УДК 004.6:004.7 DOI: 10.26565/2519-2310-2021-2-03

ДОСЛІДЖЕННЯ ПРОЦЕСІВ ПОШИРЕННЯ ІНФОРМАЦІЇ В ДЕЦЕНТРАЛІЗОВАНИХ МЕРЕЖАХ

Дмитро Орлов, Ірина Гальцева

Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна
d.eagle@ukr.net, irina.galceva@karazin.ua

Рецензент: Микола Карпінський, д.т.н., проф., університет Бельсько-Бяла, Бельсько-Бяла, Польща.
mkarpinski@ath.bielsko.pl

Надійшла: Жовтень 2021.

Анотація: *У статті наведені результати аналізу процесів розповсюдження вузлів, які створюють децентралізовану мережу, зберігають і генерують інформацію цієї мережі, та взаємодіють між собою впливаючи на загальний стан системи. Проведено моделююче дослідження ефективності дослідного алгоритму системи моніторингу і аналізу поведінки вузлів відповідної системи. Надано порівняння отриманих результатів з вже існуючими рішеннями. Для дослідження топології застосовано тестову версію програми, яка дозволяє проводити збір інформації про функціонуючи вузли шляхом безпосередньої взаємодії. Дослідження проводились на прикладі однорангових децентралізованих систем мережі Bitcoin. За результатами роботи було створено програмний продукт, який аналізує мережу Bitcoin, будує її топологію, відслідковує зміни, які здійснилися у мережі, та забезпечує візуалізацію результатів у режимі реального часу. Під час вивчення наявних реалізацій процесів розповсюдження в децентралізованих системах, було досліджено питання стосовно відкриття пірів та їх управління. Розглянуто проблему відкриття топології в біткоїн мережі та основні методи виявлення топології. Запропоновано альтернативний метод вирішення цього питання.*

Ключові слова: *блокчейн; біткоїн; децентралізація; топологія біткоїн мережі; криптовалюта*.

1 Вступ

Інтернет-комерція в більшості випадків спирається на фінансові установи, які виступають в ролі довірених посередників для проведення електронних платежів. Така схема добре працює для більшості транзакцій, але в її основі лежить довіра, що тягне за собою певні проблеми. Необхідність посередництва фінансових інститутів перешкоджає здійсненню незворотних транзакцій. Ціна цих послуг збільшує вартість транзакцій і встановлює мінімальну їх ціну, роблячи непрактичним проведення нечастих і невеликих транзакцій. Крім того, відсутність незворотних транзакцій збільшує і вартість сервісів, чиї послуги є невід'ємними. Оскільки платіж можна анулювати, продавець змушений бути насторожі, вимагаючи від покупця більше інформації, ніж йому необхідно. І певний відсоток шахрайства приймається просто, як неминучість. Ці націнки і невизначеності з банківськими платежами можуть бути подолані в разі використання готівки, однак механізму для проведення прямих електронних транзакцій поки не існує. При цьому, криптовалютні платіжні системи, що засновані на криптографії, а не на довірі, дозволяють будь-яким двом учасникам здійснити переказ коштів безпосередньо, без участі посередника. Крім того, обчислювальна «дорожнеча» процедури скасування транзакцій, потенційно, може відгородити продавців від шахрайства, а легко здійснювані механізми, відповідним чином, захистили б покупців.

2 Огляд відомих реалізацій процедур розповсюдження у децентралізованих мережах та розгляд запропонованого методу

У мережі біткоїн, піри (*тобто, учасники мережі*) ініціюють одну з можливих реалізацій *Bitcoin P2P* протоколу: понад 75% пірів мають версії Bitcoin Core *(клієнт Satoshi)*, а менше 10 % пірів мають інші реалізи (*наприклад, такі як BitcoinJ, Libbitcoin або btcd).* Всі вони можуть реалізовувати кілька функцій, таких як маршрутизація, використання гаманця ,бази даних блокчейн, майнинг, та запуск різних протоколів [1].

