SWIFT BLOCK

ADVANCED NETWORK

БЕЛАЯ КНИГА СИФАН БИНЬ

Содержание

Аннотация

SwiftBlock Coin — это принципиально новая криптовалюта, ориентированная на токенизацию трафика. Благодаря децентрализованной токенизации трафика SwiftBlock Coin обеспечивает прозрачный, безопасный и эффективный метод транзакции.

Основанная на протоколе консенсуса SwiftBlock (SCP), SwiftBlock Coin использует структуру децентрализованной автономной организации (DAO), обеспечивающую честное регулирование и развитие, направляемое сообществом. Уникальный механизм вознаграждения за рефералов способствует росту числа пользователей, стимулирует распределение вознаграждения и повышает активность экосистемы.

1. Введение

1.1. Введение в токенизацию трафика

SwiftBlock Coin - это принципиально новая криптовалюта, ориентированная на токенизацию трафика и отходящая от традиционных методов монетизации трафика. Интегрируя токенизацию трафика с технологией блокчейн, компания создает новые каналы для монетизации трафика.

1.2. Текущее состояние и будущие ожидания рынка монетизации трафика

Согласно последним данным маркетинговых исследований, в 2024 году рыночная стоимость мировой рекламной индустрии впервые превысила отметку в 1 трлн долларов. В этом пространстве доминирует цифровая реклама, на долю которой, как ожидается, к 2025 году будет приходиться 73% мировых расходов на рекламу. На китайском рынке объем рынка цифровой рекламы в 2024 году достиг миллиарда юаней, и, по прогнозам, к 2029 году среднегодовой темп роста (CAGR) составит более 15%.

1.2.1. В настоящее время глобальный рынок монетизации интернет-трафика в основном приносит доход за счет рекламы, подписки членов и транзакций с данными. Из-за широкого охвата компании и ее вовлеченности в различные отрасли промышленности сложно точно оценить ее общую рыночную стоимость.

1.2.1.1. Рынок влиятельной экономики Китая: По данным информационного агентства Синьхуа, в 2022 году объем рынка влиятельной экономики Китая достиг 1,3 трлн юаней, увеличившись на 26,9% в годовом исчислении. Ожидается, что в 2023 году он превысит 1,6 трлн юаней, а рост в годовом исчислении составит 23,8%.

1.2.1.2. Китайский рынок коротких видеороликов и потокового вещания: Согласно "Статистическому бюллетеню Национальной индустрии радио и телевидения за 2023 год", выручка от коротких видеороликов, потокового вещания и других онлайн-аудиовизуальных компаний в Китае в 2022 году превысила 420 млрд юаней, увеличившись на 33,39% в годовом исчислении.

1.2.2. Заглядывая в будущее, ожидается, что с развитием блокчейна,

искусственного интеллекта и технологий Web3.0 на рынке монетизации трафика появятся новые возможности для роста. Децентрализованная торговля трафиком, усиленная защита конфиденциальности и приложение смарт-контрактов еще больше увеличат объем рынка.

1.3. Актуальные проблемы на рынке монетизации трафика

1.3.1. **Доминирующее положение на рынке крупных компаний (Монополия)** – Монетизация интернет-трафика в основном контролируется технологическими гигантами, такими как Google, Meta (Facebook), Alibaba и Tencent, что приводит к высокому уровню централизации монополии.

В настоящее время большинство создателей контента полагаются на эти крупные компании или платформы социальных сетей для монетизации трафика, но некоторые проблемы сохраняются, например:

- **Монополия платформы и высокие комиссионные ставки**: Значительный объем данных и пользователей сосредоточен на известных платформах, что вынуждает многих создателей использовать эти платформы, которые взимают высокие комиссионные сборы из-за своей популярности.
- **Высокий порог монетизации и острая конкуренция**: Индивидуальные создатели и небольшие веб-сайты с трудом получают существенный доход от монетизации трафика, поскольку только создатели с высоким трафиком получают значительные выплаты.
- **Жесткая конкуренция за трафик на развивающихся платформах**: Рекламодатели предпочитают инвестировать в трафик высокого уровня, что затрудняет создание партнерских отношений для небольших компаний.
- **Длительные циклы расчетов**: Например, для Google Ad-Sense требуется цикл платежей продолжительностью более 30 дней, что влияет на денежные потоки владельцев небольшого трафика.

1.3.2. **Мошенничество с трафиком и поддельный трафик**

- **Трафик ботов**: Рекламодатели платят за трафик, но на самом деле он может поступать от ботов или скриптов, что приводит к низким показателям конверсии рекламных кампаний.
- **Мошенничество с кликами**: Нечестные конкуренты или мошенники используют автоматизированные программы для кликов по объявлениям, что приводит к значительным убыткам для рекламодателей.

1.3.3. **Отсутствие прозрачности и несправедливые расчеты**

- Рекламодатели не могут отследить истинный источник трафика и полагаются на предоставленные платформой данные, которые могут быть фальсифицированы или вводить в заблуждение.

1.3.4. **Проблемы конфиденциальности и неправомерное использование пользовательских данных**

- Централизованные платформы собирают огромные объемы пользовательских данных и используют алгоритмы искусственного интеллекта для таргетированной рекламы, что серьезно нарушает конфиденциальность пользователей.
- Пользователи не имеют никакого контроля над своими данными, и личная информация продается сторонним рекламным компаниям.

