від часу сесії

Отриманий вище результат означає наступне – кожен раз, коли тимчасові інтервали між моментами послідовних надходжень заявок розподілені по експоненціальному закону з математичним сподіванням $\frac{1}{\lambda}$, число надходжень заявок в інтервалі рівному t одиниць часу характеризується розподілом Пуассона з математичним очікуванням λt[22]. Вірним є і зворотне твердження.

Відповідно ймовірність того, що у заданий період часу t не наступить жодної події дорівнює

$$p_0(t) = \frac{(\lambda t)^0 e^{-\lambda t}}{0!} = e^{-\lambda t}.$$

Для проведення оцінки ефективності протоколу *Proof of Activity* обчислимо ймовірність участі вузла при побуванні наступного блоку мережі. Як вже раніше зазначалось, як тільки у мережі з'являється «кандидат» на наступний блок транзакцій, він повинен отримати перевірку у середині мережі від N інших учасників, які злегкістю перевірять коректність усіх облислень.

Оцінка ефективності у даному випадку проводиться знаходженням ймовірності згенерувати наступний блок, що буде прийнятий мережею, у період часу, коли користувач знаходиться у мережі. Іншими словами, потрібно обчислити ймовірність хоча б одного стрибку у процесі Пуассона у заданий період часу.

Зручніше у даній ситуації зручніше перейти до зворотньої задачі. Отримуємо

$$P(T) = P\{\text{в заданому інтервалі часу } T \text{ наступила хоча б одна подія}\} =$$
$$= 1 - P\{\text{в заданому інтервалі часу } T \text{ не наступило жодної події}\}.$$

Маємо: $p(t) = 1 - e^{-\lambda t}$.

Із графіку на рисунку 2.4 бачимо, що значення ймовірності появи хоча б однієї події наближається до одиниці з плином часу, тобто при достатньо довгому терміні спостереження, подія рано чи пізно обов'язково наступить. Відповідно, чом довше ведеться очікування на подію, тобто чи більше t, тим вища ймовірність того, що ця подія таки настане. Це видно із графіку функції, що монотонно зростає[16].

Чим вища інтенсивність появи події λ, тим швидше наступить ця подія, оскільки функція буде швидше наближатись до одиниці. На побудованому графіку параметр λ представляє собою крутизну лінії,

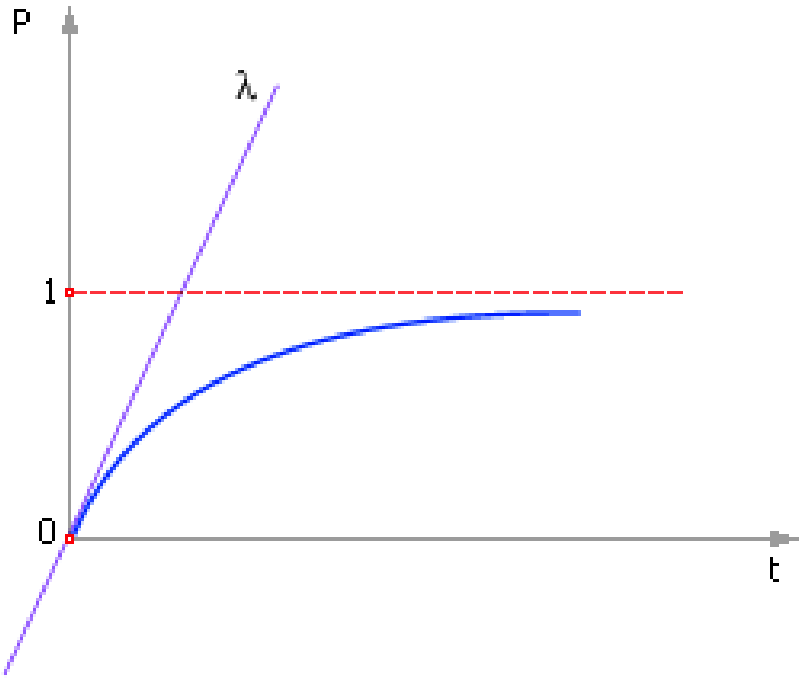

Рисунок 2.4 – Графік ймовірності появи хоча б однієї події з плином часу

тобто нахил дотичної.

