Алгоритм консенсуса для корпоративных сетей блокчейн.

March 2, 2018

Abstract

Предлагается новый алгоритм достижения консенсуса stake distributed Byzantine Fault Tolerant (sdBFT), позволяющий увеличить на несколько порядков количество узлов сети, участвующих в достижении консенсуса и существенно повысить скорость транзакций. В алгоритме используется аналогия с задачей византийских генералов. Для обеспечения равномерного распределения номеров узлов в сети предлагается применять двойное хеширование.

Введение

При создании монеты биткойн и технологии блокчейн Сатоши Накамото использовал в качестве алгоритма консенсуса алгоритм доказательства работы, так называемый Proof-of-Work (PoW) [1]. С ростом популярности технологии блокчейн проявились особенности алгоритма PoW, связанные с низкой скоростью транзакций и, соответственно, их высокой стоимостью.

Многие эксперты из IT-индустрии создали новые либо адаптировали уже существующие алгоритмы консенсуса, устраняющие слабые стороны PoW. В результате появились алгоритмы консенсуса PoS, DPoS, LPoS, PoE, PoIT, pBFT. Данной статьей мы постарались внести свой вклад в развитие алгоритмов консенсуса.

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

Время создания нового блока не более 1 мин.

1. Тип сети блокчейн – корпоративный.
2. Общее количество узлов, которые могут принять участие в выработке консенсуса, может меняться от 10^3 до 10^4.
3. Высокая скорость транзакций – не менее 10^3 транзакций в секунду.
4. Реализации алгоритма не требует существенных энергозатрат.

5. Реализация алгоритма блокчейна не должна требовать существенных вычислительных, по сравнению с блокчейнами PoW, мощностей.

Ранее предложенные алгоритмы консенсуса признаны авторами не удовлетворяющими перечисленные выше требования по следующим причинам.

Консенсус PoW – самый первый тип консенсуса, реализованный в блокчейне валюты Bitcoin. Консенсус отличается невысокой скоростью закрытия блока и малой скоростью транзакций, согласно приведенным данным [2] скорость транзакций валюты Bitcoin составляет всего 7 транзакций в секунду. 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 вычислительной гонке между участниками консенсуса.

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 [3]. 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, что приведет к несоответствию требованиям к блокчейну, изложенным выше.

Консенсус pBFT – еще одна альтернатива алгоритму консенсуса PoW. В мире предложено несколько реализаций консенсуса pBFT, один из них – «Honey Badger» [4]. Как показано на рис. 1 реализация pBFT консенсуса работает тем лучше, чем меньше количество узлов, участвующих в консенсусе.

Рис. 1

Графики зависимости времени задержки формирования блока от скорости поступающих транзакций

На рисунке 1 показан график зависимости времени задержки формирования блока от скорости поступающих транзакций, где Nodes/Tolerance – соотношение общего количества узлов, достигающее консенсуса, и числа узлов, не достигших 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 превышающим 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.

Авторы предлагают другой подход для решения проблемы потери производительности в алгоритме pBFT, реализованный в алгоритме stake distributed Byzantine

Fault Tolerant (sdBFT) консенсуса.

Основные положения и допущения **sdBFT**

Мы исходим из того, что задача достижения консенсуса в первом приближении сводится к получению согласованного решения участниками распределенной системы в случае, если некоторое их количество не приняло участие в согласовании решения. Это может произойти по следующим причинам:

Ошибка при передаче сообщения о принятии решения одним из участников.

Слишком медленная передача сообщения о принятии решения одним из участников.

Сбой в работе участника в системе.

Введение в заблуждение при принятии решения в системе, как умышленное, так и неумышленное.

