Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего профессионального образования
«Московский физико-технический институт (государственный университет)»
Национальная академия наук Украины
Физико-технический учебно-научный центр
Факультет управления и прикладной математики
Кафедра теоретической кибернетики и методов оптимального управления
(на базе Института кибернетики имени В.М. Глушкова НАН Украины)
Кафедра теоретической кибернетики и методов оптимального управления

На правах рукописи
УДК 539.12

Быхун Алексей Викторович

Исследование реализации задачи распределенной сортировки в среде технологий блокчейн

Выпускная квалификационная работа (магистерская диссертация)

Направление подготовки 03.04.01 «Прикладные математика и физика»

Выполнил _____ / Быхун А.В. /
Научный руководитель _____ / Стецюк П.И. /
Рецензент _____ / Петрухин В.А. /

Москва - Киев - 2018 г.

В работе исследована задача коммуникации в среде технологий блокчейн. В первую очередь, рассмотрена задачи синхронизации транзакций в парах несвязанных блокчейнов или похожих систем. Частным случаем такой синхронизации являются атомарные свопы (atomic swaps) – одновременный обмен криптовалютами между двумя участниками. В качестве примера, рассматриваются сети Bitcoin, Ethereum. Описывается алгоритм обмена.

Впервые вводится понятие DEP (decentralized exchange protocol) – протокола для хранения децентрализованного списка заявок на обмен (orderbook) и синхронизации процесса обмена криптовалютой. Как пример, описана спецификация протокола Swap DEP.

Исследованы реализации конкурентов. Рассмотрены варианты развития идеи децентрализованных обменов. Рассмотрены варианты построения биржи на базе протокола с дополнительными функциями: сортировка и автоматическое одобрение обменных заявок.

Содержание

1 Введение

В 2008 году Сатоши Накамото впервые была предложена надежная система децентрализованных электронных денег[1]. В его прототипе используется блокчейн – неизменяемая цепочка состояний, которая хранится одновременно на компьютерах всех участников сети. В этой цепочке содержится история всех транзакций валюты биткоин.

К 2018 году индустрия блокчейн выросла до 250 млрд долларов[2]. Существует много проектов, в той или иной мере использующих блокчейн или другие варианты децентрализации.

Интересен вопрос связывания (коммуникации) между собой различных блокчейнов. Простейшим случаем полезного связывания будет обмен внутренними валютами, например, биткоин и эфир (Bitcoin[3] и Ethereum[4]). Один из алгоритмов для такого обмена называется атомарным свопом (atomic swap[5][6][7]).

В работе описан Swap DEP (decentralized exchange protocol) – протокол для децентрализованной организации атомарных обменов между блокчейнами, хранение, синхронизация распределенного списка заявок – ордербука (orderbook). Раздел 2 – это введение в набор используемых технологий. В разделе 3 описана формальная спецификация протокола Swap DEP, а также пример реализации на JavaScript – swap.core[8]. Далее, в разделе 4, обсуждаются вопросы ограничений протокола и возможностей для дальнейшего развития.

Реализация протокола Swap DEP уже опубликована в тестовом режиме по адресу `https://swap.online`.

2 Используемые технологии

2.1 Bitcoin

Биткоин[1] – это распределенная платежная сеть, peer-to-peer версия электронных денег (то есть такая, где нет единого центрального сервера). Все транзакции сохранены в блокчейн – распределенную базу данных, дублированную на компьютерах всех участников сети. Биткоин – одно из решений для задачи "византийского консенсуса" [9][10].

Суть задачи "византийского консенсуса" – в возможности стабильной работы сети даже при условии наличия в ней ненадежных участников. Биткоин решает эту проблему с помощью подтверждения вычислением (Proof-of-Work). Во-первых, все платежные транзакции криптографически подписываются отправителем, а во-вторых, участники сети проверяют валидность подписей, и заявляют о случаях подделки или двойных трат. Если возникает конфликт, то учасники "голосуют" с помощью вычислений. Выигрывает та сторона, которая потратила больше вычислительных ресурсов в ходе такого "голосования".

Кроме этого, каждые 10 минут список одобренных транзакций замыкается в блок и фиксируется в блокчейне, что гарантирует его неизменяемость в дальнейшем.

