SOI: 1.1/TAS DOI: 10.15863/TAS

International Scientific Journal

Theoretical & Applied Science

p-ISSN: 2308-4944 (print) **e-ISSN:** 2409-0085 (online)

Year: 2019 **Issue:** 05 **Volume:** 73

Published: 27.05.2019 **http://T-Science.org**

SECTION 4. Computer science, computer engineering and automation.
UDC 004.

QR – Issue QR – Article

Oleg Yurievich Sabinin
Candidate of Engineering Sciences, Associate Professor
Peter the Great St. Petersburg Polytechnic University
olegsabinin@mail.ru

Mikhail Vladimirovich Toporov
student
Peter the Great St. Petersburg Polytechnic University
tmv1995@gmail.com

BUILDING A COMPOSITION OF CONSENSUS ALGORITHMS FEDERATED BYZANTINE AGREEMENT AND PROOF OF STAKE

Abstract: *This article discusses the theoretical composition of the two consensus algorithms in order to obtain a fundamentally new solution to the problem of consensus in a distributed ledger technologies.*

Key words: *Distributed Ledger Technologies, Consensus Algorithms, Federated Byzantine Agreement, Proof of Stake.*

Language: *Russian*

Citation: Sabinin, O. Y., & Toporov, M. V. (2019). Building a composition of consensus algorithms federated byzantine agreement and proof of stake. *ISJ Theoretical & Applied Science, 05 (73)*, 335-343.

Soi: http://s-o-i.org/1.1/TAS-05-73-49 ***Doi***: crossref https://dx.doi.org/10.15863/TAS.2019.05.73.49

ПОСТРОЕНИЕ КОМПОЗИЦИИ АЛГОРИТМОВ КОНСЕНСУСА FEDERATED BYZANTINE FAULT TOLERANCE И PROOF OF STAKE

Аннотация: *В данной статье рассматривается теоретическое построение композиции двух алгоритмов консенсуса с целью получения принципиально нового решения проблемы консенсуса в технологии распределенного реестра.*

Ключевые слова: *Распределенный реестр, Консенсус, Алгоритмы, Federated Byzantine Agreement, Proof of Stake*

1 Introduction

Если генерализировать проблемы блокчейн и распределенного реестра, то можно выделить одну наиболее явную проблему: в каждой из представленных на рынке распределенных систем требуется надежный алгоритм голосования, который обеспечит принятие решения в любом конкретном случае и при любых условиях. Эта категория алгоритмов получила название алгоритмов консенсуса, и их деятельность направлена именно на организацию голосования в распределенной среде.

Если проанализировать существующие на данный момент алгоритмы консенсуса, такие как Proof of Work[1], Proof of Stake[2] со стороны блокчейн и алгоритмы Raft[3], Paxos[4] со стороны распределенного реестра, то можно сделать вывод о том, что данные алгоритмы не являются законченными, иными словами, каждый из них обладает как своими достоинствами, так и недостатками, но при этом ни один из них не закрывает вопрос об организации голосования в распределенной среде[5].

В данной статье будет рассмотрено теоретическое построение композиции двух алгоритмов консенсуса. Выбранные для построения композиции алгоритмы являются представителями алгоритмов консенсуса хоть и смежных, но все же существенно отличающихся сфер применения. Один из них, алгоритм Proof of Stake, уже доказал на практике свою недееспособность и был заменен на Delegated

Proof of Stake[6–8]. При этом данный алгоритм реализован таким образом, что благодаря системе ставок появилась возможность выделять наиболее доверенные группы среди прочих неравных.

Эта особенность позволила данному алгоритму занять свою нишу в основе строения внутренних банковских криптовалют, поскольку банки, по сути своей, являются монополистами выпущенной криптовалюты, то, соответственно, не желают терять и контроль над ней. Схожие методы были применены и в прочих системах, подвластных, в основном, государственным органам различных стран. Как только начал разгораться интерес к криптовалюте, многие компании тогда ассоциировали свое будущее развитие и развитие электронной коммерции с криптовалютой. Однако же, когда бизнесу стало известно о том, что всегда есть вероятность потери контроля, интерес сразу же иссяк.

