Тера протокол

Аннотация

В данной работе мы показываем, что сделав более строгим консенсус POW блокчейна, у нас появляются ответы на ранее не решенные вопросы:
Что будет если в консенсусе Накамото цепочки блоков выбирать не исходя из искусственной функции сложности, а исключительно из суммы произведенной работы. Приведет ли это к увеличению производительности сети. Можно ли построить сеть с бесконечной масштабируемостью.

Введение

Выбор между POW и POS

При выборе между консенсусами POW и POS мы руководствовались следующим:
POW несмотря на высокие энергетические затраты имеет низкий порог входа - достаточно обычного компьютера или ноутбуку, чтобы запустить свою ноду участвующую в создании (майнинге) блоков. А также, как это будет показано ниже, создавать неограниченную масштабируемость.
POS имеет уязвимость вида “Nothing at Stake” и высокий риск централизации,а масштабируемость сети ограничена возможностями используемого оборудования нодами-валидаторами.
Другие протоколы на основе POS имеют те же недостатки:

DPOS - является разновидностью POS, в котором ограничено число валидаторов

POA (Proof of Authority) - по своей сути является также разновидностью POS, в котором ставка это авторитет ноды с весом 1. Это эквивалентно раздачи голосующих монет определенным доверенным людям.

В протоколах консенсусов решение принимает алгоритм. Особенность в том, что для него весь мир ограничен черным ящиком и он не может узнать достоверность того или иного события физического мира. Но есть две вещи которые он все может проверить:

1. Благодаря криптографии - цифровую подпись, если у него заранее имеется достоверная информация о публичном ключе (например из генезис блока)
2. Благодаря POW - мощность оборудования, которое использовалось для расчета блока.

Достоинство POW кроется в математике - мощность выполненной работы можно передать на огромное число устройств и шардов (других связанных участков сети), так как:

1. Низкозатратно по размеру информации (нужно передать только хэш блока и число nonce)

2. Передается быстро по времени (по логарифмической зависимости от числа нод)
3. Достоверно с точки зрения доказательства (берется число первых нулей хеша, вероятность получения которых зависит только от мощности оборудования)

Доказательством на основе POW можно покрыть все ноды на планете и фактически сделать безопасными все транзакции.

В тоже время в POS информация валидна (т.е. может быть доказана) только в рамках одной сети, так как она опирается на проверку цифровой подписи. Ноды имеют физическую возможность проверить цепочку перехода прав собственности монеты (стейка) только одной сети, а так как другие сети не имеют такой возможности по определению (для них монеты других чейнов - это только цифры), то значит невозможно создать протокол с достаточно большой и тем более бесконечной масштабируемостью.

Проблемы современных блокчейнов

Низкая производительность.

Основная проблема блокчейнов на основе PoW это низкая скорость работы. Можно выделить два важных недостатка:

- Ноды не умеют быстро синхронизировать блоки между собой. Соединения нод с друг другом случайно, нет протокола оптимальной организации сети
- Случайное время формирования нового блока. Консенсус Накамото в качестве признака формирования блока использует расчет хэша требуемой целевой сложности. Расчет хэша это случайная величина, поэтому и время случайно.

Комплексное решение этих вопросов позволяет достичь производительности блокчейнов на уровне 1000-2000 tpx на одно ядро процессора.

DApps и централизация

DApp (Decentralized Application) — децентрализованное приложение. Однако на данный момент термин применяется неправильно. Этим термином называют программу, которая взаимодействует со смарт-контрактом в Блокчейне, но фактически расположена на централизованном сервере. Та часть, которая находится на централизованном сервере является ключевой, и без неё работа невозможна. Децентрализованное приложение такого типа может гарантировать только сохранность средств пользователя, так как они находятся на Блокчейне. Практика показала, что возможны отказы в работе при массовом наплыве пользователей на DEX/Swap, например, при резком изменении курса.

Такая ситуация сложилась из-за того, что текущие Блокчейны не предоставляют в своей платформе услуг хостинга. Такого понятия как «интерфейс пользователя» нет на существующих Блокчейнах (за исключением Блокчейна TERA).

Изолированность блокчейнов

Часто крупным проектам требуется высокая производительность и собственные алгоритмы эмиссии, распределения награды, инфляции и прочее. Они запускают свои блокчейны. Но эти блокчейны изолированы от всех остальных. Это приводит к проблеме ликвидности и неудобству пользователей. Решая эти проблемы проекты строят кросс чейн мосты, которые в свою очередь имеют новые недостатки: централизация, медлительность, проблемы с безопасностью.

Тера это платформа блокчейнов. В нем нет форков, так как каждый форк - это шард, который может иметь общих майнеров со всем другими шардами. Таким образом становится возможным проведение кросс-шардинговых транзакций между ними. Создание нового шарда (блокчейна), это простой, бесплатный и независимый от третьих лиц процесс.

Низкий уровень распространенности. Трудности разработки.

Блокчейн приложения крайне низко распространены среди пользователей.
Можно выделить две причины:

- Тяжелый вход простых пользователей. Для начала работы с таким приложением требуются дополнительные действия и знания, которые могут быть непреодолимым барьером для большинства. Например установка плагина, инициализация приватного ключа.
- Для разработки приложений требуются знания специфических языков программирования,которые имеют высокую сложность и низкую популярность среди программистов. К тому же некоторые блокчейны имеют неудачную архитектуру:
    - Медленная виртуальная машина (стековый подход исполнения)
    - Один тип данных среды исполнения (uint256)
    - Неполный язык программирования, который не может полноценно работать без низкоуровневых (ассемблерных) вставок

Массовому применению Блокчейна поможет простота разработки смарт-контрактов и Даппов. Для этого язык должен быть максимально простой и максимально понятный большинству программистов. Например, необходимо использовать самые популярные современные распространенные технологии (JavaScript и HTML) адаптированные для блокчейна.
Веб-программисты являются самой большой армией IT-индустрии. Они ближе всего к пользователям, они смогут максимально быстро создать востребованные приложения на Блокчейне и сделают их максимально удобными для пользователей.
Все это позволит привлечь сотни миллионов новых пользователей блокчейнов, а границы между Даппами и обычными приложениями будут стерты.