Эта система была запущена 10 лет назад и с тех пор не была скомпрометирована, по сравнению с самыми надежными

централизованными финансовыми институтами, которые регулярно подвергаются взломам и утечкам конфиденциальных данных.

2.2 IPFS

IPFS *(InterPlanetary File System)* – это децентрализованное файловое хранилище. Компьютеры-участники сети обмениваются информацией о хранимых файлах и позволяют друг другу скачивать те, что есть у них в наличии. Скачанный файл кэшируется и его могут скачать другие учасники сети. Таким образом, нагрузка на центральное хранилище практически не растет при увеличении "спроса" на файл.

Похожим образом работает BitTorrent, но IPFS поддерживает вложенные файлы, то есть папки, и добавляет систему адресации с помощью хэша. Кроме того, IPFS спроектирована быть именно файловой системой, и хранит данные о существующих файлах глобально, а не локально в маленьких подсетях.

`QmYwAPJzv5CZsnA625s3Xf2nemtYgPpHdWEz79ojWnPbdG`

Рис. 1: Пример адреса в IPFS

В IPFS, идентификатором файла является хэш его содержимого. Хэширующий алгоритм не фиксирован: используется мультихэш. Это означает, что конкретный алгоритм задается первыми несколькими байтами. На данный момент, самый распространенный формат – SHA-256.

В IPFS есть возможность обмениваться сообщениями с "роем" (swarm)

напрямую с помощью пакета pubsub. Он делит сообщения по виртуальным "комнатам". На каждую можно подписаться и слушать сообщения оттуда, либо публиковать туда сообщение.

Например, во вставке ниже компьютер подписывается на комнату test room и отправляет туда сообщение. Он получит сообщение из этой комнаты прямо в консоль, как и все остальные, кто подписался на эту комнату.

```
tty1:     ipfs daemon --enable-pubsub-experiment &
tty2:     ipfs pubsub sub testroom &

tty3:     ipfs pubsub pub testroom "Hello, world"
tty2:     Hello, world
```

Рис. 2: Пример работы с комнатой в IPFS PubSub

2.3 Смарт-контракты

Смарт-контракт – специальный аккаунт в сети Ethereum, который контролируется заранее скомпилированным машинным кодом. Создатель контракта может запрограммировать произвольную логику, но после отправки в сеть изменить поведение контракта будет невозможно. Контракт может хранить произвольные данные, производить расчеты или пересылать средства в соответствии со своей внутренней логикой.

```
contract Mortal {
    /* Define variable owner of the type address */
    address owner;

    /* This function is executed at initialization and sets the contract owner */
    function Mortal() { owner = msg.sender; }

    /* Function to recover the funds on the contract */
    function kill() { if (msg.sender == owner) selfdestruct(owner); }
}

contract Greeter is Mortal {
    /* Define variable greeting of the type string */
    string greeting;

    /* This runs when the contract is executed */
    function Greeter(string _greeting) public {
        greeting = _greeting;
    }

    /* Main function */
    function greet() constant returns (string) {
        return greeting;
    }
}
```

Рис. 3: Пример одного из простых смарт-контрактов[11]

2.4 Atomic Swap

Atomic Swap – (атомарные свопы) алгоритм обмена монетами двух разных блокчейнов, который гарантирует атомарность операции, т.е. возможны

только два исхода: обмен произойдет полностью и по заранее оговоренной пропорции, либо не произойдет вовсе.

Ниже представлено описание одного из алгоритмов атомарного свопа[5]

```
A выбирает случайное число x
A создает TX1:
    "Отправить w BTC к <публичный-ключ-B>
    если (x от H(x) известно и подписано B)
    либо (подписано A и B)"
A создает TX2:
    "Отправить w BTC из TX1 к <публичный-ключ-A>,
    замороженные на 48 часов в будущее, подписано A"
A отправляет TX2 к B
B подписывает TX2 и возвращает A
1) A публикует TX1 в сеть.
B создает TX3:
    "Отправить v альт-коинов к <публичный-ключ-A>
    если (x от H(x) известно и подписано A)
    либо (подписано A и B)"
B создает TX4:
    "Отправить v альт-коинов из TX3 к <публичный-ключ-B>,
    замороженные на 24 часа в будущее, подписано B"
B отправляет TX4 к A
A подписывает TX4 и отправляет обратно B
```

2) B публикует TX3 в сеть

3) A тратит TX3, публикуя x

4) B тратит TX1 используя x

Алгоритм является атомарным (с тайм-аутом).

