Аннотация

Философские аспекты функционирования современного блокчейна

1 Введение

Для кого эта статья? Наверно для ИТ специалистов, не обладающих глубокими познаниями в области разработки высоконагруженных систем, криптографии, но которым хочется понять куда развивается технологии блокчейна и/или уже необходимо делать выбор по применению блокчейна, его развёртыванию и эксплуатации. Чего в этой статье нет: экономической модели существования блокчейна, реклам, рекомендаций по применению конкретного блокчейна.

Специалистов в данной области очень мало, те кто разбирается в этих вопросах на 300% заняты в различных криптопроектах. Попробуем изложить простым языком не очевидные вещи. Авторов написать данную статью побудило желание подготовить комьюнити к некоторым идеям, которые в настоящее время непонятны, неочевидны, провокационны и даже опасны.

Простым языком, не очевидные вещи.

2 Обзор

Что такое блокчейн? Это набор, в первую очередь, взаимоувязанных технологий, которые одновременно реализуются в разных элементах системы. При этом технологии выстроены таким образом, что обеспечивают информационную, алгоритмическую, и криптографическую целостность системы, не позволяя отдельным индивидам или группам ее разрушить, в определенных случаях даже включая создателей.

Авторы придерживаются следующей классификации блокчейнов:
приватный - блокчейн работу которого могут нарушить создатели системы;
корпоративный - блокчейн, в котором существуют как минимум два типа пользователей. Привилегированные пользователи, которые “условно” доверяют друг другу и участвуют в работах по поддержанию функционирования блокчейна и, возможно, получающие за это вознаграждение. А также непривилегированные пользователи, которые пользуются существующим функционалом блокчейна, но не участвуют в поддержании его работоспособности;
публичный - блокчейн, где все пользователи равнозначны, нет доверия никому, и все участники системы могут оказывать одинаково деструктивные воздействия на блокчейн.

Возвращаясь к архитектуре, можно выделить основные части:

а) база данных;
б) пиринговая сеть;
в) типы данных и типы транзакций;
г) консенсус;
д) блок;
е) кошелёк.

Почему в перечне нет криптографии? Потому, что ее невозможно выделить как некую законченную функциональную подсистему. Если обратить внимание на современное развитие криптографии и информационных систем, можно увидеть что простое применение криптографии для шифрования данных, их имитозащиты или электронной подписи сходит на нет. Теперь криптография, ее алгоритмы, механизмы, протоколы - разрабатываются под конкретный профиль информационной системы, и чем плотнее интеграция информационных алгоритмов обработки

информации с криптографическими, тем более защищённой и устойчивой она становиться. Эти веяния дошли уже до международных организаций, это можно увидеть на примере процесса разработки и стандартизации протокола TLS 1.3. В тоже самое время, блокчейн уже в полной мере реализует эту стратегию.

2.1 Базы данных

Рассмотрим базу данных, я думаю ни для кого большим секретом не является, то что первая криптовалюта Bitcoin [1] не использовала в своей работе базу данных для хранения состояния счетов на кошельках (wallet). Для того, чтобы узнать возможен ли перевод денежных средств с одного кошелька на другой, нужно было решить задачу по поиску последней транзакций на кошелек и вычислить возможность перевода (см. рис. 1).