Гаманець - це засіб для здійснення обробки і циркуляції біткоїнів (*надсилання, отримання та зберігання*). Усі піри повинні реалізувати функцію маршрутизації для енергійної роботи в мережі. Інші функції, також, можна комбінувати для визначення різних цілей для пірів у мережі Bitcoin. Переважно розрізняється дві ролі: повний вузол та легкий гаманець. Повний вузол може бути або повним блокчейн-вузлом, коли він включає маршрутизацію та повну функцію бази даних Blockchain, або «соло майнером», коли поєднується маршрутизація, повна база даних (БД) Blockchain та майнинг функції. При цьому пір, який виконує цю роль, споживає велику кількість ресурсів (*CPU-GPU, потужність, дисковий простір, тощо*). Роль «легкого» гаманця (SPV) інтегрує функції маршрутизації та гаманця. На відміну від повної ролі вузла, він розроблений для пірів із обмеженими ресурсами.

2.1 Відкриття пірів

Коли процес тільки ініціюється, пір не має відомостей про «активних» сусідів, що мають «повну» роль вузла. Щоб визначити їх адреси (*загальнодоступний IP та номер порту*), необхідно використовувати кілька механізмів для проведення однорангового пошуку: взяття збережених у ньому адрес (*локальна БД адрес*), що виконуються по внутрішній команді консольного клієнту біткоїн (*-addnode та -connect*), надсилання DNS запиту (запитуючи інші піри), надсилання повідомлення *getaddr()* або чекаємо на повідомлення відповіді від вузлів *addr()*. Сформований список адрес зберігається і оновлюється в локальній БД адрес [2].

2.2 Стохастичне управління одноранговими адресами

Біткоїн пір підтримує БД локальних адрес, яка поєднує виявлені аналоги. Для кожного запису ця база включає: номер порту, IP-адресу однорангового партнера, часову позначку останнього з'єднання, тощо. Для того, щоб сприяти вибору «сусідів», усі адреси пірів, які зберігаються в БД, систематизуються та заносяться у відповідний «контейнер». Існує два типи контейнерів: - контейнер для випробуваних адрес та контейнери для нових адрес. При цьому, контейнери для випробуваних адрес містять адреси пірів, з якими вони мали, принаймні, хоча б одну «вдалу» вхідну/вихідну взаємодію (*тобто з'єднання*).

Контейнери для нових адрес складають адреси пірів, з якими не було конекту (*встановленого раніше*), або адреси, вилучені із спроб контейнеру. Існує 256 контейнерів для випробуваних, та, відповідно, 1024 контейнера для нових адрес. Кожен контейнер може зберігати max 64 адреси. Т.ч., загальна кількість записів в БД обмежено 81920 адресами.

Якщо вхідний або вихідний коннект з піром успішно встановлено, то пізніше, ця адреса буде зафіксована у контейнерах для випробуваних адрес. Процес вибору відповідного випробуваного контейнеру визначається функцією *GetTriedBucket1 (див. рис. 1)*.

```
сКлюч: секретний ключ з 256 біт для рандомізації вибору
контейнеру
Ап: IPv4-адреса піру та його номер порту
Група:  16  IPv4  префіксів пірів  IPv4 адрес .
1. хеш1 ← хеш(сКлюч, Ап) % 8
 2.хеш2 ← хеш(сКлюч, група, hash1)
3.контейнер ← хеш2 % 256
4.Повернути контейнер
```

Рис. 1 – Функція GetTriedBucket1

Кожна 16 група IPv4, випадковим чином, обирає 8 контейнерів із 256, а потім, теж випадковим чином визначає, одне з них на базі адресу піру (*IPv4 та номер порту*). Якщо адреса піру присутня у виділеному випробуваному контейнері, то його часовий маркер оновлюється. В іншому випадку адреса пірів вставляється в 1-ше доступне положення. Якщо вибраний контейнер «заповнений», то 4-ри випадкові записи визначаються випадковим чином, а найстарше, переходить з нього назад, до нових контейнерів. Щоб збільшити кількість випро-