Если процесс нарушен, он может быть обращен вспять в любой момент.

До 1:

    Ничего не было опубликовано в сеть,

    поэтому ничего не происходит

Между 1 и 2:

    A может использовать TX2 через 72 часа

    чтоб вернуть свои деньги

Между 2 и 3:

    B может получить возврат через 24 часа.

    У A теперь есть 24 часа чтоб получить свой возврат.

После 3:

    Транзакции одобрены обоими.

- A должен потратить транзакцию в течение 24 часов

  иначе B может потребовать возврат и оставить свои деньги

- B должен потратить транзакцию в течение 72 часов

  иначе A может потребовать возврат и оставить свои деньги

Для безопасности, оба должны завершить процесс с большим запасом.

Существуют другие предложенные варианты обменов[7][6][12].

3 Swap Decentralized Exchange Protocol

Swap DEP – это протокол для децентрализованных обменов криптовалютами между блокчейнами. Список заявок на обмен и процесс обмена синхронизируются через комнату IPFS PubSub Room. Обмены происходят по алгоритму atomic swap.

3.1 Спецификация Swap DEP

3.1.1 Хранение и поиск ордеров

Введем понятия:

- Ордербук – (от англ. *orderbook*) список актуальных заявок на обмен.
- Ордер - заявка на обмен.

Перед тем, как совершить обмен, необходимо найти партнера, который согласен обменяться по такому курсу. Для этого в протоколе используется хранилище IPFS. В определенную комнату IPFS PubSub Room участники посылают желаемые обменные курсы и количества монет, формируя таким образом распределенный список заявок – ордербук.

В протоколе Swap DEP ордербук не существует в едином исполнении, а распределен между всеми слушателями комнаты. Каждый участник, подключаясь к комнате, посылает сигнал. В ответ каждый из слушателей комнаты отправляет ему свой список ордеров. При создании нового ордера или при его обновлении участник оповещает об этом комнату.

Таким образом, общий ордербук хранится у всех участников, и только у них – в IPFS PubSub Room отправляются только изменения списка.

Ордер кодируется в JSON. В полях JSON объекта содержится информация о валютах и количестве на обмен.

```
{
  "buyCurrency": "ETH",
  "sellCurrency": "BTC",
  "buyAmount": 7.0,
  "sellAmount": 1.0
}
```

Рис. 4: Пример формата заявки на обмен

3.1.2 Обмен

Когда участник (Алиса) находит заявку с подходящими курсом и количеством, она отправляет запрос на обмен в специальном формате. Создатель заявки (Боб) должен либо одобрить, либо отклонить запрос. Количество запросов к каждой заявке не ограничено – Боб может выбрать тот запрос, который его устраивает. Одобрение заявки отправляется в ту же комнату в специальном формате.

Возможна ситуация, когда у Алисы хранится неактуальная заявка, например, если Боб не в сети. Тогда обмен произведен быть не может, так как для обмена требуется активное участие обеих сторон. Возможна автоматизация процесса, и это обсуждается в разделе 4.

Далее, шаги обмена могут отличаться, в зависимости от конкретно

выбранной пары валют. В разделе 3.2 к реализации протокола swap.core.js приведен пример: обмен биткоин либо подобной валюты на эфир (Ethereum, ETH).

В зависимости от выбранного алгоритма, обмен будет проходить разным образом, но процесс обмена синхронизируется набором служебных сообщений, поэтому важно, чтоб мнение об алгоритме обмена у обеих сторон совпадало. Для этого позднее в разделе 4 обсудим версионирование протокола.