1.3.5. **Недооценка трафика**

- Нынешний рынок трафика в основном основан на цене за клик (CPC) и цене за тысячу показов (CPM), но он не позволяет эффективно измерить истинный вклад пользователей.
- Данные о поведении пользователей (такие как время ожидания и скорость взаимодействия) монетизируются неэффективно, что не позволяет поставщикам трафика получать справедливую компенсацию.

1.4. **Что такое токенизация трафика?**

- Суть токенизации трафика заключается в преобразовании трафика в свободно обращаемые токены на блокчейне. Каждый токен представляет права собственности на трафик поставщика трафика. Технология блокчейн позволяет этому трафику стать цифровым активом, обеспечивая безопасную, прозрачную и бесперебойную глобальную торговлю.
- Токенизация трафика отличается от традиционных методов монетизации трафика, обеспечивая лучшую ликвидность для создателей контента и поставщиков трафика. По сравнению с традиционными методами, это значительно снижает порог монетизации, решая проблему несправедливого заработка для мелких создателей и обычных людей, позволяя участвовать большему количеству людей.

1.5. **Блокчейн SwiftBlock Coin и системная архитектура**

Сеть SwiftBlock — это открытый блокчейн с открытым исходным кодом, основанный на протоколе консенсуса SwiftBlock (SCP), механизме Доказательства согласия (PoA). Сеть SwiftBlock Coin, основанная на SCP, работает быстрее, дешевле и энергоэффективнее, чем многие другие блокчейны. Транзакции завершаются и добавляются в блокчейн, как только компьютеры, известные как "узлы", достигают консенсуса с помощью протокола консенсуса SwiftBlock. Любой желающий может создать узел SwiftBlock и принять в нем участие, но он должен предоставить свою идентификационную информацию в общедоступной записи. Это позволяет другим узлам решать, кого включать или исключать из своих доверенных групп. SCP достигает консенсуса посредством серии процессов голосования между этими взаимно доверенными узлами. Когда достаточное количество узлов в доверенной

перекрывающейся группе (известной как кворум) соглашаются с тем, что набор транзакций и связанные с ними активы и операции являются действительными, они навсегда добавляются в блокчейн. Обычно этот процесс занимает около 5 секунд.

1.5.1. **Ключевые особенности SCP**

SCP — это первый доказуемо безопасный механизм достижения консенсуса, обладающий четырьмя ключевыми характеристиками:

- **Децентрализованный контроль**: Участвовать может любой желающий, и нет центрального органа, который мог бы определить, чье одобрение необходимо для достижения консенсуса.
- **Низкая задержка**: Узлы могут достичь консенсуса в течение времени, ожидаемого для веб-транзакций или платежных операций, как правило, в течение нескольких секунд.
- **Гибкое доверие**: Пользователи могут свободно доверять любой комбинации сторон, которую они сочтут подходящей. Например, небольшая некоммерческая организация может сыграть решающую роль в обеспечении честности крупных учреждений.
- **Асимптотическая защита**: Защита основана на цифровых подписях и хэш-семействах, параметры которых могут быть практически скорректированы для защиты от злоумышленников с невообразимо огромной вычислительной мощностью.

1.5.2. **Процесс консенсуса SCP**

Процесс консенсуса SCP разделен на четыре этапа, каждый из которых обеспечивает безопасность данных и их окончательность:

- **Предложение**: Узлы предлагают сети кандидатов на проведение транзакций, и все узлы обмениваются информацией.
- **Голосование**: Голосование проводится по срезам кворума для определения группы, достигшей консенсуса.
- **Подтверждение**: Как только транзакция получает достаточную поддержку, она переходит в состояние подтверждения, предотвращая форки.
- **Приложение**: Транзакция записывается в реестр, завершается и не подлежит изменению.

1.5.3. **Основные компоненты SCP**

- **Узлы**: Серверы, работающие по протоколу SCP, ответственные за проверку транзакций и достижение консенсуса.
- **Федеративное византийское соглашение (FBA)**: Консенсус достигается с помощью срезов кворума без необходимости полной сетевой синхронизации.

- **Срезы кворума**: Каждый узел независимо выбирает другие узлы, которым он доверяет, образуя сеть частичного доверия.
- **Реестр**: Записывает всю историю транзакций, аналогично блокчейну, но с более быстрым обновлением.
- **Ядро** SwiftBlock: основной компонент, который обрабатывает протокол SCP и поддерживает блокчейн.
- **Horizon API**: REST API для разработчиков для доступа к сети SwiftBlock, поддерживающий такие функции, как запросы транзакций и управление учетными записями.

1.5.4. Характеристики различных механизмов достижения консенсуса

mechanism	decentralized control	low latency	flexible trust	asymptotic security
proof of work	✓			
proof of stake	✓	maybe		maybe
Byzantine agreement		✓	✓	✓
Tendermint	✓	✓		✓
SCP (this work)	✓	✓	✓	✓

Существуют различия между протоколом консенсуса SwiftBlock (SCP) и другими механизмами консенсуса. Наиболее известным децентрализованным механизмом консенсуса является схема Доказательства работы (PoW), предложенная Биткойном. Биткойн использует двусторонний подход для достижения консенсуса. Во-первых, это создает стимулы для рациональных участников вести себя корректно. Во-вторых, он обрабатывает транзакции с помощью алгоритма Доказательства работы, который предназначен для предотвращения некорректной работы участников, не обладающих большей частью вычислительных мощностей системы.