буваних адрес у випробуваних контейнерах, вузол, після встановлення вихідних з'єднань, випадковим чином і періодично, підбирає піри з нових контейнерів. При цьому він намагається підключитися до них і додає IP-адреси з вдалим підключенням до контейнеру. Для нових адрес, вибирається новий контейнер, за рахунок функції *GetNewBucket1* (*див. рис. 2*).

```
сКлюч: секретний ключ з 256 біт для рандомізації вибору
контейнеру
Ап: IPv4-адреса піру та його номер порту
Група: 16 IPv4 префіксів пірів IPv4 адрес
Вих група : Префікс / 16 IPv4 вихідних  IPv4 адрес
рекламуючих
IPv4-адреси пірів .
1. хеш1 ← хеш(сКлюч, Група р, Вих група) % 64
 2.хеш2 ← Hash(сКлюч, Вих група , хеш1)
3.Контейнер ← хеш2 % 1024
 4.Повернути Контейнер
```

Рис. 2 - Функція *GetNewBucket1*

Кожна пара *(«група», «Вих група»)* випадковим чином визначає нові 64 контейнери та лише один з них обирається з використанням інформації про вхідну групу. Оскільки IPv4-адресу, яку потрібно інтегрувати, можна рекламувати одразу з декількох джерел, то вона може поєднувати до 8 різних відрізків.

При додаванні нової адреси до повного контейнеру, випадковим чином вилучається обрана адреса, з так званих, «не валідних» адрес. Адреса вважається не діючою, якщо:

а) її часовий маркер випереджається на більш ніж 10 хв. поточного часу;

б) відмічено 7 послідовних збоїв конекту за останній тиждень;

в) її часовий маркер старший більш ніж на тиждень.

2.3 Вибір випадкових «сусідів»

Біткоїн пір може відібрати до 8 (*значення за замовчуванням*) *max* вихідних з'єднань (*до інших пірів, що мають повну роль вузла*), а потім підтримувати їх. Пропускна здатність кожного вихідного конекту обмежена 160 кбіт / с, щоб запобігти перевантаження наявної пропускної здатності каналів, трафіком Bitcoin. Крім того, кожен пір може приймати *max* 117 (*значення за замовчуванням*) вхідних сусідів, або навпаки, повністю відмовитись від будь-яких вхідних взаємодій. Однак, ця відмова недоцільна, з точки зору для підтримки мережі Bitcoin. Визначення алгоритму нового вихідного з'єднання здійснюється щоразу, коли поточна кількість вихідних з'єднань менше 8. Це може статися, наприклад, коли пір перенавантажується, або коли один із його вихідних сусідів припиняє з'єднання. Як правило, пір може «закрити» вхідні з'єднання, але ніколи не повинен припиняти вихідні з'єднання, за винятком випадків, коли його вихідний сусід перебуває у забороненому реєстрі (*Stop листі*).

2.4 Труднощі відкриття топології мережі біткоїн

Існуюча структура топології мережі Bitcoin невідома, оскільки протокол Bitcoin не забезпечує виявлення топології. Ця функціональність свідомо не підтримується, щоб захистити мережу Bitcoin від проведення потенційних мережевих атак (*наприклад, Eclipse, Sybil та ін.*). Варто підкреслити, що існують рішення, які опосередковано відображають топологію Bitcoin, використовуючи для цього інформацію (*яка, від самого початку, не призначена для виявлення топології*) в протоколі Bitcoin [3].

2.5 Картографування топології на основі часових маркерів

Такий підхід був вперше використаний у *Coinscope*. В цьому разі БД локальних адрес, що підтримується кожним з пірів, містить відомості про інші піри, включаючи часову мітку. Ці часові позначки допомагають встановити ступінь новизни піру, а також, чи є він «сусі-

дом» вхідного / вихідного. Це пов'язано з тим, що ці часові маркери оновлюються щоразу, коли піри здійснюють взаємодіють, за умови, що відповідна часова мітка є старшою в заданої межі часового відрізку. Розміщуючи відповідні верхню та нижню границю на цій межі часових міток, для будь-якого піру, можна визначати його вихідних та вхідних сусідів.