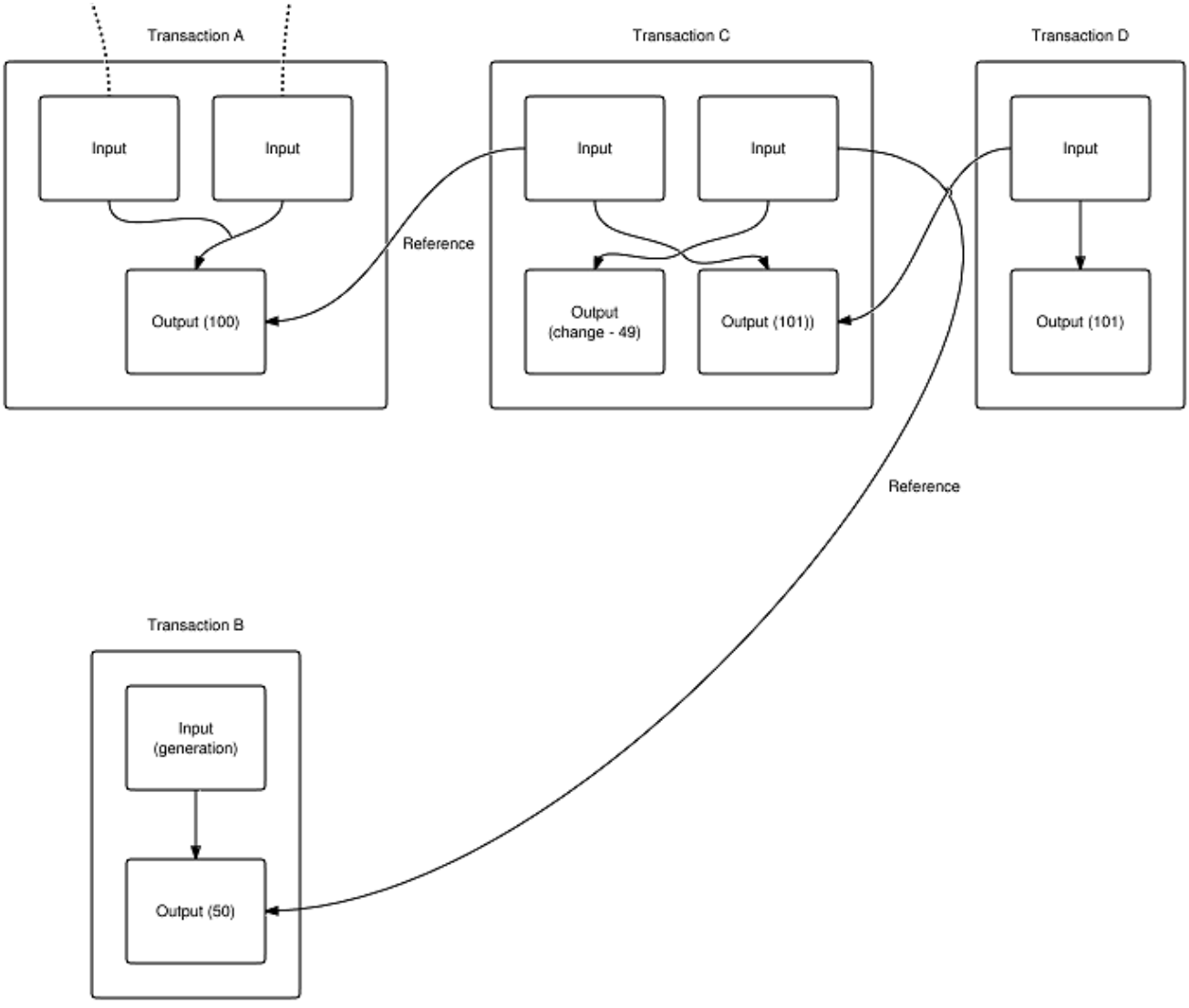

Рис. 1. Схема транзакций в сети Bitcoin

Многие современные блокчейны отказались от схемы вычисления отслеживания истории транзакций и хранят текущее состояние средств на кошельках в выделенной базе данных, например, PostgreSQL. Узел при синхронизации, получая новые блоки, обрабатывает их, выделяя транзакции. Находит в базе данных запись описывающую текущее состояние счета кошелька и изменяет его в соответствии с полученной транзакцией. Некоторые создатели идут дальше

и применяют три разных базы данных для хранения блоков, состояния кошельков и пула неподтвержденных транзакций.

На наш взгляд, существуют две крайности: первая - это блокчейн Bitcoin, в котором нет базы данных для хранения состояния кошельков, что приводит к большим сложностям при анализе истории транзакций в блокчейне; и вторая - использование нескольких баз данных, для хранения разных типов данных, что приводит к сложности их конфигурации при развёртывании и функционировании блокчейна.

Оптимальное решение - лёгкая база данных для хранения изменений кошельков, например, leveldb, и ее отображение в оперативную память - оперативная база данных в виде контейнеров, расположенных в оперативной памяти.

2.2 Пиринговая сеть

Пиринговая сеть и способы ее построения - это одна из основных задач, которую нужно решить программистам при создании собственного блокчейна, это основа децентрализованной сети. Пожалуй, наилучшим протоколом, на текущий момент времени, для пиринговой сети является протокол devp2p сети Etherium, работу которого можно кратко описать как следующую последовательность действий.

1) Формирование списка узлов. На основании ограниченного набора узлов, находящихся у программы блокчейна, формируются запросы для получения расширенного списка узлов по UDP протоколу.

2) После получения ответной информации программа блокчейна начинает устанавливать TCP соединения с узлами и выкачивать блоки блокчейна.

3) Программа блокчейна периодически начинает рассылать multicast UDP пакеты для обнаружения узлов сети блокчейна, находящихся рядом.

Интересующийся читатель сможет найти более подробное описания в ethereum/wiki. Это, пожалуй, самый совершенный протокол пиринговой сети известный авторам. Несмотря на свою сложность и совершенность, протокол не лишён недостатков, которые проистекают из его достоинств. Использование UDP multicast может привести к DDoS атаке, а доверие соседним узлам сети иногда приводит к блокировке синхронизации блокчейна. Авторы самолично наблюдали, как в тестовой сети “Rinkeby” блокчейна узлы в какой-то момент времени отставали на 138 блоков, и синхронизировать блокчейн удавалось только после ручного вмешательства.

Альтернативным решением для построения пиринговой сети является подход блокчейна Graphene(Bitshares/EoS), когда основные пиринговые узлы записываются в исходный код, а в процессе работы пирингового протокола из сети выкачиваются новые адреса узлов пиринговой сети. Этот подход так же имеет явные недостатки: блокировка основных узлов пиринговой сети приведёт к блокировке работы блокчейна. Заблокировать узлы можно несколькими способами, например, административно, или проведя DDoS атаку.

2.3 Консенсус

Консенсус PoW – самый первый тип консенсуса, реализованный в блокчейне валюты Bitcoin. Консенсус отличается невысокой скоростью закрытия блока и малой скоростью транзакций. Скорость транзакций валюты Bitcoin составляет не более 5 транзакций в секунду (при среднесуточных измерениях). Формирование блока блокчейна при консенсусе PoW требует значительных

вычислительных ресурсов. Причем чем больше у участника консенсуса вычислительных ресурсов, тем выше вероятность для него сформировать блок, что приводит к непроизводительной (бессмысленной) вычислительной гонке между участниками консенсуса [2].

Консенсус Proof-of-Stake (PoS) и его вариации DPoS, LPoS были предложены, чтобы решить проблемы консенсуса PoW, связанные с высокими вычислительными издержками и малой скоростью закрытия блока транзакций. Несмотря на высокую скорость закрытия блока и низкие требования к аппаратным ресурсам, у алгоритма PoS есть недостатки. Проблемой PoS является централизация монет (ресурсов системы). Пользователь блокчейна, имеющий максимальный объем ресурсов системы, получает еще больше монет за оказание услуг по подтверждению блока. Следовательно, количество узлов, участвующих в консенсусе, будет эволюционно уменьшаться [2].