Архитектура

TERA — это платформа для создания блокчейнов, а также ДАппов внутри них. Она состоит из хранилища программ, хранилища данных и протокола децентрализованного взаимодействия. Механизм публикации программ и данных является свободным от цензуры. Использование однорангового шардинга позволяет создавать неограниченное число равноправных подсетей (шардов) и таким образом практически бесконечно увеличивать общую производительность сети.

Консенсус

Пользователь подписывает транзакцию и отправляет в сеть. Сеть каждые три секунды формирует новый блок из полученных транзакций. Каждый блок имеет в своем составе хэш предыдущего блока, что позволяет соединять их в последовательные цепочки. Информация записанная в такие цепочки защищена от перезаписи консенсусом PoW

Хэш блока представляет собой однонаправленную криптографическую функция sha3 и имеет мощность, которая определяется числом его первых нулей.
Нахождение хэша с большим числом нулей является случайным процессом и зависит от количества перебора чисел nonce и номера счета майнера.
Ноды стремятся рассчитать как можно более мощный хэш, для того чтобы найти хэш лидер и получить награду.

> Отличие от биткоина
> В Тера майнеры соревнуются за поиск хеш-лидера, в случае такого нахождения они распространяют по сети информацию только о найденном nonce и номере своего счета для получения награды. Трафик минимальный. В Биткоине майнеры ищут блок лидера и в случае нахождения они распространяют по сети целый блок, что приводит к большому трафику и следовательно к задержкам.

В Тера длина цепочки зависит только от глобального времени сети. Если известно текущее время, то можно автоматически вычислить номер текущего блока, а значит длину всей цепи.
Основная цель в консенсусе - это правило (программный код) следуя которому любая третья независимая сторона может однозначно определить какая цепочка является основной (лидером). Поэтому правило консенсуса в Тера следующее: **цепочка лидер та, которая имеет большую сумму мощностей всех блоков.**

Для того чтобы создавать цепочку лидер каждая нода добавляет к новому блоку ссылку на предыдущий блок с максимальной силой цепочки.

> Отличие от биткоина
> В Биткоине новый блок формируется только при достижении целевого значения сложности вычисления блока (количество первых нулевых битов в хеше), а цепочка лидер определяется только ее длиной. В каждый блок записывается время его создания, через заданное количество времени (эпоха) происходит перерасчет новой целевой сложности.
> Это имеет недостаток, так как атакующий может создать цепочку большей длины, потратив небольшие вычислительные мощности путем записи в каждый блок фиктивного времени создания, так что алгоритм пересчета сложности не будет увеличивать целевую сложность. Таким образом третья сторона не сможет только по длине определить правильную цепочку. В биткоине для защиты от этого используются другие механизмы, такие как чекпоинты (записанные хеш коды в самом коде). В Тера это не требуется так как механизм определения лидера заложен в самой архитектуре.

Время сети

Время в Тера используется для определения номера текущего блока. В зависимости от него каждая нода принимает решение какие блоки нужно обрабатывать - т.е. получать от других нод, искать максимальный хэш лидера, создавать новые блоки.

Для того чтобы сеть работала синхронно нужно иметь примерно одинаковое время. В Тера разрешено отклонение в обработки блоков на одну единицу. Таким образом она обрабатывает предыдущий блок, текущий и последующий (блок из будущего). В случае если номер блока не совпадает с текущим по мнению ноды, то она только передает (транслирует) информацию о нем по сети.

Для синхронизации времени используется протокол, работающий по следующей логике:

- Собирается статистика моментов получения максимальных хэшей
- Вычисляется медиана полученной выборки
- Добавляется константа-дрейф к абсолютному времени по UTC (“настенным” часам)

Более подробно:
https://docs.google.com/document/d/1MWmqTvO9Zi5pKZzpnCGogQGBSwbEmMmLXSrVQOaPv1E/edit?usp=sharing

Синхронное состояние блоков

Каждый блок состоит из транзакций, транзакции отправляются пользователями. Для того чтобы состав блоков было одинаковый сразу во всех нодах применяется следующее правило:

- Число транзакций ограничено
- Применяется сортировка транзакций по приоритетам
- Лишние транзакции обрезаются и отправляются в следующих блоках
- Приоритет зависит от размера транзакции и сложности исполнения (числа тиков смарт-контракта), числа монет на счете отправителя и количества ранее отправленных транзакций

Таким образом, если каждая нода будет содержать один и тот же набор транзакций (это достигается сетевым протоколом гарантированной по времени доставки), то у них будет идентичный состав блоков.

Сеть

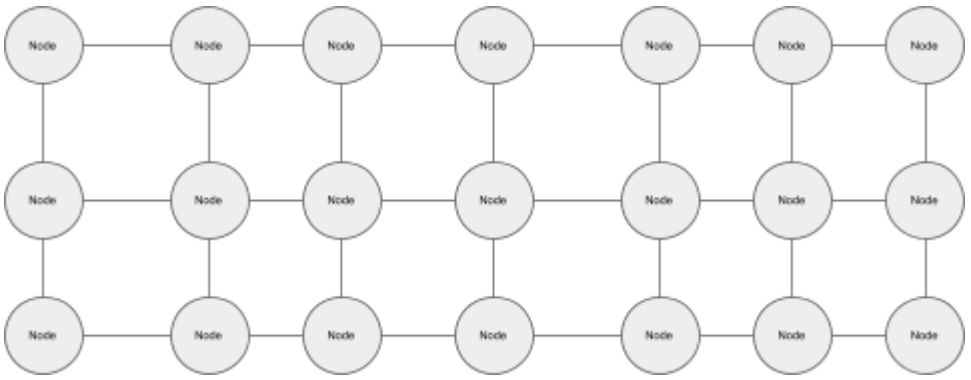
Мы используем собственный протокол **Crystal DHT** протокол, который обеспечивает:

- Высокую связность нод
- Константность времени доставки транзакций по всей сети

Такой подход позволяет обеспечить гарантированную доставку транзакций на все ноды сети.