Зазначений підхід «працював» до 0.10 версії клієнта ***Bitcoin Core***. Потім відбулися зміни, які оновлювали часові маркери лише після відключення даного піру. Таким чином стало неможливо виявляти вихідних та/або вхідних сусідів для пірів, які реалізують останні версії, використовуючи розглянутий вище підхід [3].

2.6 Топологія на основі концепції атаки Sybil

Даний підхід був запропонований відомим криптографом Люксембургського університету Алексом Бірюковим. Так, розроблений їм фреймворк спочатку підключається в мережі до всіх аналогів Bitcoin, а вже потім кожний стійкий коннект піддається з його сторони флуду (*flood*). Після цього даний фреймворк очікує від інших пірів, отримання всіх підроблених їм адрес назад, оскільки цільовий аналог передав їх вихідному / вхідному каналу сусідів.

Слід зазначити, що для даного підходу притаманно декілька особливостей:

а) при отриманні пересланого повідомлення addr() (*яке містить відповідні фальшиві адреси*), програмна структура фреймворку не може гарантувати, що відправник є першочерговим сусідом «атакованого» їм вузла [3];

б) програмний каркас отримує переслані повідомлення *«addr()»*, лише в разі, якщо він визначений пірами, як один з 1-2 випадково обраних «сусідів»;

в) для забезпечення більш коректної оцінки топології мережі Bitcoin, відповідні програмні каркаси повинні мати зв'язки з усіма пірами в межах цієї мережі.

2.7 Метод відображення топології, що пропонується

Отримавши зміни в протоколі Bitcoin, пропонується застосувати новий метод відображення оцінки топології Bitcoin мережі. Дослідний алгоритм візуалізації топології, спочатку, в режимі реального часу, взаємодіє з мережею Bitcoin. На цьому етапі забезпечується перевірка доступності мережі, та будується і оновлюється БД локальних адрес для кожного з вузлів, запитуючи її кілька разів протягом визначеного терміну часу. Після цього, алгоритм отримує в своє розпорядження всю необхідні відомості для того, щоб визначити сусідів для кожного з пірів, які емулюють реальну поведінку вузлів Bitcoin.

Як тільки дані про всіх сусідів для кожного з пірів отримані, дослідний алгоритм генерує відповідну мапу на основі відомостей, що були отримані від вузлів мережі біткоїн. Стуктурно-логічна схема роботи даного методу (*реалізовано на мові Python*) наведено на рис. 3.

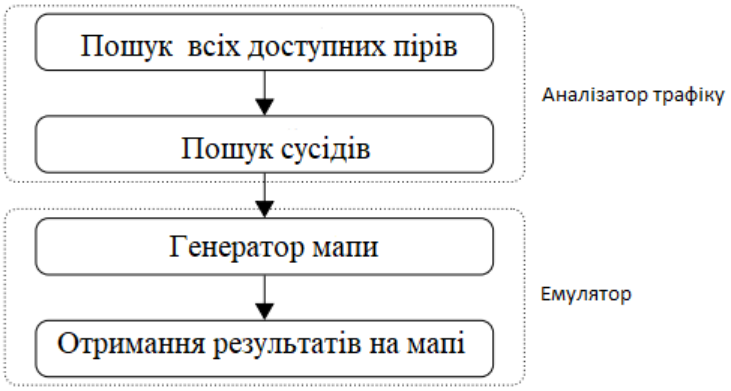

Рис. 3 – Спрощена схема методу

3 Отримані результати

3.1 Порівняння отриманих результатів мапою Bitnodes.io

Мапа *bitnodes.io* оперує даними про 13381 вузлів *(станом на 10.10.21)*, які є найбільш активними в мережі. Характерний вид цієї мапи наведено нижче, на рис. 4. Також, ця мапа оперує даними про країни, де зосереджена найбільша кількість активних вузлів (див. рис. 5).