Биткойн в полной мере продемонстрировал привлекательность децентрализованного консенсуса. Однако Доказательство работы также имеет свои ограничения. Во-первых, ресурсоемкость — Биткоин может потреблять столько же электроэнергии, сколько вся Ирландия. Во-вторых, ожидаемая задержка для безопасного проведения транзакций составляет порядка нескольких минут или десятков минут.

Наконец, по сравнению с традиционными криптографическими протоколами Доказательство работы не обеспечивает асимптотической безопасности. Что касается абсурдных взломщиков или тех, у кого есть внешние мотивы для взлома, даже небольшое вычислительное преимущество может свести на нет предположения безопасности, позволяя переписать историю в так называемую "атаку 51%." Хуже того, злоумышленник, изначально контролирующий менее 50% вычислительной мощности, может

манипулировать системой, предлагая непропорционально высокие вознаграждения тем, кто присоединится к нему.

Альтернативой Доказательству работы является Доказательство доли (PoS), где консенсус зависит от сторон, предоставивших гарантийное обеспечение. Как и в случае с Доказательством работы, награды поощряют рациональных участников соблюдать протокол, в то время как структура также наказывает за некорректное поведение. Доказательство доли открывает возможность для так называемых атак на основе "nothing-at-stake", когда стороны, которые ранее предоставили гарантийное обеспечение, но позже обналичили его и потратили, могут вернуться к тому моменту, когда у них все еще было гарантийное обеспечение, и переписать историю. Чтобы смягчить такие атаки, системы эффективно сочетают Доказательство доли с Доказательством работы — либо увеличивая объем требуемой работы пропорционально гарантийному обеспечению, либо откладывая возврат гарантийного обеспечения на срок, достаточный для того, чтобы другие согласованные механизмы установили неотменяемые контрольные точки.

Другим методом достижения консенсуса является Византийское соглашение, наиболее известным вариантом которого является PBFT (Отказоустойчивость к византийской ошибке). Византийское соглашение обеспечивает консенсус, несмотря на произвол (включая абсурдность) некоторых участников. Этот подход обладает двумя привлекательными свойствами. Во-первых, консенсус может быть достигнут быстро и эффективно. Во-вторых, доверие полностью отделено от владения ресурсами, что позволяет небольшим некоммерческим организациям помогать поддерживать честность более влиятельных организаций (таких как банки или CA). Однако дело осложняется требованием о том, что все стороны должны согласовать точный список участников. Кроме того, должны быть приняты меры для предотвращения многократного подключения злоумышленников и превышения отказоустойчивости системы, известных как атаки Сибиллы. BFT-CUP принимает неизвестных участников, но все же предполагает наличие централизованного механизма контроля допуска, устойчивого к Сибиллам. Как правило, членство в системах Византийского соглашения определяется центральным органом или закрытыми переговорами.

Tendermint использует другой подход, основывая членство на Доказательстве доли. Однако это снова связывает доверие с владением ресурсами.

SCP — это первый протокол Византийского соглашения, который предоставляет каждому участнику максимальную свободу в выборе того, каким комбинациям других участников доверять.

1.5.5. **Модель Федеративного Византийского соглашения (FBA)**

Модель Федеративного византийского соглашения (FBA), как и не федеративное Византийское соглашение, решает проблему обновления дублируемых состояний (например, реестров транзакций или деревьев сертификатов). Согласовав, какие обновления следует применять, узлы могут избежать противоречивых, непримиримых состояний. Каждое обновление идентифицируется уникальным слотом, из которого можно сделать вывод о зависимости между обновлениями. Например, слот может быть последовательно пронумерованной позицией в журнале, который применяется по порядку.

Система FBA использует протокол консенсуса, гарантирующий, что узлы согласуют содержимое слотов. Когда узел *v* безопасно применил обновления во всех слотах, от которых зависит *i*, он может безопасно применить обновление *x* в слоте *i*. Более того, он считает, что все должным образом функционирующие узлы в конечном итоге согласятся на (X) для слота *i*. На этом этапе мы говорим, что (v) вынесло во внешний файл *x* для слота *i*. Вынесенные во внешний файл значения могут вызвать необратимую реакцию со стороны внешнего мира, поэтому узлы впоследствии не смогут изменить свое мнение о них.

Проблема для FBA заключается в том, что злоумышленники могут подключаться многократно, превосходя числом настоящие узлы. Таким образом, традиционные системы кворума, основанные на большинстве голосов, не работают. Вместо этого FBA определяет кворумы децентрализованным образом, когда каждый узел выбирает то, что мы называем срезом кворума. В следующем подразделе определяются кворумы на основе срезов. В следующих подразделах приведены примеры и обсуждения. Наконец, мы определяем ключевые свойства безопасности и жизнеспособности, на достижение которых должен быть направлен протокол консенсуса.

1.5.5.1. **Срезы кворума**

В протоколе консенсуса узлы обмениваются сообщениями об объявлениях слотов. Мы предполагаем, что такие объявления не могут быть подделаны, что гарантируется, если узлы именуются открытыми ключами и они подписывают свои сообщения цифровой подписью. Когда узел слышит, что достаточное количество узлов утверждает объявление, он предполагает, что ни один должным образом функционирующий узел не будет противоречить объявлению. Мы называем это "достаточное количество узлов" срезом кворума или, более кратко, просто срезом. Чтобы обеспечить прогресс при наличии сбоев в работе узла, узел может иметь несколько срезов, любого из которых достаточно, чтобы убедить его принять объявление. Таким образом, на высоком уровне система FBA состоит из свободного объединения узлов, каждый из которых выбирает

один или несколько срезов.