З іншої сторони, при збільшенні велечини λ, то при очікуванні одноєї і тієї ж події протягом однакового проміжку часу зростає – рис. 2.5.

Важливою властивістю процесу Пуассона є те, що довжина інтервалу між виникненням сусідніх ризикових подій є взаємно незалежними випадковими величинами, і всі вони мають однаковий експоненційний розподіл ймовірності з параметром λ[23].

Звісно параметр λ на практиці визначається рядом чинників, що на нього впливають. По-перше, це складність обчислення *Proof of Work* складової. Чим складнішим є математична головоломка, тим більше у середньому часу потрібно на перебір можливих варіантів у пошуку кандидата на наступний блок ланцюга. Крім того, важливим є кількість користувачів у мережі в середньому.

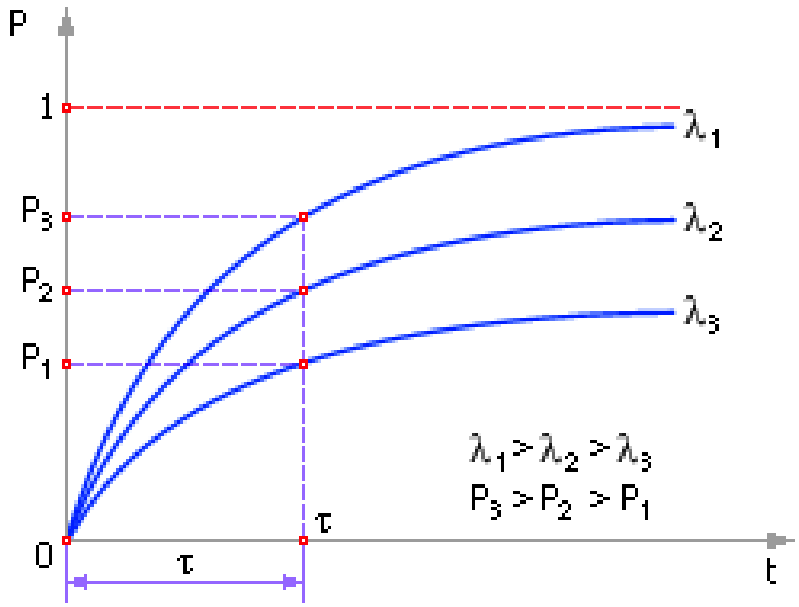

Рисунок 2.5 – Графік ймовірності появи хоча б однієї події з плином часу

2.4 Висновки

У цьому розділі роботи було проведено детальний аналіз математичних моделей протоколів консенсусу *Proof of Burn* та *Proof of Activity*. Перший із них описаний системою масового обслуговування моделлю народження-гибелі, другий – моделлю чистого народження.

Для протоколу *Proof of Activity* було обчислено ймовірність участі користувача у підтвердженні наступного блоку блокчейну у залежності від часу його поточної сесії.

Відповідно для протоколу *Proof of Burn* було обчислено ймовірність участі користувача у розіграші винагоди у залежності від кількості спалених ним монет.

3 ОЦІНКИ ЙМОВІРНОСТЕЙ ГЕНЕРАЦІЇ НОВИХ БЛОКІВ

У цьому розділі наведемо оцінку ефективності у сенсі можливості генерації наступного блоку у протоколах випадковими користувачами.