Если рассматривать причины 1-3 неучастия в выработке консенсуса, то для принятия решения достаточно чтобы выполнялось условие $n > m + 1$ [5], где m – количество участников, которые не участвовали в консенсусе, а n – количество участников принявших решение. В случае появления в системе участников, не участвующих в консенсусе по четвертой причине задача сводится к задаче византийских генералов, которая имеет решение, когда

$$n > 3 \cdot m \tag{1}$$

Если вместо n генералов рассмотреть n узлов блокчейна, участвующих в выработке консенсуса, то очевидно, что эта задача подобна задаче византийских генералов. Для решения проблемы роста времени достижения консенсуса предлагается из конечного множества узлов B_n , мощность этого множества $\overline{B}_n = n$, выделять подмножество $\overline{B}_{n'}$ $(B_{n'} = n^{'})$ и исходя из предположения о равномерности распределения свойств узлов во всей сети блокчейн решать задачу не на B_n, а на $B_{n'}$ при $n \gg n^{'}$. Общее количество узлов во всей сети блокчейн положим равным N. Обозначим множество этих узлов A_N. Пусть задана функция:

$$d : T \mapsto Y_{B_n},\ d = d(t),\ t \in T \tag{2}$$

где t – независимая переменная, соответствующая текущему времени. Будем считать, что значение $d \in Y_{B_n}$ функции (2) соответствует текущему состоянию B_n в момент t. Пусть задана функция:

$$f : Y_{B_n} \mapsto \mathbb{N}_N,\ f = f(d),\ d \in Y_{B_n} \tag{3}$$

где $\mathbb{N}_N$ – конечное подмножество множества натуральных чисел $\mathbb{N}$ мощности N.

Пусть из B_n случайным образом выбрано подмножество $B_{n'}$, причем $n^{'}$ задаётся. Тогда получим:

$$B_{n'} \subset B_n \subseteq A_N \tag{4}$$

Будем считать, что $Y_{B_{n'}}$ – множество значений всех d, соответствующих всем текущим состояниям $B_{n'}$, $Y_{B_{n'}} \subseteq Y_{B_n}$. Пусть функция f отображает $Y_{B_{n'}}$ на множество $\mathbb{N}_{n'}$, $\overline{\mathbb{N}}_{n'} = n^{'}$. Будем считать, что номера узлов, полученных таким отображением есть j_k, $k = 1, \ldots, n^{'}$. Будем полагать, что $j_{\hat{k}}$ – номер назначаемого мастер-узла, $1 \leq \hat{k} \leq n^{'}$. Если b – формируемый блок, в отношении которого в некоторый момент времени $t^{'}$ множество узлов $B_{n'}$ стремится достичь консенсуса, то функцию хеширования SHA-3 [6] над этим блоком, обозначим $H(b)$, а ее значения обозначим h. Тогда результат вычисления электронной подписи, например, по алгоритму Ed-DSA [7] с параметрами эллиптической кривой edwards25519 (???) будет равен $s = sig(h)$.

Описание алгоритма

1. Пусть в момент времени $\hat{t} \in [t, t^{'})$ узел с номером k $(1 \leq k \leq N)$ осуществляет запись I в блокчейн B_n.

2. Выберем все j_k, включая $j_{\hat{k}}$ с помощью функции f. Выработка консенсуса осуществляется на полуинтервале $[t, t^{'})$.

3. В случае признания мастер-узлом допустимым включение записи I в блок b, мастер-узел передаёт всем узлам из $B_{n'}$ эту запись для проверки и включения в блок b. В противном случае запись I отвергается без уведомления.

4. Новая запись включается в блок до наступления момента $t^{'}$. Мастер-узел рассылает сообщение тем же узлам о фиксации блока b. Все узлы из $B_{n'}$ вычисляют значение хеш-функции $H(b)$ равное, допустим, h.

5. Каждый узел вычисляет электронную подпись:

$$s_l = sig(h), \begin{cases} k = 1, \ldots, n^{'} \\ l \neq k \end{cases} \tag{5}$$

   и передаёт её на узел $j_{\hat{k}}$.

6. Мастер-узел ожидает электронные подписи время Δt после наступления момента $t^{'}$. В момент $\Delta t + t^{'}$ на мастер-узле формируется кортеж

$$s_b = (s_1, \ldots, s_j),\ 1 \leq j < n^{'} \tag{6}$$

   Мастер-узел проверяет каждую подпись из (6) и подсчитывает число корректных подписей. Подписи некоторых узлов из $B_{n'}$ могут оказаться голосующими "против" или некорректными в том случае, когда в $B_{n'}$

окажется узел с некоторым номером $j_{\tilde{k}}$, $1 \leq \tilde{k} \leq n^{'}$ который
а) признает запись I некорректной;
б) в момент времени $\tilde{t} \in [t, t^{'})$ имеет состояние блокчейна, $\tilde{d}$ отличное от состояния d для узла$j_{\tilde{k}}$;
в) исказит запись I при формировании блока блокчейна.