Более точно: Система Федеративного византийского соглашения (FBAS) представляет собой пару <V, Q>, состоящую из набора узлов V и функции кворума Q: $\nu \rightarrow 2^{2V} \setminus \{\emptyset\}$, которая присваивает один или несколько срезов кворума каждому узлу. Узел принадлежит ко всем своим собственным срезам кворума, т.е. $\forall \nu \in V$, $\forall q \in Q(\nu)$, $\nu \in q$. (Примечание: 2^X обозначает множества всех подмножеств X.)

Определение (Кворум): множество узлов $U \subseteq V$ в FBAS <V, Q> является кворумом, если и только если $U \neq \emptyset$ и *U* содержит срез для каждого члена, т.е. $\forall \nu \in U$, $\exists q \in Q(\nu)$, то есть $q \subseteq У$.

Кворум — это набор узлов, достаточный для достижения соглашения. Срез кворума — это подмножество кворума, которое убеждает конкретный узел поверить в соглашение. Срезы кворума могут быть меньше, чем кворумы сами по себе. Рассмотрим систему с четырьмя узлами на рис. 2, где каждый узел имеет один срез, а стрелки указывают на других членов среза. Срез $\{\nu_1, \nu_2, \nu_3\}$ достаточен для того, чтобы ν_1 поверил заявлению. Однако срезы ν_2 и ν_3 включают ν_4, что означает, что ни ν_2, ни ν_3 не могут утверждать заявление без согласия ν_4. Следовательно, консенсус не может быть достигнут без участия ν_4, и единственным кворумом, включающим ν_1, является совокупность всех узлов $\{\nu_1, \nu_2, \nu_3, \nu_4\}$.

Традиционное не федеративное Византийское соглашение требует, чтобы все узлы принимали одни и те же срезы, т.е. $\forall \nu 1, \nu 2, Q(\nu_1) = Q(\nu_2)$. Поскольку каждый участник принимает каждый срез, традиционные системы не в состоянии провести различие между срезами и кворумами. Недостатком является то, что членство и кворумы должны быть каким-то образом определены заранее, что исключает открытое членство и децентрализованный контроль. Традиционные системы (например, PBFT) обычно состоят из $3f + 1$ узлов, где любые $2f + 1$ узла образуют кворум. Здесь f - максимальное количество Византийских неисправных узлов (т.е. узлов, ведущих себя произвольно), которое может выдержать система.

Ключевым нововведением FBA является предоставление возможности каждому узлу ν выбирать свой собственный независимый набор срезов кворума $Q(\nu)$. Таким образом, общесистемные кворумы возникают в результате независимых решений каждого узла. Узлы могут выбирать срезы на основе произвольных критериев, таких как репутация, финансовые условия и т.д. В некоторых случаях консенсус все еще может быть достигнут, даже если ни один узел не имеет полного представления обо всех узлах системы.

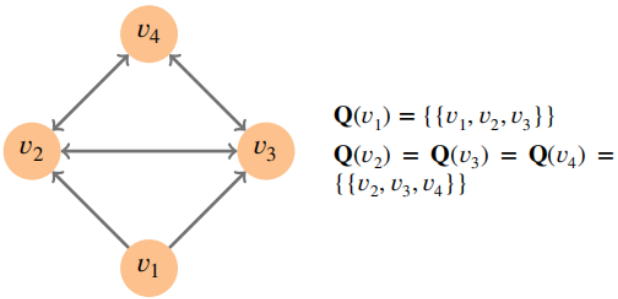

Рисунок 2: Без v 4 Срез Кворума v 1 не считался бы кворумом

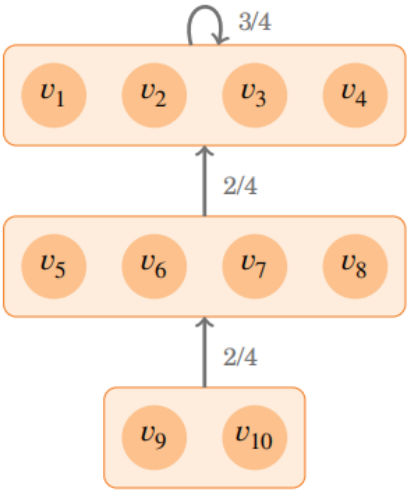

Рисунок 3: Пример иерархической структуры кворума

Верхний слой: Включает себя, с срезами, состоящими из 3 срезов: {v1, v2, v3, v4}

Средний слой: Срезы состоят из себя + любых двух узлов верхнего слоя

Листовой слой: Срезы состоят себя + любых двух узлов среднего слоя

1.5.5.2. Примеры и обсуждение

На рисунке 3 показан пример иерархической системы, в которой разные узлы имеют разные наборы срезов, что возможно только с помощью FBA. Верхний слой состоит из v1, ..., v4, структурированных аналогично системе PBFT с $f = 1$, что означает, что он может выдержать одну необъяснимую ошибку до тех пор, пока остальные три узла доступны и работают корректно. Узлы v 5, ..., v 8 образуют средний слой, который не зависит друг от друга, но опирается на верхний слой. Для формирования среза для узла среднего слоя необходимы только два узла верхнего слоя. (Верхний слой допускает не более одной необъяснимой ошибки, поэтому, если не происходит сбой всей системы, два узла верхнего слоя не могут выйти из строя одновременно.) Узлы v9 и v10 находятся в листовом слое, где срез

состоит из любых двух узлов среднего слоя. Обратите внимание, что v_9 и v_{10} могут выбирать непересекающиеся срезы, такие как $\{v_5, v_6\}$ и $\{v_7, v_8\}$; тем не менее, оба они будут косвенно зависеть от верхнего слоя.