Консенсус pBFT – еще одна альтернатива алгоритму консенсуса PoW. В мире предложено несколько реализаций консенсуса pBFT, один из лучших – «Honey Badger» [4]. Как показано на рис. 2, реализация pBFT консенсуса работает тем лучше, чем меньше количество узлов, участвующих в консенсусе.

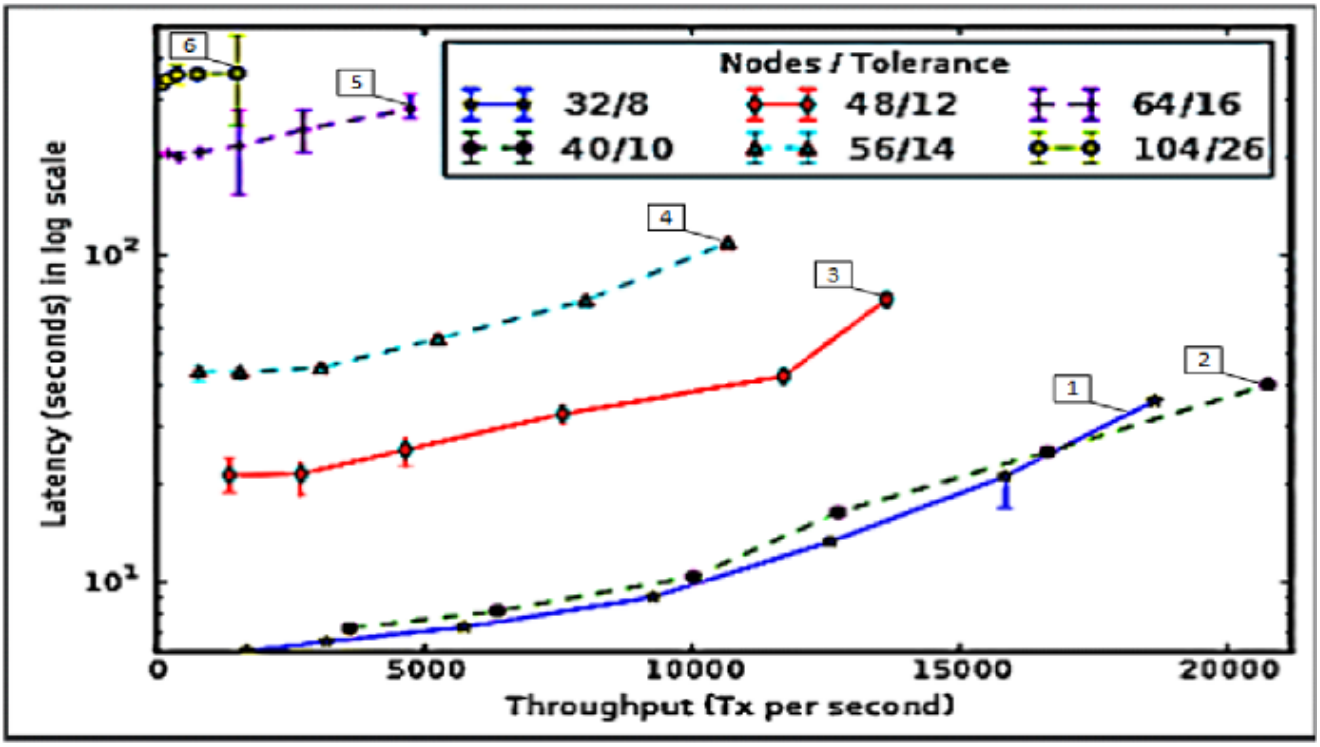

Рис. 2. Графики зависимости времени задержки формирования блока от скорости поступающих транзакций

На рисунке 2 показан график зависимости времени задержки формирования блока от скорости поступающих транзакций, где Nodes/Tolerance – соотношение общего количества узлов, достигающее консенсуса, и числа узлов, не достигающих консенсуса. Кривая 1 показывает изменение времени закрытия блока в зависимости от скорости поступающих транзакций для 32 узлов, кривая 2 – для 40 узлов, кривые 3-6 – для 48, 56, 64 и 104 узлов соответственно. Наиболее эффективно консенсус работает при количестве узлов, не превышающем 40, скорость транзакций для такого числа узлов достигает $2 \cdot 10^4$ транзакций в секунду, при этом время закрытия блока не превышает 40 секунд. Если количество узлов превышает 60, кривые 5 и 6, то скорость транзакций для такого числа узлов не превышает $0.5 \cdot 10^3$ транзакций в секунду, при этом время закрытия блока превышает 100 секунд. Время консенсуса для 104 узлов достигало 6

минут.

3 Анализ развития блокчейнов

В предыдущем разделе мы постарались обобщить основные проблемы, с которыми сталкиваются разработчики блокчейна, опустив несколько незначительных проблем, например, выбор оптимального размера блока, или схемы перевода *fee* с транзакции.

На наш взгляд, есть четыре ключевых элемента технологии блокчейна, которые будут трансформироваться в ближайшие пять лет, это:

а) консенсус;
б) пиринговая сеть;
в) шлюзы с существующими информационными системами;
г) смарт-контракты.

Мы умышленно не включили смарт-контракты в перечень элементов из которых состоит блокчейн, так как считаем, что в текущем виде смарт-контракты не применимы в долгосрочной перспективе, язык описания смарт-контрактов очень ограничен, попытки расширить функциональность смарт-контрактов приводят к неоднозначным результатам. Например, пользователи сети Etherium потеряли более 1 млн. долларов при попытке купить токены EoS, так как смарт-контракт токена EoS не был рассчитан на покупку его через кошельки бирж.