Точно такое же суждение можно отнести и к технологии распределенного реестра. Каждый архитектор, который проектирует гибкую автономную архитектуру, всегда желает удерживать контроль над ней, что также входит и в сферу интересов бизнеса. Все представленные алгоритмы консенсуса на момент написания данной работы базируются на принципе полного равноправия всех участников. В тоже время именно алгоритм Proof of Stake стал первым, который от этого принципа отошел[1].

Второй выбранный алгоритм – Federated Byzantine Agreement[9]. В данном алгоритме ключевая идея состоит в том, чтобы организовать голосование таким образом, что участники голосуют внутри доверенных групп, и только затем уже группы голосуют между собой.

В практических решениях задачи византийских генералов, которые были представлены ранее, требовалось участие каждого узла по каждому решению для достижения кворума. В случае Federated Byzantine Agreement каждый участник может решить, кому доверять, и стать частью доверенной группы принятия решений, которую можно назвать кворумным срезом.

Общее соглашение достигается в данном алгоритме за счет того, чтобы данные кворумные срезы, или же доверенные группы, специально пересекаются друг с другом. Таким образом, при определенном соотношении кворумных срезов к общему числу участников алгоритм гарантирует, что решение будет принято всегда[10]. Затем решение распространяется на всех участников голосования без исключений[9].

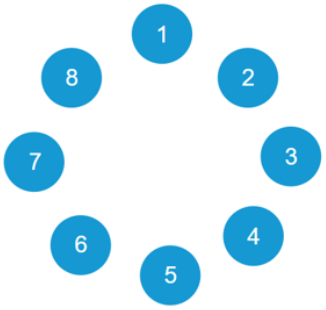
2 Composition building

Первым делом для построения композиции необходимо определить то, как будут задаваться участники и как именно они будут разделяться на кворумные срезы. Как правило, результатом работы алгоритмов консенсуса является хеш, который был вычислен одним из участников и принят остальными как результирующий, если рассматривать алгоритмы консенсуса со стороны блокчейн. Со стороны распределенного реестра решение всегда принимает некоторый лидер, поэтому результатом консенсуса является решение о том, кто именно будет лидером.

Поскольку суть консенсуса – достижение согласия по определенному вопросу, то в данной работе алгоритм будет возвращать бинарный согласованный ответ. Это позволит упростить понимание работы алгоритма, а также позволит более подробно раскрыть его суть.

Для инициализации участников будет использоваться конфигурационный файл. Это позволит провести более обширные исследования работы реализованного алгоритма, поскольку появится возможность вручную воздействовать на процесс голосования, заранее определяя патовые ситуации, в которых консенсус будет труднодостижим.

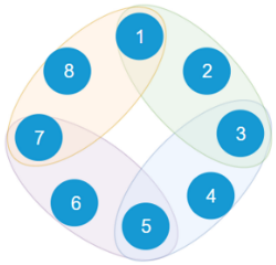
Как показано на рисунке 1, на вход алгоритму подается 8 участников. Четное число позволяет определить в конфигурации такую ситуацию, когда участники разделяются на две одинаковые группы с равным весом голосов.



Номер Участника	*Ставка*
1	2
2	1
3	1
4	4
5	1
6	3
7	1
8	1

Рисунок 1 - Инициализация участников

Первое, что решается в алгоритме – это выбор кворумных срезов и разделение на них. Поскольку мы строим композицию с Proof of Stake, то определим распределение участников не согласно их количеству, а согласно их ставкам, как представлено на рисунке 2:



Номер Участника	*Ставка*	*Срез*
1	2	1, 4
2	1	1
3	1	1, 2
4	4	2
5	1	2, 3
6	3	3
7	1	3, 4
8	1	4

Рисунок 2 - Разделение участников на кворумные срезы

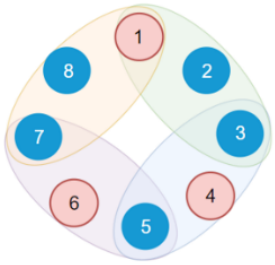
Таким образом мы получаем следующие кворумные срезы:

$$Q(1) = \{\{1, 2, 3\}\}$$
$$Q(2) = \{\{3, 4, 5\}\}$$
$$Q(3) = \{\{5, 6, 7\}\}$$
$$Q(4) = \{\{7, 8, 1\}\} \quad (1)$$