На практике верхний слой может состоять от 4 до 12 хорошо известных и пользующихся доверием финансовых учреждений. По мере роста верхнего слоя мнения о его членстве могут не полностью совпадать, но у большинства заинтересованных сторон будет значительное совпадение в понимании верхнего слоя. Кроме того, можно было бы предусмотреть несколько средних слоев, например, по одному для каждой страны или географического региона.

Эта многоуровневая структура напоминает междоменную сетевую маршрутизацию. Современный интернет держится на отдельных пиринговых и транзитных связях между парами сетей. Ни один центральный орган не диктует эти условия и не является арбитром. Тем не менее, этих парных отношений оказалось достаточно, чтобы создать представление о фактических ISP первого уровня, хотя доступность Интернета действительно страдает из-за брандмауэров, транзитивная доступность почти полная - например, брандмауэр может заблокировать Нью-Йорк Таймс, но если это позволяет Google, а Google может получить доступ к Нью-Йорк Таймс, тогда Нью-Йорк Таймс транзитивно доступен. Транзитивная доступность может иметь ограниченную функцию для веб-сайтов, но она имеет решающее значение для консенсуса; аналогичным примером может служить то, что Google принимает заявления только в том случае, если это делает Нью-Йорк Таймс.

Если мы сравним срезы кворума с доступностью сети, а кворумы - с транзитивной доступностью, то почти полная транзитивная доступность Интернета предполагает, что мы можем аналогичным образом обеспечить глобальный консенсус с помощью FBA. Во многих отношениях достижение консенсуса проще, чем междоменная маршрутизация. В то время как транзит потребляет ресурсы и сопряжен с затратами, включение срезов требует только проверки цифровых подписей. Таким образом, узлы FBA могут допускать ошибки при включении, создавая консервативные срезы с более высокой взаимозависимостью и избыточностью, чем обычно наблюдается в пиринговых и транзитных системах.

Другим примером, невозможным при централизованном консенсусе, является циклическая структура зависимостей, как показано на рисунке. Маловероятно, что такие циклы возникают намеренно, но, когда отдельные узлы выбирают свои собственные срезы, система в целом может оказаться в циклах зависимостей. Что еще более важно, по сравнению с традиционным Византийским соглашением, протоколы FBA должны

учитывать различные структуры групп.

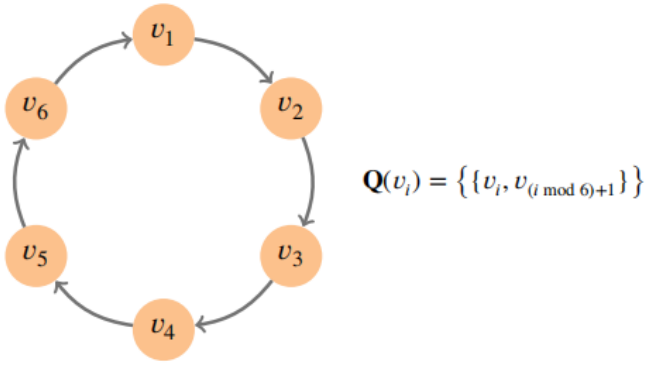

Рисунок 4: Пример структуры циклического кворума

1.5.5.3. Безопасность и жизнеспособность

Мы классифицируем узлы на корректно работающие и некорректно работающие. Корректно работающие узлы выбирают разумные срезы кворума и следуют протоколу, включая, в конечном счете, ответы на все запросы. С другой стороны, некорректно работающие узлы этого не делают. Некорректно работающие узлы могут сталкиваться с необъяснимыми ошибками, что означает, что их поведение является произвольным. Например, некорректно работающий узел может быть скомпрометирован, его владелец может злонамеренно модифицировать его программное обеспечение или он может выйти из строя.

Цель Византийского соглашения - обеспечить, чтобы корректно работающие узлы передавали согласованные значения даже в присутствии некорректно работающих узлов. Эта цель включает в себя два аспекта. Во-первых, мы хотим предотвратить расхождение между узлами и убедиться, что одно и то же значение выводится во внешний файл для одного и того же слота. Во-вторых, мы хотим обеспечить, чтобы узлы могли эффективно выводить значения во внешний файл, не заходя в тупик, когда консенсус не может быть достигнут. С этой целью мы вводим следующие два термина:

Определение (Безопасность): Набор узлов в FBAS считается безопасным, если никакие два узла не выводят во внешний файл разные значения для одного и того же слота.

Определение (Жизнеспособность): Считается, что узел в FBAS обладает жизнеспособностью, если он может выводить во внешний файл новые значения без участия каких-либо неисправных узлов (включая некорректно

работающие узлы).

Узлы, которые являются одновременно безопасными и активными, называются исправными узлами. Некорректно работающие называются неисправными. Все некорректно работающие узлы неисправны, но узлы корректно работающие также могут стать неисправными, если они будут бесконечно ждать сообщений от некорректно работающих узлов или, что еще хуже, если на их состояние повлияют некорректные сообщения от некорректно работающих узлов.