3.2 Реализация Swap DEP – swap.core

Swap.core[8] – реализация протокола Swap DEP на JavaScript. Программа состоит из нескольких частей: синглтон `SwapApp`, сервисы `SwapRoom`, `SwapOrders`, `SwapAuth`, классы `Flow`, `Swap` и их наследники `swap.flows`, `swap.swaps` – конкретные алгоритмы обменов.

`SwapApp`, `SwapRoom`, `SwapOrders`, `SwapAuth` реализуют работу по поиску ордеров и связи с сетью. Процессом обмена управляют классы `swap.flows`, `swap.swaps`.

Рассмотрим, например, один из классов `swap.flows`:

```
class BTC2ETH extends Flow {
    _getSteps()
}
```

Функция вернет список шагов, необходимых для завершения обмена:

1. Ожидать сигнала `swap sign`
2. Сгенерировать секретный ключ и передать в `submitSecret(SECRET)`
3. Если баланс недостаточен, пополнить и проверить еще раз `syncBalance()`
4. Отравить BTC для обмена, зашифрованные секретом `SECRET`, отправить сообщение `create btc script`
5. Ожидать сообщения `create eth contract`
6. Вывести свои деньги и отправить сообщение `finish eth withdraw`

4 Преимущества, ограничения и дальнейшее развитие протокола Swap DEP

4.1 Преимущества и ограничения

Основное преимущество – надежность этого решения. Деньги не приходится передавать третьей стороне, которая могла бы украсть или потерять их.

Далее, благодаря тому, что используется IPFS, обмен невозможно заблокировать, пока есть хотя бы два пользователя, желающих пользоваться системой.

Благодаря тому, что используется блокчейн, нет необходимости в лишних проверках и деньги с биржи пользователь может вывести сразу. Строго говоря, он и не вводил их, потому что кошельки существуют только у него на компьютере, а не на сервере.

С другой стороны, каждое из этих решений влечет за собой отрицательные последствия:

- Высокие комиссии из-за того, что все подтверждения происходят в системе блокчейн. Комиссии не зависят от объема, а это означает, что обменивать небольшими суммами будет чрезвычайно невыгодно. Тем не менее, пока что не стоит переживать из-за этого: стоимость транзакций в сетях биткоин и эфир на данный момент на уровне 0.1-0.5$[13].

- Принципиальная необратимость транзакций. Пользователь имеет полную власть над своими деньгами, и несет всю ответственность. Если он потеряет деньги, возместить будет некому. Эта проблема решается понятным и простым интерфейсом, который не дает возможности ошибиться.

- Обновления протокола для общения придется навязывать всему сообществу, либо ввести дополнительное поле во все сообщения, указывающее номер версии протокола.

4.2 Дальнейшее развитие

Отталкиваясь от предыдущих параграфов, видно, что основная проблема заключается в медленном и дорогом блокчейне. К счастью, существует решение для облегчения стоимости транзакций, названное Lightning Network[14].

Lightning Network[14] – это система второго уровня, то есть существующая поверх основного блокчейна Bitcoin. Она оперирует теми же монетами, но, вместо валидации в блокчейне, монеты замораживаются в "каналах" до востребования. Замороженные монеты можно пересылать в обе стороны канала, и даже за его пределы – сеть Lightning образует плотный граф, где между каждыми двумя участниками найдется путь.

С помощью LN появляется возможность реализовать автоматическое исполнение заявок и частичное исполнение заявок.

4.2.1 Автоматическое исполнение заявок

Положим, Алиса оставила компьютер запущенным, но не может подтвердить обмен. Тогда робот с заранее настроенными правилами может принимать или отклонять входящие запросы.

Точно так же робот на компьютере Боба может мониторить список заявок, и увидев подходящую, отправить запрос Алисе.

Одна из проблем с автоисполнением заявок заключается в том, что коммуникация в системе не мгновенная, поэтому возможна ситуация, когда два участника одновременно запрашивают одну заявку. Для того, чтоб решить такую проблему, будем следить за целостностью системы на уровне каждого отдельного участника. Например, если две заявки были отправлены одновременно, актуальной считается та, что пришла первой.

Кроме того, можно ввести понятие заблокированного баланса, что означает, что если денег на счету достаточно, то можно одобрить и провести обе заявки на обмен.