Которые вместе составляют следующую кворумную систему:

$$Q = \{\{1, 2, 3\}, \{1, 2, 3\ ,4, 5\}, \{5, 6, 7\}, \{1, 2, 3, 4, 5, 6, 7\}, \{5, 6, 7, 8, 1\}, \{1, 2, 3, 4, 5, 6, 7, 8\}\} \quad (2)$$

Следующим этапом выполним пересечение всех кворумных срезов. Данный шаг необходим, поскольку позволяет сохранить работоспособность алгоритма при композиции с алгоритмом Proof of Stake. Поскольку для второго этапа алгоритма требуется также определить лидеров каждого кворумного среза, то данное пересечение будет как раз состоять из лидеров каждого кворумного среза. В данной реализации алгоритма лидер будет выбираться не случайным образом, как это реализовано в алгоритме Federated Byzantine Agreement[9], а по максимальному значению ставки, что продемонстрировано на рисунке 3:



Номер Участника	*Ставка*	*Срез*
1	2	1, 4, 5
2	1	1
3	1	1, 2
4	3	2, 5
5	1	2, 3
6	3	3, 5
7	1	3, 4
8	1	4

Рисунок 3 - Выделение лидеров кворумных срезов

В таком случае у нас добавляется новый кворумный срез, который является пересечением ранее определенных кворумных срезов. Стоит отдельно отметить, что участник под номером 1 и так находился на пересечении двух кворумных срезов. Данное положение позволяет ему выступать представителем сразу этих двух кворумных срезов. В результате пересечения кворумных срезов образуется новый кворумный срез, и система кворумных срезов приобретает следующий вид:

$$Q(1) = \{\{1, 2, 3\}\}$$
$$Q(2) = \{\{3, 4, 5\}\}$$
$$Q(3) = \{\{5, 6, 7\}\}$$
$$Q(4) = \{\{7, 8, 1\}\}$$
$$Q(5) = \{\{1, 4, 6\}\} \quad (3)$$

В таком случае новая кворумная система имеет вид:

$$Q = \{\{1, 2, 3\}, \{1, 2, 3\ ,4, 5\}, \{5, 6, 7\}, \{1, 2, 3, 4, 5, 6, 7\}, \{5, 6, 7, 8, 1\}, \{1, 2, 3, 4, 5, 6, 7, 8\}, \{1, 4, 6\}\} \quad (4)$$

Сам процесс голосования проходит в два глобальных этапа. На первом этапе происходит голосование внутри кворумного среза. Результат этого голосования в обязательном порядке распространяется на всех участников среза. Далее происходит результирующее голосование между кворумными срезами. Именно для этого этапа и

выделялся последний кворумный срез, который будет определять лидеров или же представителей каждого кворумного среза. В алгоритме Federated Byzantine Agreement нет подобного этапа, поскольку изначально алгоритм был адаптирован под блокчейн, поэтому лидеры выбирались неявно случайным образом на уровне работы протокола, в рамках стратегии «кто первый откликнулся». В рамках построения композиции данный этап необходим как для имплементации Proof of Stake, так и в рамках адаптации алгоритма для достижения консенсуса в технологии распределенного реестра. Если вернутся к алгоритмам именно распределенного реестра, таким как Raft и Practical Byzantine Fault Tolerance[3,4], то видно, что на определенном этапе формируется список лидеров или же представителей каждой микрогруппы в целях принятия единого согласованного решения. Интересно и то, что к такому же решению пришли и в компании Etherium, когда реализовали Proof of Stake в протоколе Casper[7].