На рисунке 5 показаны типы неисправностей узлов, которые могут возникнуть. Слева изображены необъяснимые ошибки, которые являются некорректно работающими узлами. Справа представлены два типа корректно работающих, но неисправных узлов. Узлы, которым не хватает жизнеспособности, называются блокирующими узлами, в то время как узлы, которым не хватает безопасности, называются дивергентными узлами. Атаки, нарушающие безопасность, строго говоря, более серьезны, чем те, которые нарушают только жизнеспособность, поэтому мы рассматриваем дивергентные узлы как подмножество блокирующих узлов. Наше определение жизнеспособности относительно мягкое, поскольку оно указывает на то, что узлы могут выводить новые значения, но это не обязательно так.

Таким образом, это подтверждает состояние постоянного вытеснения, когда консенсус всегда возможен, но сеть постоянно препятствует процессу, задерживая или переупорядочивая критические сообщения нежелательным образом. В чисто асинхронных, детерминированных системах постоянное вытеснение неизбежно при наличии сбоев в работе узлов. К счастью, преимущество носит временный характер. Это не означает сбоя узла, так как система может восстановиться в любой момент. Протоколы могут смягчить эту проблему, вводя случайность или делая реалистичные предположения о задержках сообщений. Допущения о задержке более практичны, когда кто-то хочет ограничить время выполнения или избежать необходимости в доверенном дилере, что обычно требуется для более эффективных рандомизированных алгоритмов. Конечно, от времени отправки сообщения должно зависеть только завершение, а не безопасность.

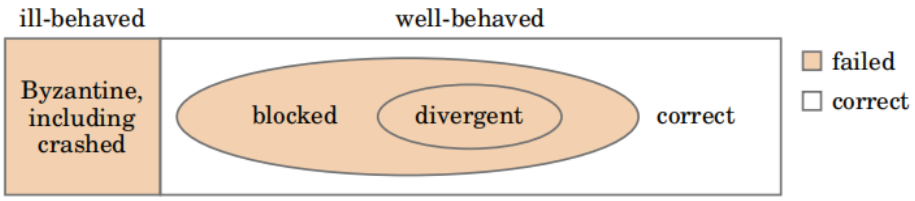

Рисунок 5: Диаграмма Венна неисправностей узлов

1.5.5.4. Оптимальная устойчивость

Безопасность и работоспособность узлов зависят от нескольких факторов: от выбранных ими срезов кворума, от того, какие узлы ведут себя некорректно, а также от конкретного протокола консенсуса и поведения сети. Как это обычно бывает в асинхронных системах, мы предполагаем, что сеть в конечном итоге будет доставлять сообщения между исправными узлами, но в противном случае сообщения могут испытывать произвольные задержки или переупорядочиваться.

В этом разделе рассматривается следующий вопрос: Учитывая конкретные значения V,Q и определенное подмножество некорректно работающих узлов в V, какую максимальную безопасность и жизнеспособность может гарантировать любой протокол Федеративного Византийского соглашения (FBA), независимо от состояния сети? Сначала мы обсудим пересечение кворума, подчеркнув, что отсутствие этого свойства приведет к невозможности гарантировать безопасность. Далее мы вводим концепцию одноразовых наборов, которые представляют собой наборы неисправных узлов, которые, несмотря на их наличие, по-прежнему обеспечивают безопасность и работоспособность.

1.5.5.5. **Пересечение кворума**

Протокол может обеспечить согласованность только в том случае, если срезы кворума, представленные функцией Q, удовлетворяют тому, что мы называем свойством пересечения кворума.

Определение (Пересечение кворумов): FBAS имеет пересечение кворумов тогда и только тогда, когда любые два кворума совместно используют хотя бы один узел — то есть для всех кворумов U_1 и U_2,$U_1 \cap U_2 \neq \emptyset$.

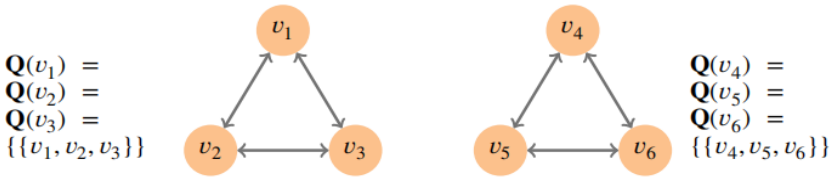

Рис. 6. FBAS, у которых отсутствует пересечение кворумов

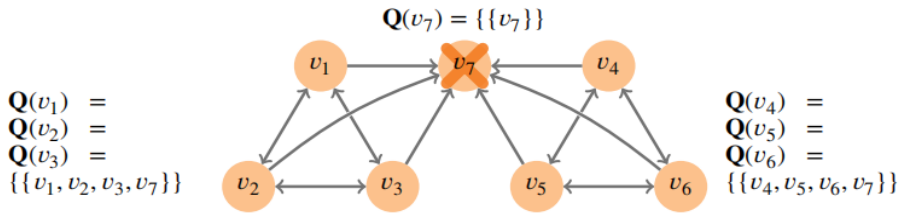

Рис. 7. Некорректно работающий узел v_7 может подорвать пересечение кворумов.

На рисунке 6 показана система, в которой отсутствует пересечение кворумов, где Q допускает два непересекающихся кворума $\{v_1,v_2,v_3\}$ и $\{v_4,v_5,v_6\}$.Эти непересекающиеся кворумы могут независимо согласовывать противоречивые утверждения, тем самым подрывая согласованность всей системы. При наличии нескольких кворумов пересечение кворумов завершается неудачей, если любые два кворума не пересекаются. Например, множество всех узлов $\{v_1,\ldots,v_6\}$ пересекается с двумя другими кворумами, но в системе по-прежнему отсутствует пересечение кворума, поскольку два других кворума не пересекаются друг с другом.