Смарт-контракты требуют отдельного рассмотрения, но если смарт-контракт не может существовать без блокчейна, то блокчейн - самодостаточен.

3.1 Консенсус

Значительные усилия сообщества будут направлены на поиски более совершенных алгоритмов консенсуса, а также их математического обоснования. Например, алгоритм DPoS подобных блокчейнов, применяемых в Graphene, математически не обоснован. Нет строго математического доказательства корректности работы алгоритма DPoS.

Если взять классификацию блокчейнов (приватный, корпоративный и публичный), то можно утверждать - для приватного блокчейна непринципиально совершенствовать алгоритмы консенсуса, есть хорошие алгоритмы, класса BFT, которые эффективно и быстро работают на количестве узлов, не превышающих 100 (рис. 2).

Консенсус в публичном блокчейне является очень критичной задачей, но помимо консенсуса PoW на текущий момент времени ничего не придумано. Попытка перевести на алгоритм консенсуса PoS блокчейн Etherium на момент написания статьи закончились неудачей. К тому же возникает вопрос, а будет ли блокчейн, основанный на технологии PoS публичным? На взгляд авторов, сегрегация пользователей блокчейна на “доверенных”, участвующих в консенсусе, и обычных, автоматически делает его корпоративным.

Корпоративный консенсус - пожалуй самое перспективное направление для развития консенсуса. В существующих консенсусах для корпоративных блокчейнов не достаёт следующего:

1) увеличения числа узлов, которые могут участвовать в консенсусе, количества в 100 узлов (см. рис. 2) явно не хватит;
2) увеличение скорости подтверждение блока, в том числе для решения задачи кассового обслуживания (высокоскоростных платежей), скорость подтверждения блока 10-15 секунд;
3) динамическое отслеживание попыток атаки изнутри на консенсус.

Как решить подобную задачу? Мы сделали попытку решить проблему консенсуса разработав собственный алгоритм sdBFT. Ключевые элементы алгоритма изложены далее в статье. Детали отражены в следующих документах (тут будут ссылки).

3.2 Пиринговая сеть

Можно разработать совершенный консенсус, но если скорость распространения неподтвержденных транзакций до узлов, принимающих консенсус, будет превышать 10 секунд, то можно забыть о решении задач кассового обслуживания (высокоскоростных платежей). Для решения задачи с учётом первого и второго требования к консенсусу необходимо увеличить число узлов и увеличить скорость подтверждения блока. Чем больше узлов сети, тем более сложная пиринговая сеть. Чем сложнее пиринговая сеть, тем медленнее распространяются в пиринговой сети транзакции. Какое решение можно предложить для ускорения пиринговой сети? Самое кардинальное - не использовать пиринговую сеть. Но тогда как распространять большие объёмы данных - синхронизацию блоков блокчейна? На наш взгляд, путём создания гибридной сети, разделив информацию, циркулирующую в блокчейне на два типа:

а) оперативную, к которой относятся транзакции, которые нужно включить в блок, а также информация о создании нового блока, служебные данные пиринговой сети;
б) архивную, к которой относятся созданные блоки блокчейна.

Оперативные данные не должны передаваться посредством пиринговой сети. Для передачи этих данных нужно использовать схему схожую со схемой динамической маршрутизации. По большому счёту узлы блокчейна можно представить как узлы некой сети, в которой по определённому закону можно построить сбалансированное B-дерево.

B-дерево — структура данных, дерево поиска. С точки зрения внешнего логического представления, сбалансированное, сильно ветвистое дерево. Сбалансированность означает, что длина любых двух путей от корня до листьев различается не более, чем на единицу.

Архивные данные передаются с использованием классической пиринговой схемы передачи данных. Тем самым мы из подсистемы пиринговой сети создаём гибрид пиринговой сети и сети с явными маршрутами для передачи оперативной информации.

3.3 Шлюзы с существующими информационными системами

Когда создавался первый блокчейн, никто не задумывался о его популярности и не прорабатывал механизмы по сопряжению блокчейна с существующими автоматизированными системами. Для большинства читателей очевидно, что существующие, совершенствующиеся десятилетиями автоматизированные системы, не уйдут в прошлое после появления блокчейна, но многие владельцы данных систем уже задумываются о возможности интеграции своих систем с блокчейном и, на взгляд авторов, подобные работы в ближайшее время будут очень актуальными, в особенности двунаправленные шлюзы.

3.4 Смарт-контракты

Как уже писалось выше, существующий подход к созданию смарт-контрактов в текущим их понимании сообществом, по мнению авторов ошибочен. Наша позиция, это создание DRPC сервиса внутри блокчейна, когда пользователь не пишет собственный код смарт-контракта, но

может вызывать распределенное исполнение кода, например, код создания нового токена, с именем *"name"*, количеством *"assetAmount"* на кошельке *"WalletAddress"*.

```
MsgPack::object message{

{ "type", "CreateToken" },

// string
{ "name", name },

// string amount token
{ "fundsAmount", assetAmount},

// wallet address
{ "user",
MsgPack::binary(WalletAddress.begin(),WalletAddress.end()) },

// array<unsigned char, 64>
{ "signature", MsgPack::binary(signature.begin(), signature.end()) }

};
```

4 Дальнейшая эволюция

Современные блокчейны будут эволюционировать в направлении PoS, совершенствования алгоритмов консенсуса класса BFT, проблем масштабируемости, и при этом, конечно, придётся решить множество сложных, прикладных инженерно-математических задач. Сформулируем основные направления, по которым будет эволюционировать PoS. Для начала опишем все системы и подходы по ее построению.