Для успешного прохождения первого этапа голосования внутри каждого кворумного среза участники голосуют согласно идее алгоритма Proof of Stake. Иными словами, каждый участник делает ставку на то, что считает верным. Проиллюстрировать это можно на примере одного из срезов, как показано на рисунке 4:

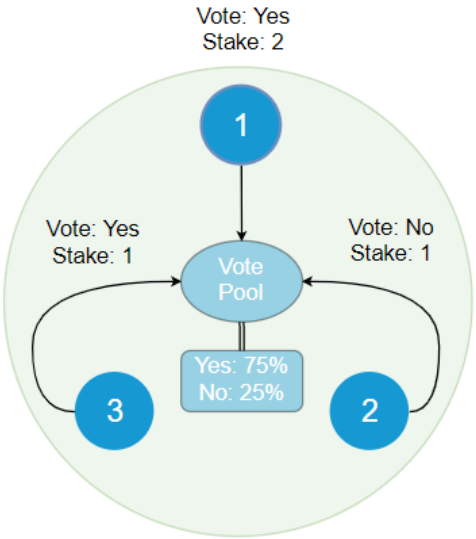
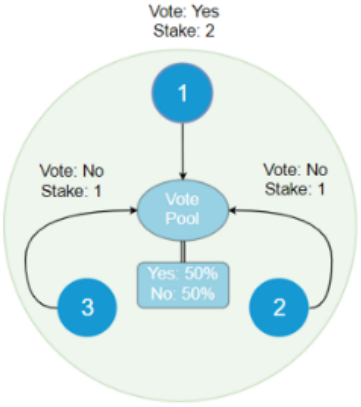


Рисунок 4 - Выделение лидеров кворумных срезов

На данном этапе необязательно достижение консенсуса внутри кворумного среза. Как уже рассматривалось ранее, в Federated Byzantine Agreement подразумевается исход, при котором кворумный срез может и не достигать консенсуса[9]. В таком случае общий консенсус по всем участникам все равно будет достигнут, что будет продемонстрировано позднее. Предположим, что возникла именно такая ситуация в одном или нескольких кворумных срезах, как показано на рисунке 5:



Номер Участника	*Ставка*	*Голос*
1	2	Yes
2	1	No
3	1	No

Рисунок 5 - Патовая ситуация при голосовании

Поскольку кворумные срезы пересечены между собой, то независимо от итогов голосования внутри кворумного среза исключается ситуация возникновения патовой ситуации в конце работы алгоритма. Рассмотрим более подробно то, как именно работает пересечение. Пусть имеются два кворумных среза, как показано на рисунке 6:

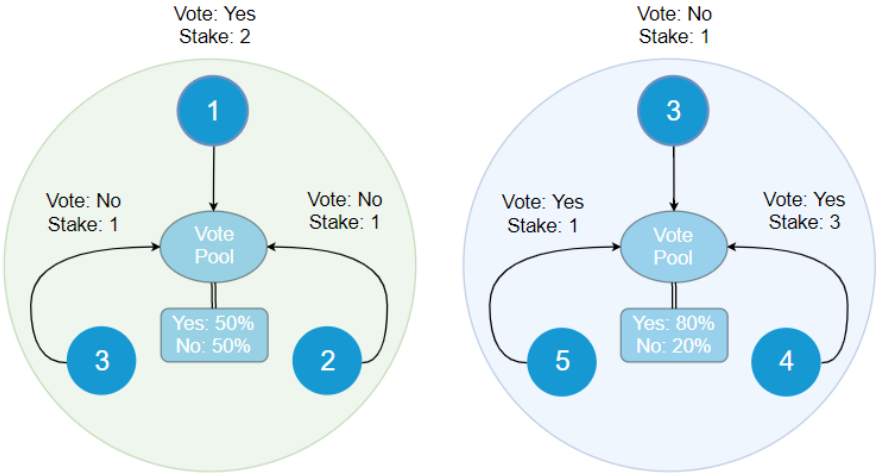
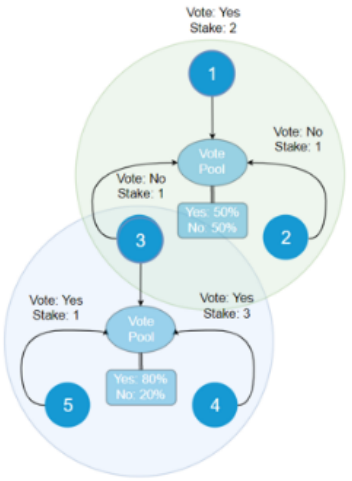