Без пересечения кворумов ни один протокол не может обеспечить безопасность, поскольку эту конфигурацию можно рассматривать как две независимые системы FBAS, которые не обмениваются никакими сообщениями. Однако даже при пересечении кворумов безопасность все равно может быть нарушена в присутствии некорректно работающих узлов. Смотрите рисунок 6 (показаны два непересекающихся кворума) и рисунок 7 (показаны два кворума, пересекающиеся в одном узле v7 , где v_7 ведет себя некорректно). Если v_7 делает противоречивые заявления для левого и правого кворумов, то эффект эквивалентен получению непересекающихся кворумов.

На самом деле, поскольку некорректно работающие узлы никак не влияют на безопасность, ни один протокол не может обеспечить безопасность, если корректно работающие узлы сами не обеспечивают пересечение кворумов. В конце концов, в наихудшем случае для обеспечения безопасности некорректно работающие узлы всегда могут делать любые возможные (противоречивые) заявления, чтобы удовлетворить кворумы. Поскольку некорректно работающие узлы двойственны, два кворума, которые пересекаются только на некорректно работающих узлах, могут снова работать как две отдельные системы FBAS. Короче говоря, FBAS <V,Q> может допускать Византийские ошибки в наборе узлов B⊆V, если V, Q имеет пересечение кворумов после удаления узлов в *B* из V и всех срезов в Q. Более формально:

Определение (Удалить): Если <V,Q> - это FBAS, а B⊆V - набор узлов, то

удаление B из <V,Q>, обозначаемого $<V,Q>^B$, означает вычисление модифицированного FBAS $<V\setminus B,Q^B>$, где $Q^B(v)=\{q\setminus B | q\in Q(v)\}$.

Каждый узел *v* отвечает за то, чтобы Q(*v*) не нарушал пересечение кворумов. Один из способов добиться этого - выбирать консервативные срезы, тем самым формируя более крупные кворумы. Конечно, злоумышленник *v* может намеренно выбрать Q(*v*), чтобы нарушить пересечение кворумов. Но злоумышленник *v* также может ложно сообщить значение Q(*v*) или проигнорировать Q(*v*), чтобы сделать произвольные утверждения. Короче говоря, когда *v* работает некорректно, значение Q(*v*) становится бессмысленным. Вот почему необходимое свойство для обеспечения безопасности — пересечение кворумов между корректно работающими узлами после удаления некорректно работающих узлов - не зависит от срезов некорректно работающих узлов.

Предположим, что рисунок 6 эволюционирует от трехузловой FBAS $\{v_1,v_2,v_3\}$ с пересечением кворума к шестиузловой FBAS без пересечения кворумов. Когда v_4, v_5, v_6 объединяются, они злонамеренно выбирают срезы, которые нарушают пересечение кворумов, и ни один протокол не может гарантировать безопасность V. К счастью, удаление некорректно работающих узлов для получения $<V,Q>^{\{v4,v5,v6\}}$ восстанавливает пересечение кворумов, что означает, что по крайней мере $\{v_1,v_2,v_3\}$ могут наслаждаться безопасностью. Обратите внимание, что удаление носит концептуальный характер и предназначено для описания оптимальной безопасности. Протокол должен обеспечивать безопасность v_1, v_2, v_3, не требуя, чтобы они знали о некорректной работе v_4, v_5, v_6.

1.5.5.5. Необязательные наборы

Мы обеспечиваем отказоустойчивость выбора срезов узлов с помощью понятия необязательных наборов, или DSet. Неофициально, DSet обеспечивает безопасность и работоспособность узлов за пределами DSet, независимо от поведения узлов внутри DSet. Другими словами, в оптимально устойчивой FBAS, если один набор данных содержит все некорректно работающие узлы, он также должен содержать все неисправные узлы, и наоборот, все узлы за пределами DSet являются исправными.

Например, в централизованной системе PBFT с $3f+1$ узлами и размером кворума $2f+1$ любой набор из f или меньшего количества узлов образует DSet. Поскольку PBFT действительно допускает f Византийские ошибки, ее устойчивость является оптимальной.

В менее распространенных случаях $\{v_1\}$ является DSet, поскольку сбой в работе одного узла верхнего слоя не влияет на остальную систему. $\{v_9\}$

также является DSet, поскольку ни один другой узел не зависит от правильности v_9. $\{v_6,\ldots,v_{10}\}$ является DSet, поскольку ни v_5, ни узлы верхнего слоя не зависят ни от одного из этих пяти узлов. Однако $\{v_5, v_6\}$ не является DSet, поскольку это срез для v_9 и v_{10}, и, если он полностью вредоносный, он может ввести в заблуждение v_9 и v_{10}, заставив их делать утверждения, несовместимые друг с другом или с остальной частью системы.

Чтобы некорректная работа DSet не повлияла на корректность работы других узлов, должны быть выполнены два свойства. В целях безопасности удаление DSet не должно нарушать пересечение кворумов. Для обеспечения жизнеспособности DSet не должен препятствовать правильному функционированию других узлов. Это приводит к следующему определению:

Определение (DSet): Пусть $<V,Q>$ - это FBAS, а $B \subseteq V$ - набор узлов. Мы называем B необязательным набором или DSet тогда и только в случае:

1. (Пересечение кворумов, несмотря на B):$<V,Q>^B$ имеет пересечение кворума, и
2. (Наличие кворума, несмотря на B): V\B - это кворум в <V,Q> или B=V.