4.1 Genesis block

Что должно содержаться в нем, чего там не должно быть, возможные атаки на излишне сложный *genesis block*. Если посмотреть на структуру первого генезис блока [5] , так называемого нулевого блока блокчейна Bitcoin, можно заметить, что он минимален по своему содержанию. Время идёт и появляются идеи размещать в *genesis block* иную, важную для существования блокчейн. Один из первых PoS блокчейнов NxT [6] через *genesis block* реализовал первичное распределение всех своих токенов. Iroha [7] - размещает информацию о своих пиринговых узлах, и т.д.

Для *genesis block*, как ни странно, наибольшую угрозу несут его создатели. Основная угроза с их стороны, это когда владельцы НОД, указанные в нулевом блоке на каком-то эволюционном этапе, сформируют альтернативный блокчейн, который отменит/ перепишет основной блокчейн. Т.е. НОДы, записанные в *genesis block*, должны быть выведены из эксплуатации, как самая эффективная мера борьбы с мошенничеством создателей, а для этого должны быть предусмотрены механизмы исключения НОД и альтернативные каналы распространения актуальных НОД в сети. Сейчас мы живём в рамках философии, что истина находится только в блокчейне и все то

ложь, что не находится в блокчейне. Это, в нашем понимании, ошибочная позиция, мир как всегда сложнее ...

Тот, кто начал развивать блокчейн, имеет преимущества перед всеми остальными, включая комьюнити и сообщество. Во-первых, может быть организована атака на отказ в обслуживании, во-вторых, на перехват и создание альтернативной ветки блокчейна. Для этого не нужно создавать полноценную альтернативную ветку, достаточно переключать на неё новых пользователей системы. Со временем основной блокчейн начнёт терять пользователей, так как они в первую очередь ориентируются на блокчейн, с *genesis block* в начале. Когда новый пользователь обращается на корневые НОДы, его могут повести по той ветке блокчейна, по которой захотят. Все остальные пользователи основного блокчейна со временем умрут, так как количество узлов и пользователей начнёт уменьшаться, что в конечном итоге закончится тепловой смертью. Для перезапуска блокчейна потребуется доработать код, который позволит использовать иные узлы блокчейна, т.е. потребуется подгружать список из вне.

Возникает интересный вопрос: подписывать или не подписывать *genesis block*? Подписанный блок это правильно, но кто гарантирует сохранность / доступность ключа подписи? Такой вариант допустим в случае, если у системы есть владелец, а если его нет - не работает. Положим, что систему запускает комьюнити, и все активные участники развернули НОДы, сгенерировали закрытые ключи, дали информацию о себе для включения в *genesis block* ... Значит, они могут его и подписать. В нашем понимании, эта проблема сильна увязана с концептуальной проблемой усечения блокчейна. Но об этом чуть дальше.

4.2 Блокчейн

База данных, которая хранит информацию об изменениях. В общем случае это направленный набор блоков, имеющих сильные криптографические связи, которые позволяют однозначно и гарантированно определять факт модификации данных. Сам блок предназначен для хранения транзакций, верифицированных сетью блокчейна. Так как, общем случае, блок содержит переменное количество транзакций, и блок может достигать размеров и в 10 Мбайт, и в 100 Мбайт, то возникает задача оперативной работы с блоком, как с массивом информации, целостность которого можно проверить по неким контрольным точкам, без получения полной информации, содержащейся в блоке. В нашем понимании, это “скелет” блока, который содержит в себе:

а) заголовок блока, содержащий хеш предыдущего блока;
б) дерево Меркла, которое однозначно отображает набор транзакций размещённых в блоке;
в) кортеж голосов, эскортирующих НОД, принимавших участие в консенсусе этого блока;
г) утверждающая электронная подпись мастер НОДы над хешом закрываемого блока.

Перекрёстные проверки “скелета” блока могут однозначно подтвердить подлинность блока и однозначно связать с хранимыми в нем данными. Стоит отметить, что в отличии от классических блокчейнов, которые хранят входы и выходы транзакций, современный блокчейны начинают отказываться от такого механизма. Для своего времени, это было элегантное решение по организации системы быстрого поиска транзакций в процедуре проверки двойной траты. Современные компьютеры позволяют организовать более эффективную работу с данными, в т.ч. и поиском. Т.е. каждая НОДа в каждый момент времени хранит актуальный *state* блокчейна, т.е. блокчейн как база изменений, сворачивается в базу балансов. После формирования и получения следующего блока, происходит изменение *state* базы данных. Видится ряд оптимизационных задач:

а) периодическое сохранении и восстановлении *state* базы, который используется для быстрого запуска НОДы;

б) раздельное хранение “скелета” блока от содержимого блока, что будет сильно востребовано так называемыми "легкими"кошельками.

в) оперативный контроль целостности локальной копии блокчейна и *state* базы по “скелетам” блоков.

4.3 Усечение блоков

Концептуальная проблема бесконечного роста блокчейна, конечно, компенсируется экстенсивным развитием вычислительной техники, но особого смысла постоянно оперировать данными из большого временного интервала - нет. Если в концепции PoW решать такую задачу бессмысленно по причине индивидуализма в формировании каждого блока, то в концепции PoS, когда мы имеет консенсусную природу формирования блока, усечение блокчейна идеологически возможно.

Усечение блокчейна - это как маленькая смерть и последующее воскрешение, все эти процессы взаимоувязаны на *genesis block*. И каждый цикл усечения должен формировать новый *genesis block*, генетически связанный со всей историей блокчейна. Если ранее мы рассуждали о способах формирования *genesis block* и обратили внимание на порядок его подписи, который на первый взгляд кажется избыточным, то в комплексе с задачей формирования *genesis block* для нового цикла, такая специфическая работа с ним становится критически важной. Т.е., когда идёт усечение блокчейна, по факту идёт формирование нового *genesis block*, который однозначно должен быть подписан участниками (НОДами). Получается, что *genesis block*, как в нулевой точке, так и во всех в промежуточных точках блокчейна, имеет единый принцип формирования, что говорит о корректности данного логического утверждения.