Рисунок 6 - Два кворумных среза

В случае, если кворумные срезы не пересекаются, то согласно идее работы алгоритма Federated Byzantine Agreement, данные срезы будут работать как две независимые друг от друга системы[9]. Однако же, если при составлении кворумных срезов соблюдалось условие пересечения, то можно получить следующий вид системы:

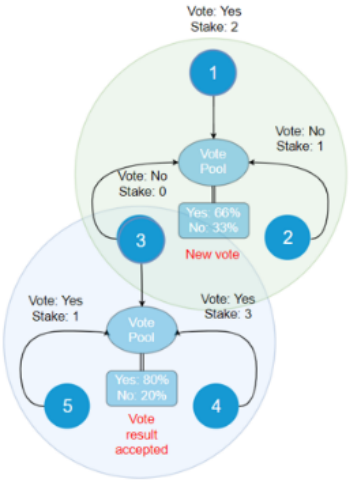


Номер Участника	*Ставка*	*Голос*
1	2	Yes
2	1	No
3	1	No
4	3	Yes
5	1	Yes

Рисунок 7 - Пересечение двух кворумных срезов

На рисунке 7 видно, что участник под номером 3 принадлежит сразу нескольким кворумным срезам. В данном случае этот участник попадает в патовую ситуацию при голосовании внутри первого кворумного среза, однако же, во втором кворумном срезе четко видно результирующий ответ. Согласно работе алгоритма Federated Byzantine Agreement, данный участник будет вынужден принять решение, которое будет принято во втором кворумном срезе[9]. Поскольку данный участник голосовал против данного решения, то его ставка сгорает, согласно идее алгоритма Proof of Stake [1], а значит, что и в первом кворумном срезе будет смещен баланс при сгорании его ставки, как только участник номер 3 оповестит свой кворумный срез об изменениях в его ставке, тем самым перезапустив голосование. В тоже самое время новое голосование внутри второго кворумного среза не вызывается, это

обеспечивается тем, что при успешном голосовании все участники второго кворумного среза изменяют свое состояние на accept.



Номер Участника	*Ставка*	*Голос*
1	2	Yes
2	1	No
3	-	-
4	3	Yes
5	1	Yes

Рисунок 8 - Переголосование в случае изменения ситуации

Однако же, наличие такого пересечения не гарантирует того, что консенсус будет достигнут. Существуют ситуации, когда участник, по которому происходит пересечение двух кворумных срезов является «поврежденным». В таком случае два кворумных среза будут работать независимо, что может привести к неверному результату работы алгоритма. Однако же при нормальной работе алгоритма гарантируется то, что в конце первого этапа голосования может остаться не более одного неопределившегося кворумного среза.

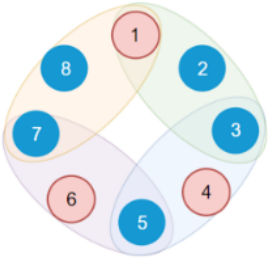
Данный случай разрешается двумя различными способами:

1.Создание кворумного среза, в который войдут представители всех кворумных срезов.

2.Второй этап голосования, при котором представители кворумных срезов будут голосовать между собой, а ставки участников, которые попадают в пересечение, становятся несгораемыми.

Как в первом, так и во втором случае, проблема с поврежденным участником на пересечении кворумных срезов разрешается. При этом данные этапы алгоритма также используются для построения композиции с Proof of Stake, а, следовательно, проблема разрешается без добавления лишних действий и проверок в алгоритм.

Как только первый этап завершается, происходит подсчет ставок. На данном этапе каждый кворумный срез проголосовал, и результаты можно представить в следующем виде:



Номер Участника	*Ставка*	*Срез*	*Голос*
1	2	1, 4, 5	Yes
2	1	1	No
3	1	1, 2	No
4	3	2, 5	Yes
5	1	2, 3	Yes
6	3	3, 5	Yes
7	1	3, 4	Yes
8	3	4	No

Рисунок 9 - Выделение лидеров кворумных срезов

Согласно идее алгоритма Proof of Stake [1], участники кворумного среза делают ставки в соответствии со своими возможностями. Далее ставки подсчитываются, и выбирается тот вариант ответа, на который было больше поставлено. В стандартной реализации блокчейн участники, которые сделали ставки на проигравший вариант, спонсируют работу победителей. В данной композиции, которая реализуется для технологии распределенного реестра, проигравшие ставки