3.1 Оцінка ймовірності генерації нового блоку протоколу Proof of Burn

По суті, протокол *Proof of Burn* це модифікація *Proof of Stake* протоколу, проте із дещо своєрідним правилом розподілу винагороди. Тобто користувач на черговому етапі розіграшу винагороди може зробити ставку, тобто спалити монети, або взагалі не виконувати ніяких дій.

Отже, проведемо оцінку ймовірності генерації нового блоку у протоколі *Proof of Burn*.

Нехай C – блок, що є кандидатом на наступний блок блокчейну.

$$hash(hash(prev, N, timestamp)) \leq burned(N) \cdot complexity(C),$$

де

– N – користувач із своєю адресою.

– $prev$ – попередній блок у ланцюжку.

– $timestamp$ – мітка часу.

– $complexity(C)$ – складність обчислення блоку. Вона визначається відношенням $\frac{M}{D}$, де $D \in [1, M]$.

– $burned(N)$ – кількість спалених монет користувачем у межах добування нового блоку. Іншими словами – це ставка учасника.

Далі проводи аналогічне виведення ймовірності як у протоколі *Proof-of-Stake*. Введемо час генерації блоку C учасником N – $T(r)$, та обмеження

на дійсні блоки протоколу:

$$U \leq \theta \leq 1,$$

де U – випадкова величина. Вона задана ріномірним розподілом на відрізку $[0,1]$ ($U \sim Un[0,1]$). U визначається наступним чином: нормалізація після гешування даних. Таким чином отримуємо значення із відрізку $[0,1]$.

У випадку протоколу *Proof of Burn* θ задається відношенням спалених учасником монет до складності відповідного блоку – $\theta = \frac{burned(N)}{D}$.

Тоді розподіл величини $T(r)$ матиме вигляд. Нехай n кількість необхідних спроб, щоб досягти дану нерівність.

$P(T(r) \leq t) = P(n \leq t) = 1 - P(n > t) = 1 - (1 - \frac{burned(N)}{D})^t =$
$= 1 - exp(ln(1 - \frac{burned(N}{D})t).$

Оскільки
$\frac{burned(N)}{D} < 1$ та $D = \sum_A burned(A)$,
то

$$\Rightarrow ln(1 - \frac{burned(N)}{D}) \approx -\frac{burned(N)}{D}.$$

$$\Rightarrow P(T(r) \leq t) = 1 - \exp\left(-\frac{burned(N)}{D}t\right).$$

У системі із N користувачів, час, необхідний для знаходження наступного блоку дорівнює $T_C = min(T_1, T_2, ..., T_{N-1}, T_N)$, де T_i – час, необхідний для отримання відповідного значення $i-$им учасником.

$$\Rightarrow P(T_C \leq t) = 1 - \exp(-\frac{t}{D}\sum_{i=1}^{N} burned(A_i)).$$

Бачимо, що необхідний час для генерації блоку має експоненційний розподіл із параметром $\lambda = -\frac{\sum_{i=1}^{N} burned(A_i)}{D}$.

Отже, ймовірність генерації наступного блоку випадковим учасником i дорівнює

$$P(T_C = T_i) = \frac{burned(A_i)}{\sum_{j=1}^{N} burned(A_j)}.$$

Отриманий результат свідчить про те, що майнинг у протоколі *Proof of Burn* є справедливим процесом відносно кожного користувача.

3.2 Оцінка ймовірності генерації нового блоку протоколу Proof of Activity

Як вже було проаналізовано раніше, протокол *Proof-of-Activity* поєднує у собі 2 протоколи консенсусу – *Proof-of-Work* та *Proof-of-Stake*.

Тобто, у цьому алгоритмі майнинг виглядає абсолютно «традиційним», як у *Bitcoin*. Користувачі змагаються у вирішенні математичної головоломки.

У момент її вирішення, система перемикається на *Proof-of-Stake* складову. На основі інформації в заголовку вибирається випадкова група валідаторів для підписування нового блоку. Чим більшою кількістю монет в системі володіє валідатор, тим вища ймовірність того, що він чи вона буде обрана. Шаблон стає повноцінним блоком, як тільки всі валідатори його підпишуть[27].