7. Мастер-узел вычисляет количество корректных подписей μ и проверяет выполнение неравенства:
$$\mu > \frac{2}{3} n^{'} \tag{7}$$
Если (7) не выполняется, то мастер-узел делает вывод, что консенсус не достигнут, в противном случае для блока b составляется число:
$$b \parallel s_{k_1} \parallel \ldots \parallel s_{k_\mu},\ 1 \leq l < n^{'},\ l = 1, \ldots, \mu \tag{8}$$
для которого вычисляется $H(b \parallel s_1 \parallel \ldots \parallel s_\mu)$ и sig, являющаяся электронной подписью узла с номером $j_{\tilde{k}}$.

8. Число
$$d^{'} = b \parallel s_1 \parallel \ldots \parallel s_\mu \parallel sig(H(b \parallel s_1 \parallel \ldots \parallel s_\mu)) \tag{9}$$
назовём новым закрытым блоком, которое будем считать новым состоянием блокчейна $d^{'}$, соответствующим моменту $t^{'}$. Мастер-узел рассылает (9) всем узлам из множества A_N.

9. На каждом узле из A_N, предположим, с номером $1 \leq m \leq N$, осуществляется проверка s_b и sig (9). Если проверка пройдена, то блок b добавляется в блокчейн узла номер m и блокчейн на узле переходит в состояние $d^{'} = d(t^{'})$. Если этот узел не получил указанных выше подписей для проверки в промежутке времени $[t^{'} + \Delta t, t^{'} + \Delta t + \lambda]$, где λ – время задержки передачи информации, то узел с номером m сочтёт консенсус недостигнутым и выберет новое множество $B_{n^{''}}$ на основе старого состояния d, применяя (3).

Построение функции f

Вычислим двойной хеш от (9), обозначив полученное число ν. Построим псевдослучайную битовую последовательность вида:
$$\nu_1 = H(H(d)),\ \nu_2 = H(H(d+1)),\ \ldots \tag{10}$$
Получим следующую битовую запись:
$$R = \nu_1 \parallel \nu_2 \parallel \ldots \tag{11}$$
Битовую запись (11) разделяем последовательно без пропусков и перекрытий на кортежи по r бит в каждом, из которых строим множество номеров j_k узлов из $B_{n'}$, $k = 1, \ldots, n^{'}$, $1 \leq j_k \leq N$, причем $\hat{k} \triangleq 1$, вследствие

чего мастер-узлом всегда будет узел, номер которого сформируется первым. Если случайно повторится уже полученный номер j_k, то повторно полученное число пропускается.

Выводы

Предложенный алгоритм sdBFT по мнению авторов должен обладать более высоким быстродействием по сравнению с BFT алгоритмами. Измененяя мощность множества $B_{n'}$, можно будет управлять скоростью создания новых блоков, другими словами – скоростью работы алгоритма.

Потенциально большое число участников консенсуса усложняет предварительный сговор, когда группа голосующих узлов формирует новый блок, управляя составом блока по своему усмотрению, так как при следующем установлении консенсуса будет выбрано другое множество $B_{n'}$ голосующих узлов.

Псевдослучайный выбор множества голосующих узлов $B_{n'}$ по мнению авторов не позволит оказать существенного влияния на выбор узлов при следующем голосовании.

В дальнейшем авторы предполагают получить экспериментальное подтверждение теоретических положений, изложенных в этой статье, оценить скорость работы алгоритма sdBFT, исследовать возможные блокировки сети блокчейн.

References

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

[2] (2018.01.10). “Transactions Speeds: How Do Cryptocurrencies Stack Up To Visa or PayPal?”. https://howmuch.net/articles/crypto-transaction-speeds-compared .

[3] 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 .

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

[6] National Institute of Standards and Technology. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf.

[7] S. Josefsson, I. Liusvaara. “Edwards-Curve Digital Signature Algorithm (EdDSA)”. IETF RFC. https://tools.ietf.org/html/rfc8032 .