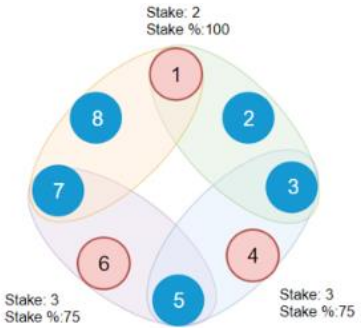
сгорают, а сумма выигрышной ставки становится ставкой выбранного представителя данного кворумного среза.

Таким образом, после первого этапа голосования, где участники голосуют исключительно внутри своих кворумных срезов, можно видеть такую картину:

Таблица 1. Состояние участников голосования

Номер Среза	Участники	Результат	Ставка
1	1, 2, 3	Yes	2
2	3, 4, 5	Yes	4
3	5, 6, 7	Yes	4
4	7, 8, 1	Yes/No	3/3
5	1, 4, 6	Yes	8

Как можно заметить, для кворумного среза под номером 4 получилась патовая ситуация, несмотря на пересечение. Данная конфигурация встречается крайне редко и была подобрана неслучайно, чтобы показать, как алгоритм справляется с подобными задачами. Стоит отдельно отметить, что, как и во всех алгоритмах консенсуса распределенного реестра, здесь сразу обозначаются представители каждого кворумного среза. В данной реализации представителем или же лидером выступает тот, чья процентная доля в ставке была больше всех, именно эти представители и формируют 5-ый кворумный срез, участники которого были обозначены красным цветом на следующем рисунке:

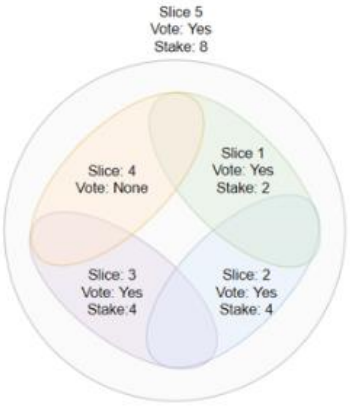


Номер Участника	*Ставка*	*%*
1	2	100
4	3	75
6	3	75

Рисунок 10 - Процентная ставка лидеров

После первого этапа голосования все участники всех кворумных срезов имеют некоторое решение, которое считается безусловно единогласным внутри их кворумного среза. Также для каждого кворумного среза был выбран представитель, который будет принимать участие во втором этапе голосования, а его ставка определяется как суммарная ставка победителей голосования внутри кворумного среза.

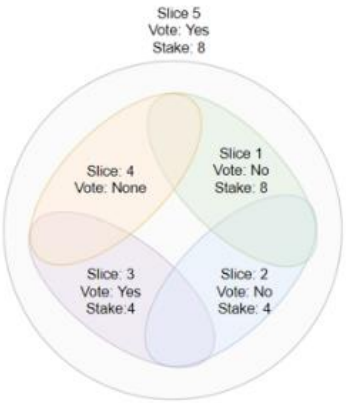
На втором этапе голосования представители каждого кворумного среза делают свои ставки на то, какой вариант ответа считают правильным. Далее осуществляется подсчет результатов и, в случае достижения консенсуса, решение распространяется на всех участников голосования:



Номер Среза	*Ставка*	*Голос*
1	2	Yes
2	4	Yes
3	4	Yes
4	-	None
5	8	Yes

Рисунок 11 - Голосование между кворумными срезами

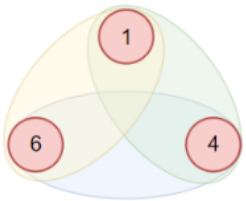
Однако же на данном этапе также возможна патовая ситуация, когда ставки уравниваются. В таком случае выбранные представители кворумных срезов разделяются на дополнительные кворумные срезы и повторяют голосование уже между собой, начиная с первого этапа. Промоделируем данную ситуацию, искусственно изменив некоторые ставки:



Номер Среза	*Ставка*	*Голос*
1	8	No
2	4	No
3	4	Yes
4	-	None
5	8	Yes

Рисунок 12 - Моделирование патовой ситуации

Если на втором этапе решение не может быть принято, то из лидеров первого этапа формируется новая система кворумных срезов.