Описание протокола

Все ноды самоорганизуются в упорядоченные соединения образуя многомерную регулярную решетку. Ниже показана решетка в ее двухмерном представлении, но на практике она имеет многомерную (8-мерную и более) структуру.

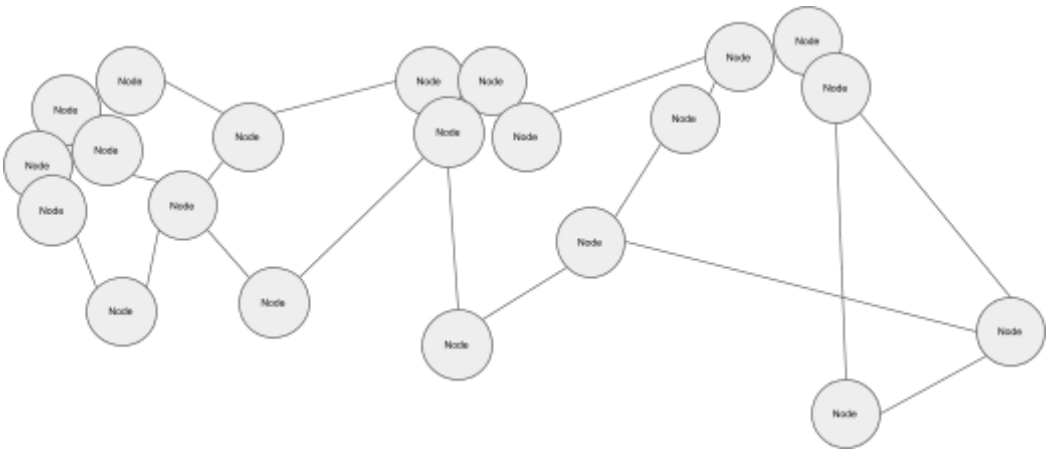


Узлы образуют соединения между собой, который основывается на подобии их хеша их сетевого адреса. Такой хэш является случайной величиной с длиной 32 байта и не меняются в процессе работы узла. Случайность хеша используется для деления нод на уровни связей, с экспоненциально убывающим числом. На первом уровне доступны 1/2 нод, на втором 1/4, потом 1/8 и т.д. Внутри каждого уровня есть приоритет в виде работы с более "отзывчивой" нодой. Так как ноды распределены на уровнях неравномерно, то вначале уровней доминируют более быстрые ноды, а затем более редкие (т.е. случайные). Нижние уровни связывают ноды в локальные группы (т.е. ноды которые находятся в одинаковых регионах), верхние уровни - связи между локальными группами. Заметим, что достаточно такой одной связи для локальной группы. Чем больше группа, тем больше вероятность что найдется хоть одна связь.
Другими словами такой алгоритм блокирует создание закрытых локальных групп не связанных с друг другом.

> Пример в цифрах:
> Максимальное расстояние регулярной сети нод зависит логарифмически от количества узлов в сети, что обеспечивает константное время доставки транзакций. Так, если в сети будет 1 000 000 000 узлов, а среднее время доставки между узлами будет составлять не более 100 миллисекунд, то максимальное время составит: 10 * 100 мс = 1 сек. 10 – это логарифм по основанию 8 (числа активных соединений) из 1 миллиарда нод. При этом время задержки доставки транзакций между узлами менее 100 мс внутри локальной группы, и не более 500 мс для связи с другим континентом.
> Таким образом блокчейну для доставки данных от 1-го узла до последнего требуется не более 3 секунд.

Для того чтобы “зацементировать” удачные соединения, каждый узел для себя записывает статистику удачных обменов с другим узлом, с которым обменивается содержимым блоков. Эта статистика влияет на приоритет соединения и позволяет защититься от атаки типа “затмения”, в которой зловредные ноды пытаются окружить одиночную ноду и таким образом исказить информацию о сети и цензурировать блоки и транзакции.

> Отличия в других сетях
> В традиционных блокчейнах не применяется упорядочивание узлов друг с другом, в них отсутствуют протоколы формирования сети, вместо этого применяется gossip подход, согласно которому информация распространяется случайно, а диаметр сети не является константой. Общая картина сети выглядит так:



Данная случайная организация связей не гарантирует быструю доставку блоков между всеми узлами. Поэтому время формирования блоков в других pow блокчейнах порядка минут.

Трафик сети

Для снижения трафика мы используем два подхода:
Первый - это протокол (который можно абстрактно описать как обдувание нод друг другом слабым ветром перед тем как отправить большой ураган данных)

- Вначале ноды отправляют друг другу короткие хэши от транзакций (слабый ветерок)
- Затем отправляются данные которые не были зафиксированы от друг друга на первом шаге

Такой протокол позволяет оптимально (т.е. идеально) оптимизировать трафик передачи транзакций.

Второй подход - это отправка только информации о хешах лидеров, вместо отправки блоков лидеров как сделано в других блокчейнах. Это позволяет на практике снизить технический трафик (т.е. когда нет пользовательских транзакций) до примерно 4 Кбайт/сек

Можно выполнить моделирование сети блокчейна здесь:
https://terafoundation.org/JINN/model/model.htm
Особенность в том, что используется тот же самый код (библиотека jinn) который запущен в mainnet Tera.

Среда исполнения

Язык исполнения смарт-контрактов JavaScript работает в среде V8, который разработан компанией Google. В нем компиляция исходного кода выполняется сразу в собственные машинные инструкции минуя стадии байт-кода. Это дает высокую производительность и позволяет обрабатывать тысячи транзакций в секунду на одном ядре.
Язык JavaScript полный по Тьюрингу, поэтому для ограничения времени исполнения смарт-контрактов используется понятие Tik, который равен исполнению одной строки кода. Максимальный размер ограничен константой. Tik - это аналог GAS в Ethereum, но только за него не платится комиссия.
Для обеспечения безопасности код смарт-контрактов проходит предварительную трансляцию/компиляцию и выполняется в специальной защищенной виртуальной среде.