4.2.2 Частичное исполнение заявок

Если Алиса готова обменять 100 биткоинов, но на рынке есть только двое желающих (Боб и Чарли) купить по 50 биткоинов каждый, то в текущей системе обменяться они не смогут.

Два варианта решения этой проблемы:

- Ордер с плавающей суммой - Алиса установит, что желает продавать от 50 до 100 биткоинов за раз
- Частичное исполнение заявки - вначале Боб купит половину монет, и заявка Алисы уменьшится до желаемых 50 монет, и тогда Чарли сможет ее выполнить целиком

4.2.3 Децентрализованные обменники без атомарных свопов

Есть альтернативные способы реализовать распределенный обмен.

Первый способ – биржа на смарт-контрактах, и такие биржи работают только с ERC20-токенами (виртуальные монеты поверх блокчейна Ethereum).

Второй – логика биржи записана в смарт-контракте, но оперирует она реальными криптовалютами с помощью специального коннектора (gateway). Однако этот коннектор потенциально уязвим, так как не поддерживается ни одним из участвующих блокчейнов.

Более подробное исследование этого вопроса есть по ссылке[15].

4.2.4 Финансовые инструменты

С помощью комбинации атомарных свопов и LN можно собрать современные финансовые инструменты на блокчейне. В качестве примера можно привести торговлю с плечом или бинарные опционы. Такая система была бы надежнее привычных бирж, и при этом дешевле.

5 Заключение

В работе решена задача хранения осмысленных данных в среде технологий блокчейн. С помощью децентрализованного хранилища IPFS построен распределенный ордербук. Используя IPFS как средство коммуникации, участники могут надежно обмениваться криптовалютой с помощью алгоритма атомарных свопов. Весь процесс поиска и обмена формализован в виде протокола Swap DEP.

Реализация протокола Swap DEP уже запущена в тестовом режиме по адресу `https://swap.online`. Исследование в реальных условиях позволяет доработать спецификацию для следующих версий протокола.

Децентрализация гарантирует независимость от чужих ошибок. Взлом стороннего сервиса не должен влиять на сохранность личной информации или денег гражданина. С помощью Swap DEP возможна полная децентрализация финансовой системы – не только в рамках одной криптовалюты, а как комплекс интероперабельных систем.

В целом, децентрализация означает куда большую надежность: цена ошибки – это не отказ системы целиком, а ослабление мощности сети на доли процентов. В будущем, будут появляться все более децентрализованные системы, а существующие будут становиться распределенными. В связи с этим важно разработать грамотный и надежный механизм коммуникации между ними. Таким механизмом может стать Swap DEP.

Список литературы

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. https://bitcoin.org/bitcoin.pdf.

[2] "CoinMarketCap - Cryptocurrency Market Capitalization,"

[3] "Bitcoin official website," https://bitcoin.org.

[4] "Ethereum official website," https://ethereum.org.

[5] TierNolan, "Atomic swap algorithm," https://bitcointalk.org/index.php?topic=193281.msg2224949#msg2224949.

[6] B. Wiki, "Atomic cross-chain trading," https://en.bitcoin.it/wiki/Atomic_cross-chain_trading.

[7] "Smart-contract example: Trading across chains," https://en.bitcoin.it/wiki/Contract#Example_5:_Trading_across_chains.

[8] "Swap Core – Swap DEP implementation package," https://github.com/swaponline/swap.core.

[9] "Understanding Blockchain Fundamentals, Part 1: Byzantine Fault Tolerance," https://medium.com/loom-network.

[10] "Byzantine fault tolerance - Wikipedia," https://en.wikipedia.org/wiki/Byzantine_fault_tolerance#Byzantine_Generals'_Problem.

[11] "Create Hello World Ethereum Smart Contract," https://www.ethereum.org/greeter.

[12] "Exchange protocol p2ptradex," https://bitcointalk.org/index.php?topic=91843.0.

[13] "Bitcoin fee estimate," https://bitcoinfees.earn.com/.

[14] "Lightning Network - Scalable Off-chain instant payments," http://lightning.network.

[15] "Are the popular Bitcoin/Altcoin exchanges really decentralized and safe?," https://wiki.swap.online/.