Соответственно, при запуске проверяется подлинность электронных подписей у *genesis block*, и задачей корневых НОД, либо НОД пришедших им на смену, будет выпустить такой же *genesis block*, который содержит актуальные балансы активных кошельков (учётных записей) на некий момент времени t, так называемый *state* базы данных блокчейна. Он может быть любого объёма, так как происходит усечение блокчейна, который уже есть на всех НОДах, и пересылки по сети 100 Мбайт или 1 Гбайта не требуется. Возникает очень сложная, встречная проблема — проблема отсутствия НОДы. Промежуточный *genesis block* должен иметь подписи всех НОД, актуальных на момент времени t, но что делать если какие-то НОДы умерли, либо саботирую такую важную миссию? Видится целесообразным применять методологию консенсуса, в которой необходимо собрать квалифицированное большинство, а именно $\frac{2}{3}$ для выполнения операции усечения. Т.ее НОДы, которые не приняли участие, должны быть исключены из списка НОД автоматически, конечно сохраняя их балансы и позволяя им в дальнейшем стать обратно НОДой.

Такая чистка в чем-то даже полезна, так как она позволяет обеспечить эффективность консенсуса и пиринговой сети в следующем цикле функционирования блокчейна за счёт исключения“мёртвых” НОД. Такую процедуру даже можно назвать “Reflection point”.

Следующие аспекты должны быть учтены при построении алгоритма усечения:

а) принцип выбора номера блока, на котором начинается усечение блокчейна;

б) формирование блока, содержащего множество транзакций со всех непустых кошельков в адрес системного кошелька;

в) закрытие блока при достижении консенсуса между всеми НОДами в парадигме $\frac{2}{3}$ голосов

НОД, зарегистрированных в блокчейне;

г) формирование зеркального блока по отношению к предыдущему, в котором расходятся транзакции с системного кошелька на все кошельки;

д) формирование дополнительных транзакции в блок, которые регистрируют “живые” НОДы (участвовали в консенсусе предыдущего блока) как НОДы в новом *genesis block*;

е) когда блок закрывается, у блокчейна появляется новый *genesis block* для нового цикла блокчейна;

ж) блоки блокчейна, относящиеся к предыдущим циклам, не используются в оперативной работе системы, но могут храниться на “архивных” НОДах и использоваться в статистических задачах.

4.4 Peer net

Понятие транзакции сети блокчейна эквивалентно сообщению внутри пиринговой сети. Какие типы сообщения внутри сети блокчейна (пиринга) должны ходить? Если проводить аналогию с системами реального времени - архивные сообщения и оперативные (т.е.относящиеся к конкретному слайсу времени).

Сеть блокчейна условно можно разделит на две сети: первая - низкоприоритетная (пиринг), вторая - высокоприоритетная (консенсуальная сеть). В быстрой сети два типа сообщений: сама транзакция и анонс нового блока блокчейна. Анонсом нового блока может являться просто сообщение, либо информация о маршрутизации распространения нового блока, или это может быть "скелет"блока, на основании которого может быть получена маршрутная информация по распространению нового блока. Динамические маршруты в каждом раунде должны перестраиваться для максимально быстрой доставки транзакций и формирования очередного блока блокчейна узлами принимающими участие в консенсусе. Выбор остаётся за разработчиком и его профессионализмом. Любопытным решением может оказаться использование UDP в быстрой сети, но это потребует защиты UDP-фрейма от его повторного навязывания для противодействия атакам типа отказ в обслуживании. Это можно сделать путём увязки адресно-маршрутной информацией пакета и электронной подписью НОДы.

Получается, что в консенсуальной сети все взаимоувязано: консенсус, B-дерево, пиринговая сеть, и подчинённо одной цели, быстрому формированию и закрытию блока. Тем самым будет обеспечено “бессмертие” блокчейна. Как этого можно достичь?

Положим, что активные НОДы, т.е. ноды, участвующие в консенсусе, формируют B-Дерево состоящее из трёх уровней: вершина мастер-нода; второй уровень - эскортирующие ноды; третий уровень - все остальные НОДы. Требуется обеспечить гарантированную доставку анонса о новом блоке (“скелет блока”). При этом есть ограничения, мы не знаем, кто из НОД физически недоступен. Чтобы обойти эту проблему, строится дерево, где плоская вершина (второй уровень) состоит из НОД участвовавших в консенсусе, и данные НОДы должны гарантированно разослать анонс.

Есть альтернативный подход, НОДы не участвовавшие в консенсусе должны установить соединения со случайной НОДой участвующей в консенсусе. Тем самым будет обеспечено равномерное подключение НОД к эскорту, и после формирования блока обеспечена гарантированная доставка анонса до всех включённых НОД. Тем самым задача оперативного получения анонса нового блока перекладывается на наименее нагруженных участников сети, что является достаточно эффективным решением. Развивая данный подход, необходимо разгрузить мастер-ноду, чтобы она фактически являясь дирижером текущего раунда консенсуса, выполняла только одну

задачу - синхронизировать работу эскорта, все остальные, не важные соединения должны быть сброшены. Второй уровень выполнит задачу балансировки транзакций в сети и распределение закрытого блока по сети.