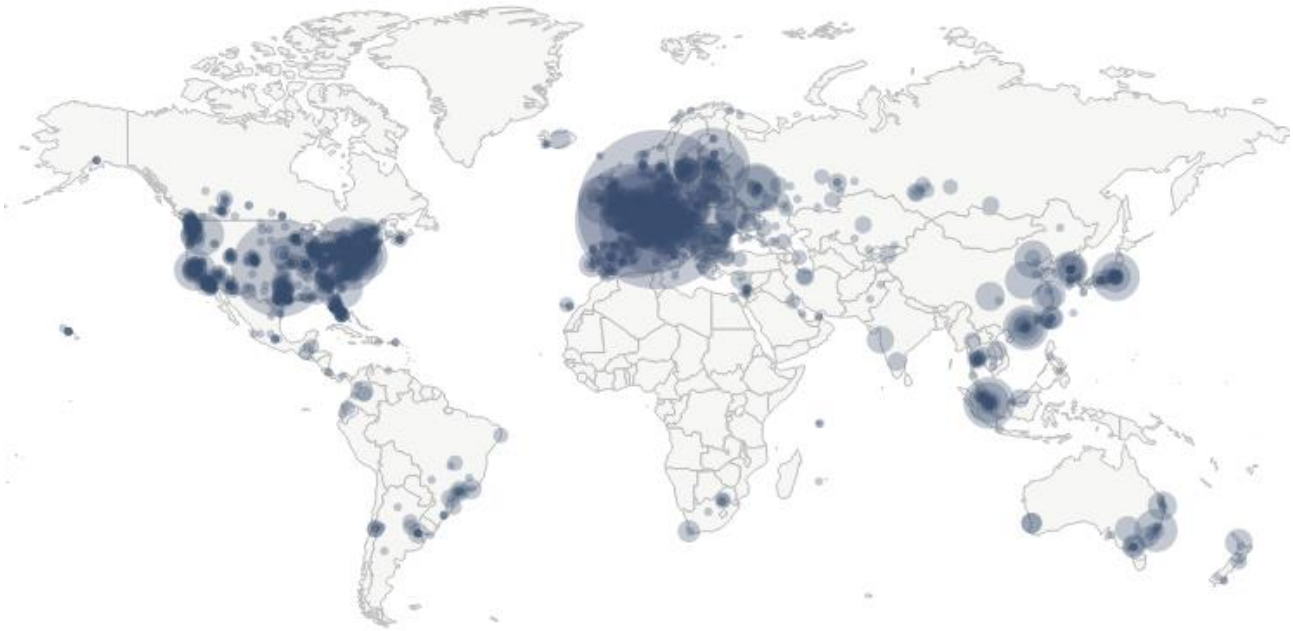

Рис. 4 – Мапа місцезнаходження біткоїн пірів *(за даними bitnodes.io)*

RANK	COUNTRY	NODES
1	n/a	5965 (44.58%)
2	United States	1859 (13.89%)
3	Germany	1807 (13.50%)
4	France	532 (3.98%)
5	Netherlands	377 (2.82%)
6	Canada	311 (2.32%)
7	United Kingdom	267 (2.00%)
8	Russian Federation	190 (1.42%)
9	Finland	187 (1.40%)
10	Switzerland	133 (0.99%)

Рис. 5 – Поточні дані про країни з найбільшою кількістю вузлів *(за даними bitnodes.io)*

Якщо порівняти наявні, на час проведення моделювання, статистичні дані, то можна побачити, що відомості, котрі були отримані запропонованим методом (*мапа на рис. 6 та, відповідна кількість вузлів на рис. 7*), у своєму відсотковому співвідношенні, практично співпадають з даними, які були отриманими з ресурсу *bitnodes.io*.

Різниця між двома мапами полягає в тім, що при використанні запропонованого методу його алгоритм оперує набагато більшою кількістю даних, а саме 58440 вузлів, в той час як *bitnodes.io* демонструє дані лише про 13381 пір. Різниця відчутна, що безсумнівно є великою перевагою даного методу дослідження топології користувачів біткоїн мережі. В середньому, за вісь цикл моделювання, кількість даних, що отримані з використанням методу, який пропонується, приблизно в 4,2–4,5 рази більша ніж даних, якими оперує *bitnodes.io*.