Нехай C – кандидат на новий блок у межах даного протоколу із коректно обчисленою PoW складовою. Іншими словами

$$hash(hash(prev, N, timestamp)) \leq activity(N)\frac{M}{D},$$

У цьому запису визначені наступні величини:

– N – користувач та відповідно його адреса.

– $prev$ – попередній блок у блокчейні.

– $timestamp$ – мітка часу.

– D – складність блоку. На цю величину також додається умова: $D \in [1, M]$.

– $activity(N)$ – числове значення, що виражає активність користувача.

По аналогії із *Proof-of-Stake*, вводимо $T(r)$, що відповідає за час у секундах необхідний для генерації блоку C користувачем N.

Введемо обмеження на дійсні блоки протоколу.

$$U \leq \theta \leq 1.$$

U – випадкова величина, що визначена ріномірним розподілом на відрізку $[0, 1]$, тобто $U \sim Un[0, 1]$. Ця величина отримана шляхом гешування даних та їх нормалізації для отримання значення на відрізку $[0, 1]$.

У протоколі *Proof-of-Activity* величина θ визначається відношенням активності користувача до складності блоку, тобто $\theta = \frac{activity(N)}{D}$.

Тоді розглянемо розподіл величини $T(r)$. Нехай n кількість необхідних спроб, щоб досягти дану нерівність. Тоді:

$P(T(r) \leq t) = P(n \leq t) = 1 - P(n > t) = 1 - (1 - \frac{activity(N)}{D})^t =$
$= 1 - exp(ln(1 - \frac{activity(N}{D})t)$.

Враховуючи те, що активність користувача визначається системою випадково, це означає, що один користувач не може буде єдиним валідатором, отже $\frac{activity(N)}{D} < 1$ та $D = \sum_A activity(A)$.

$$\Rightarrow ln(1 - \frac{activity(N)}{D}) \approx -\frac{activity(N)}{D}.$$

$$\Rightarrow P(T(r) \leq t) = 1 - \exp\left(-\frac{activity(N)}{D}t\right).$$

Нехай у системі присутні N користувачів. Відповідно час знаходження наступного дійсного блоку дорівнює

$T_C = min(T_1, T_2, ..., T_{N-1}, T_N)$. Величина T_i відповідає за час, необхідний для отримання відповідного значення $i-$им учасником. Тоді

$$P(T_C \leq t) = 1 - \exp(-\frac{t}{D} \sum_{i=1}^{N} activity(A_i)).$$

Отже, час, що необхідний для генерації нового блоку ланцюжку має експоненційний розподіл із параметром $\lambda = -\frac{\sum_{i=1}^{N} activity(A_i)}{D}$.

З цього випливає, що ймовірність генерації наступного блоку випадковим користувачем i дорівнює

$$P(T_C = T_i) = \frac{activity(A_i)}{\sum_{j=1}^{N} activity(A_j)}.$$

Виведення дуже схоже на аналогічну оцінку для протоколу *Proof of Stake*, проте у протоколі *Proof of Activity* ключову роль відіграє саме активність користувача.

З отриманого результату видно, що процес майнингу у цьому протоколі є справедливим для кожного із учасників системи. Іншими словами, ймовірність згенерувати наступний блок випадковим користувачем дорівнює його «активності у мережі».

Згідно проведеного дослідження можна зробити висновок, що критичні моменти у роботі цього протоколу співпадають із проблемою його складових, тобто зі сторони *Proof-of-Work* це потреба у великій обчислювальній потужності та енергії для неї.

Частина, що визначається протоколом *Proof-of-Stake* відповідно має свій недолік – ніщо не обмежує валідатора двічі підписати блок. Так звана проблема подвійного підписання[28].