Существует проблема, как работать НОДе, в случае рассинхронизации ее *state* со *state* сети. Считаем, что ей нужно работать в любом случае, но быть по факту "плохой" НОДой, так как есть достаточно высокая вероятность того, что транзакции включаемые в блок будут удовлетворять текущему *state* "плохой" НОДы. В любом случае, при закрытии должен быть достигнут консенсус, и "плохая" НОДа на него повлиять никак не сможет.

Другая проблема, как узнать НОДе, кто в консенсусе. От любых НОД получить потенциально корректные "скелеты" блоков, выстроить их в целостную цепочку и актуализировать свой *state*, выкачивая информационную часть блока, начиная с конца блокчейна.

Для снижения накладных расходов на сеть блокчейна должно быть максимальное сокращение команд, сообщений, анонсов. Оперативная транзакция и сообщение о принятии нового блока. Все. Оперативная транзакция, это то, что должно попасть в блок. Если создавать систему оптимизированную под кассовые операции (сверхбыстрые платежи), то эти сообщения должны мгновенно доставляться до НОД, участвующих в консенсусе. 20 сек. - это максимальное время на закрытие блока.

Как они могут приниматься: они не должны распространяться по пирингу, никогда, они не должны попадать в какие-то пулы , очереди сложные, синхронизации или пересинхронизации, не должны выдаваться сообщения о принятии или непринятии транзакции . У них должно быть два маршрута: 1) отправка транзакции, 2) получение готового блока. Если в блок она попала, значит, она принята, если ее нет, значит, нужно ее повторить. Все остальное, это попытка уничтожить систему, попытка ее перегрузить.

Для более интеллектуального общения с клиентами, в перспективе можно ввести следящую / архивную НОДу.

Оперативные транзакции должны бегать не по пирингу, а по специальным маршрутам, которые должны вырабатываться специальным образом, таким, чтобы максимально сократить количество лишних пересылок внутри пиринговой сети, так как ценность этих транзакций стремится к нулю, до момента включения в блок.

Когда она в блоке - она ценна! До этого момента она мусор, и, если транзакция не попала в блок, клиент должен осуществить ее повторную отправку. Можно рассуждать об интеллектуальном взаимодействии следящей ноды с кошельком, когда будет приходить сообщение с подтверждением о включении транзакции в блок, либо сообщение с кодами ошибки почему транзакция была отвергнута сетью.

Анонс о выработке нового блока должен очень быстро пролетать на все узлы, участвующие в консенсусе, и только в этом случае мы можем обеспечить эффективное формирование новых блоков .

4.5 О комиссии

Существует взгляд централизованных систем - с транзакции сформированной пользователем, в рамках своей бизнес логики, вычитать комиссию из транзакции. С точки зрения философии криптоанархизма, как высшей степени проявления свободы, можно взять только ту комиссию, которую указал пользователь, или отклонить транзакцию как не соответствующую интересам сети. Отталкиваясь от этой идеологии, в каждой транзакции должна быть указан комиссия, которую рассчитывает клиентский кошелёк (пользователь), а НОДы, принимающие транзакцию,

проверяют правильность расчёта комиссии, а именно, что она не меньше установленной. Тем самым подсистема учёта и распределения комиссии с транзакций проведёт суммирование всех комиссий от всех транзакций, включённых в блок, и сформирует дополнительную транзакцию, подписанную мастер-нодой о переводе всей комиссии на системный кошелёк или кошелёк мастер-ноды (в зависимости от некой бизнес логики). Стоит отметить, что в блокчейнах с быстрыми транзакциями отпадает необходимость в произвольных размерах комиссий, так как нет времени на фильтрацию/сортировку транзакций, включаемых в блок по сегрегационному признаку - размеру транзакции. В таких системах должен быть принцип - первым пришёл первый вышел (FIFO).

4.6 Описание консенсуса sdBFT

Необходимость создания собственного алгоритма консенсуса возникла при попытке создать блокчейн, отвечающий следующим требованиям к сети блокчейн:

1) Время создания нового блока не более 20 сек.
2) Тип сети блокчейн – корпоративный.
3) Общее количество узлов, которые могут принять участие в выработке консенсуса, может меняться от 10^3 до 10^4.
4) Высокая скорость транзакций – не менее 10^3 транзакций в секунду.
5) Реализация алгоритма блокчейна не должна требовать существенных вычислительных, по сравнению с блокчейнами PoW, мощностей.
6) Награда за поддержание блокчейна в работоспособном состоянии должна распределяться равновероятно между всеми узлами.

Авторы рассмотрели существующие блокчейны и алгоритмы консенсуса и пришли к выводу что подходящих под наши условия блокчейнов и алгоритмов консенсуса не существует, и решили разработать собственный алгоритм консенсуса stake distributed Byzantine Fault Tolerant (sdBFT), позволяющий увеличить на несколько порядков количество узлов сети, участвующих в достижении консенсуса, по сравнению с существующими алгоритмами семейства BFT [3], и существенно повысить скорость транзакций. Как мы этого добились? Давайте по порядку.

4.6.1 Владение доли

Нашей целью не было создание блокчейна, в котором любой желающий мог бы скачать дистрибутив и начать“майнинг” криптовалюты. Мы ставили перед собой задачу создать стабильную систему способную решать целевые задачи по переводу криптоактивов между участниками системы за кратчайшее время без недельного ожидания попадания транзакции в блок. Поэтому мы выделили два типа пользователя: пользователей, которые участвуют в работе системы и получают комиссии с транзакций и пользователей, целью которых является перевод средств с одного кошелька на другой.