Мотивация майнеров

Каждая нода помимо четкого следования правилам по созданию цепочки лидера может включиться в гонку расчета максимального хэша блока, чтобы в случае успеха получить награду. Для этого она перебирает числа nonce и максимальные значения мощности отправляет в сеть (при этом в сеть отправляется только доказательство: ссылка на текущий блок, числа nonce и номер счета для награды).

Хэш с максимальным номером добавляется в блок, а майнер его нашедший получает награду во время исполнения блока - в виде системной coinbase транзакции, которая неявно добавляется в каждый блок.

Размер награды определяется через DAO каждым шардом отдельно в соответствии с собственной политикой токеномики. Например, 1-2 монеты каждый блок и целевым показателем инфляции 2% в год.

Конвейерная обработка

Для обеспечения высокой производительности блокчейна мы используем конвейерную обработку, состоящую из шагов:

1. "STEP_ADDTX" - добавление транзакции в новый блок
2. "STEP_TICKET" - отправка информации (тикета в виде короткого хэша) о транзакции соседним нодам
3. "STEP_TX" - отправка самой транзакции (если в соседней ноде такой еще нет)

4. "STEP_NEW_BLOCK" - майнинг нового блока и распространение информации о хэше лидере
5. "STEP_SAVE" - запись нового блока в базу данных на диск
6. "STEP_USE_TX" - изменение состояний аккаунтов и KeyValue хранилища
7. "STEP_RESEND" - переотправка транзакций, которые не попали в блок

Как можно заметить в конвейере нет шага отправки блока, так как из-за особенностей протокола блоки одинаковы во всех нодах уже на шаге STEP_NEW_BLOCK.
Это достигается тем что транзакции гарантированно доставляются всем нодам, внутри блока одинаково ранжируются по приоритетам, а coinbase транзакция добавляется автоматически на шаге STEP_USE_TX.

Слои обработки

Мы разделили процессы на два основных слоя:
1. Слой быстрой синхронизации нод и доставки блоков
2. Слой исполнения (применения) транзакций

Напомним что такое блокчейн: это компьютерная сеть в которой каждый узел является равноправным, их число неограниченно, общение между осуществляется посредством организации единой цепочки данных, в которую информация записывается поблочно в виде команд (транзакций).
В биткоин подобных блокчейнах в блоки записываются только платежные транзакции, при этом не допускается запись в блок транзакции, которая не является валидной (например, не имеет правильную цифровую подпись или не достаточно денег на входном адресе или двойная трата и т.д.).
В Тера, а также других блокчейнах, таких как Эфириум, это допускаются. Это сделано для быстроты обработки, а также из-за того что заранее невозможно узнать результат обработки отдельной транзакции вне состава и порядка в блоке.

В блокчейне TERA допускается запись любой информации в блок, а блокчейн используется как транспорт. Каждая запись (далее мы их будем называть транзакциями**)** имеет свою строгую нумерацию и разделены на блоки, блоки связаны с друг другом посредством необратимой криптографической хеш-функции. На первом слое задача блокчейна обеспечить одинаковость информации в каждой ноде сети. Эта задача решается посредством классического консенсуса **PoW**.

Интерпретация правильности информации лежит на втором слое. На втором слое реализована поддержка криптовалюты - встроенной монеты **Тера**. Начисление награды за блок майенрам. На этом же слое реализованы смарт-контракты.
Т.к. на первом слое гарантирована одинаковость данных, и так как код программы на втором слое одинаков на всех нодах сети, очевидно, что выполняя одни и те же действия все ноды будут иметь одинаковый результат: одинаковые остатки по счетам

пользователей, одинаковые состояния смарт-контрактов. Таким образом, если в этих блоках будут невалидные транзакции типа двойных трат, то валидирующий слой одинаково их отклонит на всех узлах сети.

Валидацию можно выполнять в другое время и в другом процессе, не мешая блокчейну. Более того - это можно это делать гораздо быстрее за счет так называемой пакетной обработки (массовости проверок) — мы можем группировать операции и ускорить работу за счет меньшего количества обращений к Базе Данных.

TERAHASH

Цель данного алгоритма уравнять между собой людей майнящих на CPU и на GPU. Для достижения этого мы предлагаем использовать память, но в отличие от других похожих алгоритмов (например Ethash) в нашем алгоритме память не замедляет работу GPU, а ускоряет работу CPU. Это можно осуществить тем, что при подборе хеша использовать не целое число nonce, а определенное значение, которое трудоемкое для вычисления - например вычисленное по алгоритму sha3 и допустить применение этого значения для перебора в широком диапазоне вычисления хеша блоков. Таким образом будет выгоднее сохранять эти значения в памяти для подбора, чем вычислять заново. Такая выгода должна сохраняться даже если скорость вычислительных ресурсов увеличится в 1000 раз.

Алгоритм

На вход подается 32-байтный хеш данных текущего блока *CurrentDataHash*, нужно найти такой *Nonce* (целое число), чтобы в результате получился подходящий для нас хеш блока с максимальным значением начальных нулей.
Ограничение:

1. Поиск должен быть оптимизирован на использование памяти - для защиты от GPU-майнинга
2. Проверка должна осуществлять при минимальном количестве памяти и выполняться быстро - примерно со скоростью вычисления sha3

Порядок расчета:

1. Вычисляется *HashNonce*= sha3(*PrevHashN* , *Nonce*)
2. Получаем *Hash* = `SimpleMesh`(*HashTemp,CurrentDataHash)*

где:

PrevHashN - 32-байтный хеш некоторого предыдущего блока, отличающегося от текущего на *NDelta* блоков (максимальная глубина ограничена определенным числом, например 1000 блоков)
Nonce - число для перебора значений от 0 до макс целого числа
`SimpleMesh`() - быстрая функция перемешивания (не криптографическая). Она должна удовлетворять условиям:

1. Сохранение энтропии
2. Высокая скорость - в 1000 раз больше скорости вычисления функции на шаге 1 (в данном примере sha3).
3. Должна хорошо перемешивать данные - чтобы предотвратить быстрый поиск, т.е. гарантировать перебор HashNonce

После поиска максимально удовлетворяющего нас хеша в *блокчейн* записывается: *CurrentDataHash*,*Nonce*, *NDelta* - по которым быстро восстанавливается хеш блока

Дополнительное требование: майнер должен доказать, что база не менялась. Он должен найти следующие два nonce:

1. Один который подходит текущему хешу блока
2. Другой к предыдущему хешу блока

В общем виде формула будет такова:
Hash = XOR(DataHash, HNonce1)
Hash2 = XOR(PrevHash,HNonce2)
PowerHash=min(Power(Hash),Power(Hash2))

Безопасность

Транзакции попадают в блок который является последним по времени, этот блок называется текущим, через следующие 3 секунды будет создан новый блок, затем еще один блок и так далее. При этом каждый блок связан с предыдущим, образуя цепочку. Число таких блоков, которые созданы после блока с отправленной транзакцией называется числом подтверждений транзакции.

Протокол имеет вероятностную финальность транзакции - чем больше подтверждений тем выше вероятность что перезапись будет невозможна.

Пока число честных майнеров остается больше 50% атака считается невозможной, так как вероятность перезаписи блока меньше **0.5**, а произведение вероятностей стремится к нулю с ростом числа подтверждений.

Защита от двойной траты

Транзакция защищена от двойной траты если практически невозможно перезаписать блоки (т.е. вероятность стремится к нулю).
Пользователи Bitcoin для защиты от двойной траты ждут от 10 минут до часа, ожидая нужного числа подтверждений.

В сети TERA блоки создаются быстрее, а время первого подтверждения 8 секунд, но, если вы хотите такую же степень надежности, как и в сети Bitcoin, вам нужно ждать аналогичное время. Время ожидания – это степень надежности. Здесь нет магии, в любом алгоритме PoW вы меняете время на надежность. В TEPA мы сделали более гибкую возможность выбора. Можете ждать 8 секунд, 1 минуту, а можете 1 час (если суммы переводов значительны).

Защита от replay-атаки

TERA использует Блокчейн с консенсусом PoW. Это позволяет расположить все транзакции последовательно друг за другом. Выполнение транзакций также производится последовательно. При списании со счета проверяется наличие необходимой суммы. В момент списания увеличивается счетчик номера операции ("OperationID"). Каждая последующая платежная транзакция должна иметь следующий или больший номер "OperationID" для предотвращения применения одной и той же транзакции несколько раз.

Защита от DDoS атак

Размер блока ограничен константой (например 350Кб), а также ограничено число транзакций в блоке (например 3000). Транзакции в блоках сортируются по приоритетам, а лишние обрезаются и ограниченно перенаправляются в другие блоки. Приоритет зависит от размера транзакции и сложности исполнения (числа тиков смарт-контракта), числа монет на счете отправителя и количества ранее отправленных транзакций.
Таким образом чем больше транзакций было отправлено от конкретного счета, тем меньше становится приоритет и меньше вероятность попадания в блок.

Защита от атаки Сибиллы и атаки “Затемнения”

В сети все ноды равноправны и анонимны, поэтому возможны атаки Сивиллы, когда злоумышленник создает большое количество нод и пытается вести злонамеренные действия.
“Хорошая” нода отличается от “плохой” ноды, тем что она четко следует протоколу.
“Плохая” нода может:

- не передавать информацию когда это нужно

- передавать информацию когда это не нужно
- передавать ложную информацию

Цель каждой ноды в сети корректно передать информацию. Результат корректной передачи это успешно синхронизированный блок, отсутствие орфан цепочек. Этот показатель виден спустя несколько секунд после обмена и он объективен, т.к. защищен консенсусом pow блока, поэтому его подделать нельзя, не имея 51% мощностей (но мы в рассуждениях всегда исходим из того, что 51% мощностей в сети - честные). Таким образом у нас есть в наличии достаточно хороший механизм определения правильности следования протоколу тех нод, с которыми мы обмениваемся. Поэтому для того чтобы “зацементировать” удачные соединения, каждый узел для себя записывает статистику удачных обменов с другим узлом. Эта статистика влияет на приоритет формирования связей при создании сети по типу многомерной регулярной решетки.

Шардинг

Понятие блокчейна как философии децентрализованной передачи ценностей глубоко впиталось в общество и в нем появилась потребность создания такой глобальной сети, которая полностью децентрализована и не имеет проблем с масштабируемостью.

На основе нового протокола шардинга можно создать сеть с миллионами нод и общей производительностью измеряемой миллионами транзакций в секунду. Основной принцип: в сети создается неограниченное количество равноправных блокчейнов, объединенных с друг другом через общий хеш сети. Блокчейны коммуницируют с друг другом напрямую через кросс-шардинговые транзакции. Так как в сети нет главного чейна через который осуществляются такие транзакции, то и нет узкого места, что позволяет обеспечить практически бесконечную масштабируемость. Безопасность сети осуществляется внешним контуром легкого блокчейна, он состоит только из заголовков общих хешей. Майнеры сети выполняют работу по подбору такого общего хеша сети и занимаются валидацией дочерних блокчейнов.

Введение в шардинг

Обеспечить максимальную децентрализацию в виде практически неограниченного количества блок-продюсеров возможно на консенсусе PoW. Но простое разбиение нод сети на шарды в виде отдельных блокчейнов хоть и приводит к кратному росту общей производительности, но также приводит к кратной потери безопасности. Становится проще провести атаку 51% на каждый блокчейн в отдельности. Действительно если мы всю сеть разбили на 100 равных по мощности майнинга шардов, то для атаки на отдельный блокчейн достаточно 0,51% мощности майнинг-оборудования всей сети.