3.3 Висновки

У даному розділі було оцінено ефективність протоколів консенсусу *Proof of Burn* та *Proof of Activity*.

Крім того було обчислено ймовірність генерації нового блоку випадковим користувачем для обох протоколів *Proof of Burn* та *Proof of Activity*.

На основі отиманих результатів можна зробити висновок, що протоколи консенсусу *Proof of Burn* та *Proof of Activity* є справедливими щодо користувачів з точки зору майнингу. У цих протоколах ймовірність успіху залежить від основного ресурсу. Для *Proof of Burn* це кількість спалених монет, для *Proof of Activity* – активність учасника.

ВИСНОВКИ

У ході даної роботи був проведений аналіз опублікованих джерел за тематикою дослідження, а саме:

– протоколи консенсусу *Proof of Work*, *Proof of Stake*, *Proof of Burn* та *Proof of Activity*,

– системи масового обслуговування

– моделі чистого народження та народження-гибелі систем масового обслуговування

Враховуючи те, що обрані протоколи є досить новими на ринку криптовалют, постало питання дослідження їхньої ефективності з точки зору користувачів, а саме ймовірністі генерації наступного блоку блокчейну.

Для цього було детально побудовано математичні моделі протоколів *Proof of Burn* та *Proof of Activity*, обчислено наступні ймовірності:

– Для протоколу *Proof of Activity* – ймовірність участі користувача у підтвердженні наступного блоку в залежності від часу його поточної сесії,

– для протоколу *Proof of Burn* – ймовірність участі користувача у розіграші винагоди у залежності від кількості спалених монет,

– для обох протоколів – ймовірності генерації нового блоку випадковим користувачем.

Отримані результати показують, що *Proof of Burn* та *Proof of Activity* є справедливими щодо користувачів з точки зору майнингу. У цих протоколах ймовірність успіху залежить від основного ресурсу. Для *Proof of Burn* це кількістьспалених монет, для *Proof of Activity* – активність учасника.

На основі отриманих результатів можна продовжити дослідження даних протоколів консенсусу, зокрема детально дослідити стійкість до атак подвійних витрат (*double-spends attacks* та атак форків. Враховуючи те, що обидва протоколи базуються на *Proof of Work* та *Proof of Stake*

системах, є потенційний напрям подальших досліджень та потенційних модифікацій даних протоколів.

ПЕРЕЛІК ПОСИЛАНЬ

1. What is Cryptocurrency: Everything You Must Need To Know! [Електронний ресурс]. — Режим доступу: https://blockgeeks.com/guides/what-is-cryptocurrency/.

2. Proof of burn – Bitcoin Wiki [Електронний ресурс]. — Режим доступу: https://en.bitcoin.it/wiki/Proof_of_burn.

3. Proof of Burn (Cryptocurrency) – Investopedia. — Режим доступу: https://www.investopedia.com/terms/p/proof-burn-cryptocurrency.asp.

4. Bentov, I., Gabizon, A., and Mizrahi, A. 2014. Cryptocurrencies without Proof of Work. Preprint. [Електронний ресурс]. — Режим доступу: http://www.cs.technion.ac.il/~idddo/CoA.pdf.

5. Bitcoin wiki. P2SH. [Електронний ресурс]. — Режим доступу: https://github.com/bitcoin/bips/blob/master/bip-0016.mediawiki.

6. Hearn, M. 2011. Bitcoin wiki: Contracts. [Електронний ресурс]. — Режим доступу: https://en.bitcoin.it/wiki/Contracts.

7. Maxwell, G. 2011b. Bitcoin wiki: Zero knowledge contingent payment. [Електронний ресурс]. — Режим доступу: https://en.bitcoin.it/wiki/Zero_Knowledge_Contingent_Payment.

8. Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake. Cryptology ePrint Archive. [Електронний ресурс]. — Режим доступу: https://eprint.iacr.org/2014/452.pdf.