Для того что бы стать пользователем, участвующим в работе блокчейна и выработке консенсуса, пользователь должен создать кошелек и перевести на данный кошелек определённую сумму денег. На наш взгляд, достаточно крупную в реальном эквиваленте. Пользователь формирует специальную транзакцию “regiester node”. В рамках данной транзакции проверяется сумма на кошельке, и, если количество средств достаточно, то эта сумму блокируются на кошельке (депозит). Информация о появлении (исключении) НОДы записывается в блокчейн, в том числе указывается IP адрес по которому будет доступна данная НОДа. Пока в балансе НОДы

заблокирован депозит, НОДа может принимать участие в процедурах консенсуса. НОДа может в любой момент сформировать транзакцию “unregiester node”, которая исключит ее из списка нод и разблокирует депозит.

4.6.2 Выбор участников консенсуса

Очевидно, см. рис.2, что все НОДы, зарегистрированные в системе, участвовать в консенсусе не способны. Поэтому нужно обеспечить выбор некоего множества НОД, которые будут принимать участие в создании блока, причём равновероятно. Как это сделать?

Единственной общей информацией для всех НОД является информация, содержащаяся в блокчейне. Поэтому мы предложили получать псевдослучайную последовательность, рассчитывая хэш от последнего принятого блока $\nu = H(H(d))$. Из полученной псевдослучайной последовательности получаются номера НОД, которые будут участвовать в консенсусе. Первую НОДу в этом списке мы называем Мастер НОДой, остальные из списка будут эскорт НОДами, а НОДы не попавшие в список будут пассивными НОДами.

В случае, если псевдослучайной последовательности не хватает, происходит вычисление хэш по формуле

$$\nu_n = H(H(d+n)) \tag{1}$$

где $n = 1, 2.....$

4.6.3 Последовательность выработки нового блока

Давайте рассмотрим последовательность создания нового блока. Пусть в некий момент времени пользователь формирует транзакцию I. Данная транзакция передаётся ближайшей НОДе, с которой связан данный клиент. НОДа может находится в одном из трёх состояний: пассивная, эскорт или мастер. Если нода пассивная, то она проверяет транзакцию и передаёт ее далее по пиринговой сети, пока транзакция не дойдёт до эскорт НОДы. Эскорт НОДа пересылает транзакцию мастер НОДе. Мастер-нода проверяет транзакцию и, если транзакция корректная, пересылает ее эскорт НОДам, а так же записывает транзакцию I в формируемый блок. Эскорт ноды приняв транзакцию I проверяют ее на корректность и записывают ее в формируемый блок. Данная последовательность действий повторяется до момента завершения блока, не более 1 минуты. После чего мастер-нода рассылает сообщение о завершении блока. Каждая эскорт НОДа рассчитывает хэш блока транзакций, электронную подпись хэша, и пересылает полученный хэш мастер-ноде.

Мастер-нода рассчитывает количество корректных, по ее мнению, электронных подписей. Если полученное число корректных подписей превышает 2/3 от общего значения эскорт-нод участвующих в консенсусе, то блок считается сформированным. В противном случае блок не формируется.

Мастер-нода записывает хэш предыдущего блока, необходимую заголовочную для блока информацию, транзакции, хэш дерева Меркла и список электронных подписей эскорт-нод, которые подтвердили данные транзакции, а также рассчитывает хэш блока и подтверждает его своей подписью.

Если в течении заданного времени новый блок не был получен, то НОДы начинают выбирать следующий набор эскортирующих НОД по формуле 1.

Строгое математическое описание алгоритма консенсуса представлено в нашей статье sumus.team/wiki/app1.

4.6.4 Комиссия

Все дети как дети -
Живут без забот,
(Счастливое детство...)
А Боб на диете -
Не ест и не пьёт,
(Бедненький мальчик...)
В копилку кладёт

Проблема мотивации НОД на добропорядочную работу по поддержке сети блокчейна заключается в "корректном" распределении комиссий с транзакций, можно решить несколькими способами. Очевидный способ: распределять с каждого блока комиссию между всеми НОДами или между НОДами, участвующими в работе сети. Наше решение более оригинальное - вся комиссия с транзакций закрытого блока переводится мастер НОДе. Почему именно так? С одной стороны, алгоритмически это проще, значит, надёжнее. В этому случае не нужно делать сложных вычислений при расчёте распределения комиссии между участниками консенсуса, нет необходимости отслеживать работоспособность всех НОД сети и решать схожие задачи, которые появляются при попытке сложного распределения комиссии между множеством НОД.

4.7 В качестве заключения

Технологии вокруг блокчейна развиваются очень быстро, предугадать что будет сделано, а тем более востребовано, очень сложно. Поэтому авторы статьи сделают все возможное от них, чтобы их блокчейн содержал максимум прорывных технологий. А на сколько это будет востребовано, ... решать в том числе и читателям.

Список литературы

[1] Satoshi Nakamoto (2009). "Bitcoin: A Peer-to-Peer Electronic Cash System". www.bitcoin.org . .

[2] BitFury Group (2015.09.13). "Proof of Stake versus Proof of Work". http://bitfury.com/content/5-white-papers-research/pos-vs-pow-1.0.2.pdf .

[3] Leslie Lamport, Robert Shostak, Marshall Pease (1982). "The Byzantine Generals Problem". ACM Transactions on Programming Languages and Systems. T.4, 3: 382–401 .

[4] Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, Dawn Song (2016). "The Honey Badger of BFT Protocols". https://eprint.iacr.org/2016/199.pdf .

[5] Bitcoin Wiki. Genesis block. https://en.bitcoin.it/wiki/Genesis_block .

[6] NxT Wiki. Genesis block. https://nxtwiki.org/wiki/Blocks_and_Blockchain .

[7] GitHub. Iroha. Genesis block. https://github.com/hyperledger/iroha/blob/master/example/genesis.block .