Чтобы предотвратить такую угрозу можно применить совместный (merged) майнинг. В этом случае майнеры рассчитывают только общий хеш сети, сложность хеша остается такой же и таким образом безопасность сети не ухудшается. Но возникает проблема как правильно организовать валидацию шардов, т.к. ресурсы одной ноды ограничены (для гарантии децентрализованности майнинга будем исходить из того что одна нода может валидировать не более одного шарда).
Центральная идея решения этой проблемы это использование социального аспекта майнеров - использование “теории шести рукопожатий”. Майнеры могут иметь несколько нод, майнеры могут иметь связи (друзей которым доверяют).

Шардинг в TERA

На практике возникает простой вопрос - что делать обычному рядовому пользователю, который скачал программу майнинга, у которого только один компьютер и соответственно ресурсов для валидации хватает только на один шард?

В консенсусе PoW существует интересная зависимость между безопасностью сети и временем. Чем больше времени затрачиваем на создание хеша, тем больше безопасность сети и наоборот. Это что-то вроде закона, который нельзя обойти, но который нужно использовать. Отсюда очевидный вывод: если нет достаточного количества майнеров поддерживающий валидацию шарда, то значит этот шард будет реже по времени иметь подтверждающий блок. Шарды которые имеют достаточно редкое количество подтверждений будут менее предпочтительны для пользователей, это будет отражаться на их рыночной капитализации (т.к. каждый шард это самостоятельный блокчейн со своей криптовалютой).
Выглядеть это будет так: майнер указывает связь между своими нодами и шардами, которые эти ноды валидирует. При майнинге в меркл-дерево блока победителя включаются **только** хеши провалидированных шардов. Пользователи будут считать блок шарда подтвержденным только если он включен в общий хеш сети.
Таким образом у майнера у которого только одна нода будет провалидированным только один шард, а у майнера у которого много нод - практически все шарды. Т.к. вероятность нахождения блоков будет у майнера, у которого больше ресурсов (нод), то:

- Если в сети будут превалировать ноды профессиональных майнеров у которых много нод, то в сети большинство блоков будут иметь все провалидированные шарды.
- Если количество нод (с условно равными мощностями) будет равно, то в среднем все шарды будут провалидированы через один блок.
- Если будет больше одиночных майнеров, то вероятность валидации шарда будет находиться в прямой зависимости от его популярности (и в обратной от общего количества шардов).

Таким образом:

- Новый майнер с единичной нодой сможет достаточно просто участвовать в майнинге, при этом он мотивирован для валидации других шардов в любой момент когда пожелает нужным.
- Для введения нового шарда достаточно чтобы его провалидировал хотя бы один майнер. Время подтверждения блоков шардов зависит от популярности и рыночной эффективности шарда.

Все это вместе обеспечивает плавный и органичный механизм образования и поддержания шард, который будет регулироваться рыночными методами - путем изменения курса встроенной криптовалюты шарда.

Термины шардинга

Сеть - под этим термином понимается виртуальная (релейная) сеть в виде связи нод (компьютеров) с друг другом в определенной последовательности, сгруппированных по разным шардам (блокчейнам) и работающих под одним протоколом передачи данных.
Шард - в данной версии документа под шардом понимается отдельный блокчейн, особенностью которого является то что его валидацию выполняют майнеры общей сети. Шард это часть сети.

Монеты шардов

Каждый шард имеет собственную встроенную монету (криптовалюту),
кросс-шардинговые транзакции - это фактически кросс-шардинговые свопы (обменники). Нет глобальной криптовалюты. Обмен криптовалют из разных шард возможно только при создании смарт-контрактов в виде децентрализованных бирж, курс в которых зависит от спроса и предложения.
Для хождения единой криптовалюты, например Теры, каждый шард может имплементировать платежные каналы в виде смарт-контрактов, который меняет один токен Теры из одного шарда на другой токен Теры в другом шарде (ид токена зависит от самого смарт-контакта, от того что он считает Терой). Общий баланс монет и токенов внутри шарда остается постоянным.
Такой подход обеспечивают финансовую безопасность для пользователей шардов, если предположить что один шард скомпрометирован, то остальные шарды не пострадают.

Нода

Нода это основная единица сети. Все ноды равноправны. Они объединяются с друг другом по определенным правилам и образуют так называемую релейную сеть, которая позволяет оптимально обмениваться информацией в сети. Ноды не соединены напрямую с друг другом, каждая нода имеет ограниченное число связей - оно пропорционально логарифму количества нод во всей сети.

Функции ноды как участника сети:

1. Валидация и обмен данными шарда (блокчейна). Стандартно нода может валидировать только один шард.
2. Валидация и обмен данными кросс-шардинговых транзакционных буферов (максимальное число ограничено)

Горизонтальное масштабирование при совместном майнинге

При большом количестве шардов (например тысяч и сотни тысяч) валидация всех шардов одним обычным майнером практически невозможна - одна нода может провалидировать только один шард, а число нод у обычного майнера ограничено. Горизонтальное масштабирование позволяет увеличить эти возможности следующим образом:

1. Первый способ - создать кластер из собственных нод. Для этого в настройках каждой ноды задается принадлежность к одному кластеру путем ввода общего секретного пароля. В этом варианте число поддерживаемых шардов ограничено числом собственных нод майнера.
2. Второй способ - включение внешнего кластера нод, которому майнер доверяет (например своего друга) в дерево доверительных кластеров. Это делается путем добавления публичного ключа внешнего кластера и указания уровня иерархии доверия (можно ли доверять в свою очередь его дочерним узлам). При таком варианте число поддерживаемых шардов может многократно превышать число собственных нод майнера. Причем можно указывать степень

Майнер может комбинировать оба варианта для достижения максимальной надежности и прибыли.