Номер Участника	*Ставка*	*Срез*	*Голос*
1	8	1, 2	No
4	4	2, 3	No
6	4	3, 1	Yes

Рисунок 13 - Голосование между лидерами

Определим новые кворумные срезы:

$$Q(1) = \{\{1,4\}\}$$
$$Q(2) = \{\{4,6\}\}$$
$$Q(3) = \{\{6,1\}\} \quad (5)$$

И, исходя из новых кворумных срезов, выстраивается новая кворумная система:

$$Q = \{\{1,2\},\{1,2,3\},\{3,1\},\{1,2,3\}\} \quad (6)$$

После первого этапа голосования получаем следующий расклад:

Таблица 2. Результаты голосования на втором этапе

Номер Среза	Участники	Результат	Ставка
1	1, 4	No	12
2	4, 6	No	4
3	6, 1	Yes	4

Как можно видеть, второй кворумный срез изначально также попадает в патовую ситуацию, однако, благодаря пересечению, ставка участника под номером 4 сгорает, и устанавливается однозначный ответ.

Данный подход гарантирует то, что решение будет принято в любом случае. Поскольку на финальной итерации в любом случае остается всего два кворумных среза, если решение, конечно, не было принято до этого. Эти два кворумных среза, согласно условиям деления[9] будут иметь как минимум одного участника, общего для обоих кворумных срезов, что в свою очередь исключает патовую ситуацию, при которой ставки двух противоборствующих сторон оказываются равными. Как можно видеть, решение будет принято при любом исходе.

3 Conclusion

В данной работе было рассмотрено теоретическое построение композиции двух алгоритмов консенсуса Proof of Stake и Federated Byzantine Agreement.

Также были рассмотрены и некоторые нестандартные ситуации, при которых полученный алгоритм способен сохранять свою работоспособность.

На данный момент не существует алгоритма консенсуса децентрализованной системы, который бы гарантировал стабильность платформы, защищенность данных и при этом был бы максимально унифицированным. За решением данной проблемы исследователи обратились к более старым методам, описанных еще в начале 2000-х годов на математическом языке. Как показала практика с алгоритмами Raft & Paxos[3], реализация этих методов зачастую имеет очень высокий потенциал.

Многие алгоритмы по данному направлению появились лишь в последние несколько лет, однако все еще не нет однозначного решения по поводу того, как достигать консенсуса в распределенной среде.

References:

1. (2015). *BitFury Group. Proof of Stake versus Proof of Work* // BitFury Gr.
2. Nadal, S., & King, S. (2012). *PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake* [Electronic resource] // PeerCoin. Retrieved 2019, from https://peercoin.net/assets/paper/peercoin-paper.pdf
3. Lamport, L., et al. (2004). *In Search of an Understandable Consensus Algorithm* // Proc. 2014 USENIX Annu. Tech. Conf.
4. Lamport, L. (2005). *Generalized Consensus and Paxos.* Microsoft Research Technical Report MSR-TR-2005-33.
5. Hammerschmidt, C. (2018). Consensus in Blockchain Systems [Electronic resource]. Retrieved 2019, from https://medium.com/@chrshmmmr/consensus-in-blockchain-systems-in-short-691fc7d1fefe
6. Fan, X., & Chai, Q. (2017). *Roll-DPoS: A Randomized Delegated Proof of Stake Scheme for Scalable Blockchain-Based Internet of Things Systems* // IoTeX.
7. Buterin, V., & Griffith, V. (2017). *Casper the Friendly Finality Gadget* // Ethereum Found.
8. Wüst, K., & Gervais, A. (2017). *Do you need a Blockchain?*. IACR Cryptol. ePrint Arch.
9. Mazieres, D. (2016). *Stellar Consensus Protocol*. p. 32.
10. Garcia-Perez, A., & Gotsman, A. (2016). *Federated Byzantine Quorum Systems* // IMDEA Softw. Institute, Madrid, Spain.