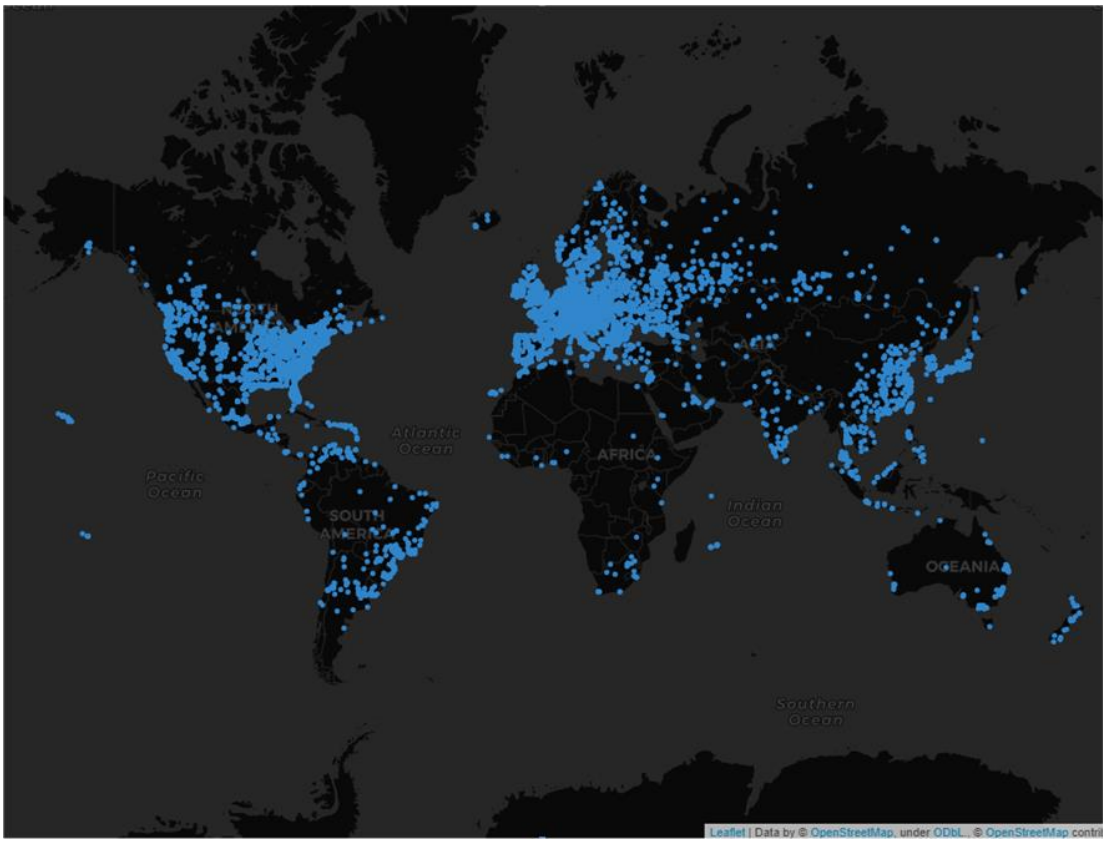

Рис. 6 - Мапа локації біткоїн пірів, яка отримана запропонованим методом

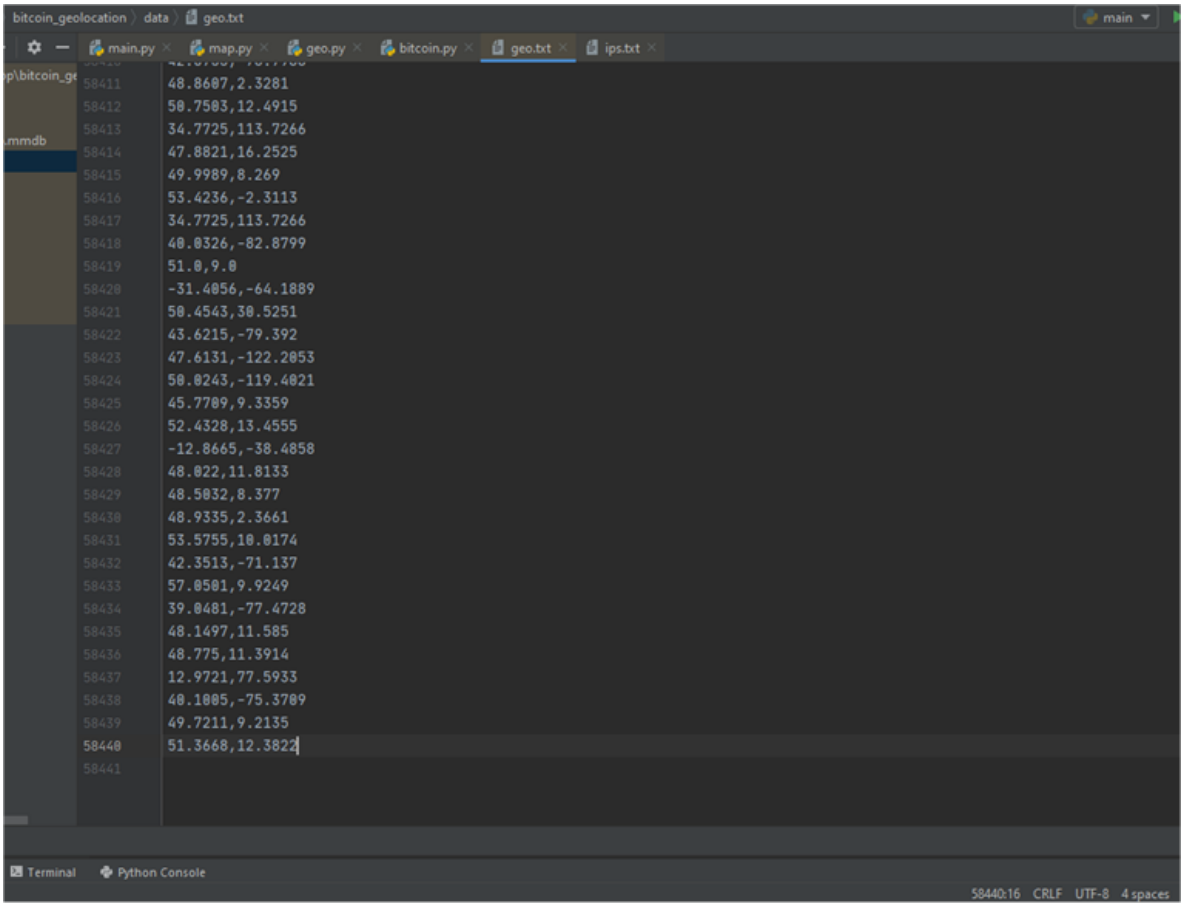

Рис. 7 – Кількість отриманих геолокацій
(*фрагмент інтерфейсу тестового програмного додатку*)

Зазначена перевага дозволяє отримати помітно більший обсяг геоданих, стосовно активних пірів, і як наслідок, синтезувати більш адекватну, для поточної ситуації, структуру мережі. Відповідно, детальний аналіз більш ємних та інформативних вихідних даних, надає змогу сформувати набагато більш об'єктивну картину користувачів мережі біткоін.

4 Висновки

1. За результатами проведеного циклу досліджень запропоновано новий метод аналізу мережі біткоїн, який будує її топологію, відслідковує всі поточні зміни, що здійснилися у мережі, та здійснює візуалізацію узагальнених відомостей в режимі реального часу.

2. Запропонований метод відрізняється від відомих тим, що формує відчутно більш адекватну *(за кількості вузлів, що аналізуються)* структуру мережі, чим створює відповідні умови для покращення можливостей автоматизованих систем моніторингу змін у децентралізованих мережах, та посилює функції відстеження аномальної активності в мережі.

3. Отримані результати можуть бути використані при реалізації систем моніторингу та захисту мережевих систем, які створені за децентралізованим принципом.