Вариант стратегии пользователя:

1. добавить свои ноды (5 штук по одному шарду)
2. добавить кластер нод друзей (2 кластера с одним уровнем иерархии, т.е. только ноды друзей)
3. добавить кластер из 100 шардов известной компании типа VISA/MasterCard

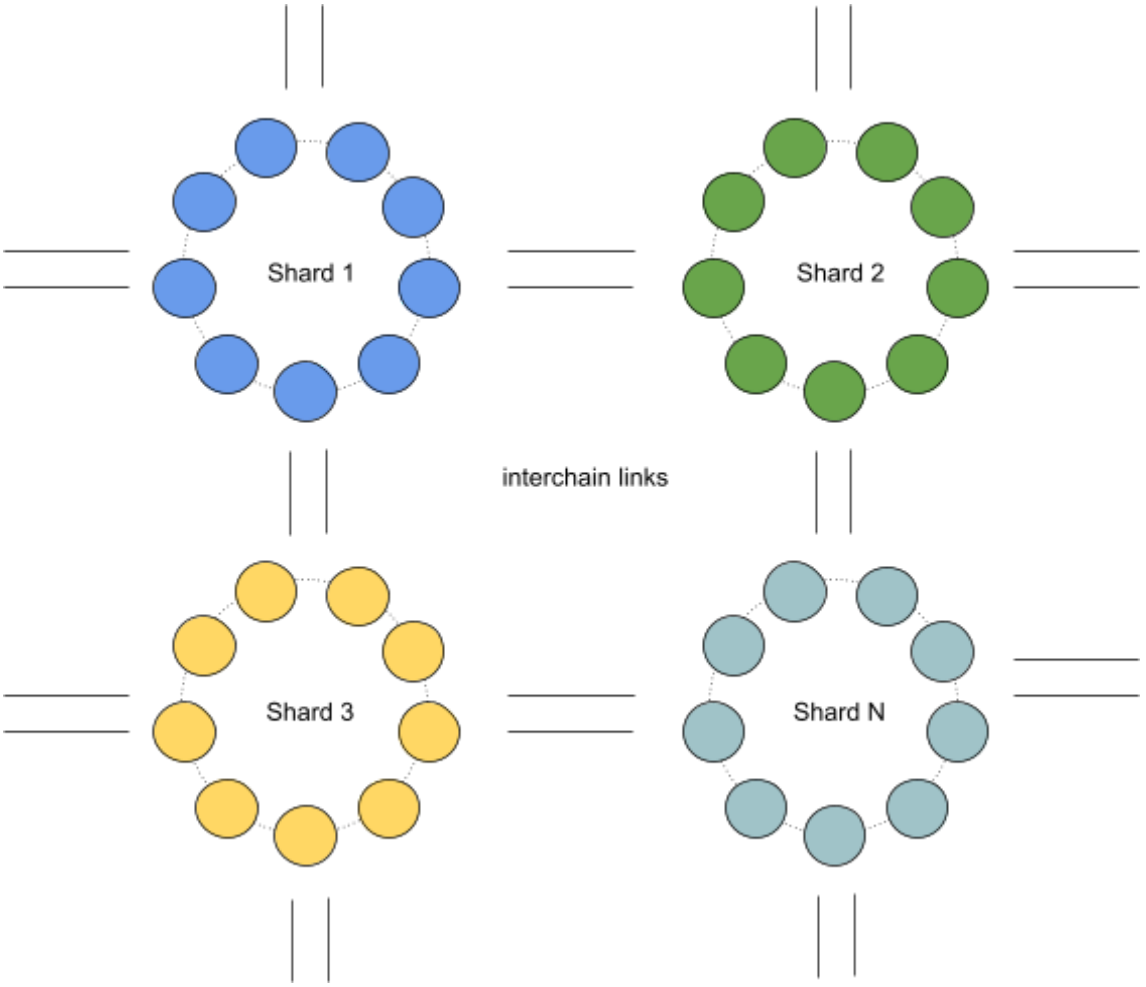
Мотивация майнера шарда

Мотивация майнера - это получение наград в каждом шарде, таким образом он стимулирован чтобы провалидировать и включить в состав заголовка хотя бы один шард. Алгоритмом допускается возможность включения ноль шардов в этом случае хеш шардов (ShardsHash) имеет нулевое значение.
Для того чтобы максимизировать прибыль майнер должен стремиться провалидировать как можно больше шардов, т.к. награды он получает независимо в каждом из них. Так как нода может валидировать только один шард, то майнер может запускать несколько нод с настройкой на валидацию разных шард, а чтобы в блок

включать информацию по нескольким шардам он может объединить свои ноды в доверенный кластер.
Оптимальная стратегия майнера будет заключаться поиск кластеров, которым он может доверять для включения их в свой майнинг лист. В идеале он будет стремиться к 100% охвату шардов сети.

Схема обмена сообщениями

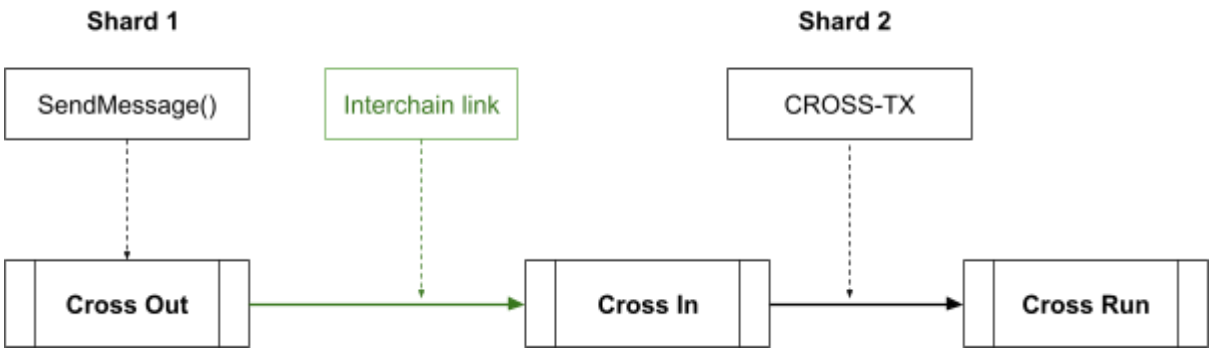
Смарт-контракты могут обмениваться сообщениями между друг другом, в том числе между разными шардами.
При создании смарт-контракта задается высота в блоках, которая определяется финальность передачи сообщения из одного щарда в другой. При достижении такого значения сообщение передается на дальнейшую обработку в смарт-контракт.
Минимальная высота - это фактическ величина безопасности в случае если смарт-контракт является каналом передачи ценностей или другой важной информации. Чем он больше тем более безопасным является канал, но с другой стороны - больше времени нужно ожидать пользователям для завершения операции.
Процессом переноса сообщений из одного шарда в другой занимаются те майнеры, которые одновременно майнят эти два шарда. В случае если нет таких майнеров, то сообщение будет отложено и ждать их появления.