9. Litecoin wiki. Comparison between litecoin and bitcoin. [Електронний ресурс]. — Режим доступу: http://litecoin.info/User:Iddo/Comparison_between_Litecoin_and_Bitcoin#Faster_transaction_time.

10. Rosenfeld, M. 2012b. Dynamic block frequency. Bitcoin forum thread. [Електронний ресурс]. — Режим доступу: https://bitcointalk.org/index.php?topic=79837.0;all.

11. Sompolinsky, Y. and Zohar, A. 2013. Accelerating bitcoin's transaction processing. [Електронний ресурс]. — Режим доступу: http://

www.cs.huji.ac.il/~avivz/pubs/13/btc_scalability_full.pdf.

12. Babaioff, M., Dobzinski, S., Oren, S., and Zohar, A. 2012. On bitcoin and red balloons. In ACM Conference on Electronic Commerce, B. Faltings, K. Leyton-Brown, and P. Ipeirotis, Eds. ACM, 56–73.

13. Proof of Burn | Consensus Through Coin Destruction. Coincentral. [Електронний ресурс]. — Режим доступу: https://coincentral.com/proof-of-burn/.

14. What is a Coin Burn? Beginner's Guide to Proof of Burn. Blockonomi. [Електронний ресурс]. — Режим доступу: https://blockonomi.com/proof-of-burn/.

15. Курсовая по системах массового обслуживания. Studfiles. [Електронний ресурс]. — Режим доступу: https://studfiles.net/preview/3606999/page:2/

16. Лекция 28. Поток случайных событий. Stratum. [Електронний ресурс]. — Режим доступу: http://stratum.ac.ru/education/textbooks/modelir/lection28.html

17. Birth-death processes. Netlab. [Електронний ресурс]. — Режим доступу: https://www.netlab.tkk.fi/opetus/s383143/kalvot/E_bdpros.pdf

18. Birth-death processes. Suaybarslan. [Електронний ресурс]. — Режим доступу: http://www.suaybarslan.com/birthdeathprocess4datamodelling.pdf

19. Birth and death processes and order statistics. Neutrino. [Електронний ресурс]. — Режим доступу: http://neutrino.aquaphoenix.com/ReactionDiffusion/SERC5chap4.pdf

20. Системы массового обслуживания: Методические разработки / Составители Т.Я. Лазарева, И.В.Диденко, Рецензент В.Г. Серегина, Тамбов. Тамбовский государственный технический университет, 2001 год.

21. М. В. Швецкий «Информатика и программирование Шаг за шагом».// it.kgsu.ru. Режим доступу: http://it.kgsu.ru/IO

22. Коваленко И.Н. «Теория массового обслуживания», 1963 рік, видавництво ВИНИТИ, 125 с.

23. Reliability and Risk Analysis. moodle.unob.cz [Електронний ресурс]. — Режим доступу: https://moodle.unob.cz/pluginfile.php/43057/mod_resource/content/1/Poisson%20process.pdf

24. Proof of burn. en.bitcoin.it. [Електронний ресурс]. — Режим доступу: https://en.bitcoin.it/wiki/Proof_of_burn

25. What is Proof of Burn?. 99bitcoins. [Електронний ресурс]. — Режим доступу: https://99bitcoins.com/what-is-proof-of-burn/

26. Blockchain Consensus Algorithms: The Proof of Stake slice. medium.com. [Електронний ресурс]. — Режим доступу: https://medium.com/coinmonks/blockchain-consensus-algorithm-the-proof-of-stake-slice-a4bda6658bbe

27. A (Short) Guide to Blockchain Consensus Protocols. www.coindesk.com. [Електронний ресурс]. — Режим доступу: hthttps://www.coindesk.com/short-guide-blockchain-consensus-protocols

28. Proof of Stake. en.bitcoin.it. [Електронний ресурс]. — Режим доступу: https://en.bitcoin.it/wiki/Proof_of_Stake