Посилання

[1] A. Antonopoulos, Mastering Bitcoin: Programming the Open Blockchain, 2nd Edition. Reading, MA: AO'Reilly Media, 2017.
[2] Satoshi client node discovery. DOI: https://en.bitcoin.it/wiki/Satoshi_Client_Node_Discovery
[3] BTCmap: Mapping Bitcoin Peer-to-Peer Network Topology. Varun Deshpande, Hakim Badis, Laurent George. – 2018

Reviewer: Mikołaj Karpiński, Dr. of Sciences (Eng.), Full Prof., University of Bielsko-Biala, Bielsko-Biala, Poland.
E-mail: mkarpinski@ath.bielsko.pl

Received on October 2021.

Authors:
Dmytro Orlov, student, Faculty of Computer Science, V. N. Karazin Kharkiv National University, Kharkov, Ukraine.
E-mail: d.eagle@ukr.net

Irina Galceva, Senior Lecturer, Department of Security of Information Systems and Technologies, V.N. Karazin Kharkiv National University, Ukraine.
E-mail: irina.galceva@karazin.ua

Research of information dissemination processes in decentralized networks.

Abstract. The article presents the results of the analysis of the processes of node propagation, which create a decentralized network, store and generate information of this network, and interact with each other influencing the general state of the system. A modeling study of the effectiveness of the experimental algorithm of the system of monitoring and analysis of the behavior of the nodes of the corresponding system was carried out. A comparison of the obtained results with existing solutions is given. To study the topology, a test version of the program was used, which allows to collect information about functioning nodes by direct interaction. The research was conducted on the example of peer-to-peer decentralized Bitcoin network systems. As a result of the work, a software product was created that analyzes the Bitcoin network, builds its topology, and tracks the changes that have taken place dreamed in the network, and provides visualization of results in real time. During the study of the existing implementations of distribution processes in decentralized systems, the issues concerning the opening of peers and their management were investigated. The problem of topology discovery in bitcoin network and basic methods of topology detection are considered. An alternative method of solving this problem is proposed.

Keywords: Blockchain; Bitcoin; Decentralization; Bitcoin Network Topology; Cryptocurrency.

Рецензент: Николай Карпинский, д.т.н., проф., Университет Бельсько-Бяла, ул. Виллова 2, 43-309 Бельсько-Бяла, Польша.
E-mail: mkarpinski@ath.bielsko.pl

Поступила: Октябрь 2021.

Авторы:
Дмитрий Орлов, студент факультета компьютерных наук, Харьковский национальный университет имени В. Н. Каразина, Харьков, Украина.
E-mail: d.eagle@ukr.net

Ирина Гальцева, ст. преподаватель кафедры Безопасности информационных систем и технологий, ХНУ имени В.Н. Каразина, Харьков, Украина.
E-mail: irina.galceva@karazin.ua

Исследование процессов распространения информации в децентрализованных сетях.

Аннотация: В статье приведены результаты анализа процессов распространения узлов, которые создают децентрализованную сеть, сохраняют и генерируют информацию этой сети, взаимодействуют между собой, влияя на общее состояние системы. Проведено моделирующее исследование эффективности опытного алгоритма системы мониторинга и анализа поведения узлов соответствующей системы. Приведено сравнение полученных результатов с уже существующими решениями. Для исследования топологии применена тестовая версия программы, которая позволяет проводить сбор информации о функционирующих узлах путем непосредственного взаимодействия. Исследования проводились на примере одноранговых децентрализованных систем сети Bitcoin. По результатам работы создан программный продукт, который анализирует сеть Bitcoin, строит ее топологию, отслеживает изменения, которые произошли в сети, и обеспечивает визуализацию результатов в режиме реального времени. При изучении имеющихся реализаций процессов распространения в децентрализованных системах, были исследованы вопросы открытия пиров и их управления. Рассмотрена проблема открытия топологии в биткоин сети и основные методы выявления топологии. Предложен альтернативный метод решения этого вопроса.

Ключевые слова: блокчейн; биткоин; децентрализация; топология биткоин сети; криптовалюта.