Кросс-шардинговые транзакции (сообщения)

Если у нас есть нода, которая майнит несколько шардов одновременно, то при создании блока она может добавить в него специальные транзакции, которые содержат информацию о наличии в других шардах сообщений, предназначенных для нашего блокчейна. Такие транзакции сами по себе не участвуют в обмене между нодами до создания блока. Они могут передаваться другим нодам только неотрывно от блока.

Майнеры выполняют голосование за тот или иной состав сообщений в блоках путем создания виртуальных цепочек через специальные транзакции голосования, чем больше блоков содержит цепочка, тем больше достоверность сообщения. При достижении определенного порога, задаваемого изначально, происходит фиксация сообщения в шарде получателе. Фактически майнеры выполняют роль оракулов - свидетелей наличия какой-либо информации в шарде источнике.



Безопасность каналов

Безопасность каналов можно определить через стоимость атаки 51% на время которое необходимо для успешной передачи сообщения в шард приемник. Чем быстрее канал, тем дешевле атака или другими словами больше риск. Управляя рисками мы можем построить схему, которая обеспечивает одновременно высокую безопасность и высокую производительность. Для этого создадим два канала передачи ценностей:

1) Основной канал - медленный, но надежный канал. Время фиксации всегда указывается достаточно большое, например 1 млн блоков. Этот канал имеет свой токен в шарде приемнике, который и отождествляет монету шарда источника.
2) Быстрый канал используется для быстрой передачи небольшого размера ценностей (по сравнению со стоимостью атаки). Например, фиксация операции через 100 блоков. Канал тоже имеет свой токен,который будем называть **транзитным**, он нужен только для кратковременного использования.

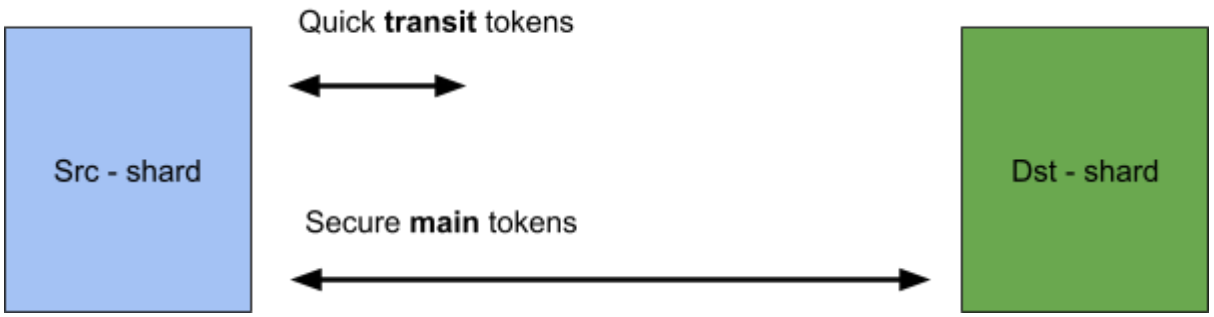


Схема работы такова:

1. Пользователь отправляет сумму ценностей через быстрый канал (сумма ценностей всегда меньше стоимости атаки)

2. Получив их в шарде приемнике в виде транзитных токенов он их меняет на основные токены через встроенный DEX. Основные токены передаются через медленный канал, поэтому безопасны.
3. Обеспечением ликвидности транзитных токенов занимается арбитражник. Он занимается передачей транзитных токенов назад в шард источник, а основных токенов в шард приемник. Чтобы мотивировать его выставлять ордера на DEX курс будет не 1 к 1, а с определенной наценкой в которую закладывается стоимость заморозки капитала и его прибыль. Конкретный курс будет регулировать рынок.

Отправка монет из кошелька между шардами

Есть возможность отправки монет напрямую из интерфейса кошелька между счетами соседних шардов, для это адрес счета получателя должен содержать название шарда и номера счета, разделенные двоеточием: “SHARD:AccNumber”
При этом название шарда резолвится в номер аккаунта со смарт-контрактом, который является шлюзом (как вариант этот список соответствий можно зашивать в сам интерфейс кошелька).

Причина по которой используется промежуточный смарт-контракт - это изолирование ответственности. В случае компрометации одного шарда ценности других шардов не пострадают. При кросс-переводах монеты остаются на счете смарт-контракта внутри блокчейна, в другой блокчейн попадает только своеобразная расписка, на основании которой делаются движения токенов.

Такая модель позволяет создавать шарды очень широкого назначения, например временные но с высокой производительностью внутри.
Пример: Создается шард1, продаются токены на него, пользователи начинают работать с даппами внутри него совершая 1000 tps, через некоторое время когда размер базы превышает разумный предел, например 1000 Гбайт, создается Шард2, токены меняются на токены шарда 1, все переходят работать в шард2, майнеры отключают поддержку 1-го шарда.

Более детально:
https://docs.google.com/document/d/1eglQs3uRiHj7R_mTciF3rKtHRytLmmsO-ordGgzVv2M/edit?usp=sharing

Выводы

Возможно достижение бесконечной масштабируемости путем опционального использования доверия при валидации шард. Но в отличии от других блокчейнов такая доверительная система не имеет проблем с децентрализацией, т.к. по умолчанию она отключена, а если майнер задал группу доверия и при этом ошибся, то в целом сеть от этого не пострадает, т.к. другие майнеры имеют отличные от него группы доверий.