Наличие кворума, несмотря на B, не позволяет узлам в B отказываться отвечать на запросы и блокировать работу других узлов. Пересечение кворумов, несмотря на B, предотвращает противоположный сценарий — узлы в B делают противоречивые утверждения, позволяя другим узлам выводить несогласованные значения для того же слота. Узлы должны сбалансировать эти две угрозы при выборе среза. При прочих равных условиях более крупные срезы приводят к более сильным кворумам и большему перекрытию, что означает, что меньшее количество наборов неисправных узлов B нарушит пересечение кворумов при удалении. С другой стороны, более крупные срезы с большей вероятностью будут содержать неисправные узлы, что ставит под угрозу доступность кворума.

Наименьший DSet, содержащий все некорректно работающие узлы, может также включать в себя корректно работающие узлы, что отражает тот факт, что достаточно большой набор некорректно работающих узлов может привести к сбою корректно работающих узлов. Например, наименьший DSet, содержащий v_5 и v_6, равен $\{v_5,v_6,v_9,v_{10}\}$. Множество всех узлов V всегда является DSet, так как FBAS$<V,Q>$ имеет нулевое пересечение кворумов, несмотря на V, и, в частном случае, доступность кворума, несмотря на V. Особый случай обусловлен тем фактом, что при наличии достаточного количества некорректно работающих узлов V может быть наименьшим DSet, содержащим все некорректно работающие узлы, что указывает на то, что ни один протокол не может гарантировать лучшего результата, чем полный отказ системы в таком сценарии.

DSet в FBAS априори определяются функцией кворума *Q*. Корректная или некорректная работа узлов зависит от работы во время выполнения, например, от взлома компьютера. Интересующие нас DSet содержат все некорректно работающие узлы, поскольку они помогают нам отличать узлы, корректность которых должна быть гарантирована, от тех, у которых ее гарантировать нельзя. С этой целью мы вводим следующие термины:

Определение (Неповрежденный): Узел v в FBAS является неповрежденным тогда и только тогда, когда существует DSet B, содержащий все некорректно работающие узлы, и $v \notin B$.

Определение (Зараженный): Узел v в FBAS считается загрязненным тогда и только тогда, когда он не является неповрежденным.

Зараженный узел v окружен достаточным количеством неисправных узлов, которые, даже если сам узел v работает корректно, препятствуют его продвижению или нарушают его состояние. Ни одна FBAS не может гарантировать корректность работы зараженных узлов. Однако лучшая система FBAS может гарантировать, что каждый неповрежденный узел остается исправным. Следующая теорема облегчает анализ, показывая, что набор зараженных узлов всегда является DSet в FBAS с пересечением кворумов.

well-behaved / ill-behaved	Local property of nodes, independent of other nodes (except for the validity of slice selection).
intact / befouled	Property of nodes given their quorum slices and a particular set of ill-behaved nodes. Befouled nodes are ill-behaved or depend, possibly indirectly, on too many ill-behaved nodes.
correct / failed	Property of nodes given their quorum slices, a concrete protocol, and actual network behavior. The goal of a consensus protocol is to guarantee correctness for all intact nodes.

Рисунок 8: Основные свойства узлов FBAS

Теорема 1: Пусть U - кворум в FBAS <V,Q>, пусть B⊆V - набор узлов, и пусть U'=U\B. Если U'=∅, то U' - это кворум в <V,Q>$^{B.}$

Доказательство: Поскольку U является кворумом, для каждого узла $v \in U$ существует $q \in Q(v)$, такое, что $q \subseteq U$. Учитывая, что $U' \subseteq U$, для каждого $v \in U'$ также существует $q \in Q(v)$, такое, что $q \setminus B \subseteq U'$. Переписав это, используя обозначение удаления, для каждого $q \subseteq U'$ существует $\exists q \in Q_B(v)$, такое, что $q \subseteq U'$. Поскольку $U' \subseteq V \setminus B$, это показывает, что U’ - кворум в <V,Q>$^{B.}$

Теорема 2: Если B_1 и B_2 являются DSet в FBAS <V,Q> с пересечением кворумов, то $B=B_1 \cap B_2$ также является DSet.

Доказательство: Пусть $U_1=V\backslash B_1$ и $U_2=V\backslash B_2$. Если $U_1=\emptyset$, то $B_1=V$ и $B=B_2$ (DSet), завершая доказательство. Аналогично, если $U_2=\emptyset$, тогда $B=B_1$, что также завершает доказательство. В противном случае обратите внимание, что при наличии кворума, несмотря на существование наборов B_1 и B_2, U_1 и U_2 остаются кворумами в <V,Q>. По определению, объединение двух кворумов также является кворумом. Следовательно, $V\backslash B=U_1 \cup U_2$ является кворумом, и наличие кворума сохраняется, несмотря на B.

Теперь мы докажем пересечение кворумов, несмотря на B. Пусть U_a и U_b - любые два кворума в $<V,Q>^{B.}$ Пусть $U=U_1 \cap U_2=U_2\backslash B_1$. По пересечению кворума в <V,Q>, $U=U_1 \cap U_2 \neq \emptyset$. Однако, согласно Теореме 1, $U=U_2\backslash B_1$, поэтому U должен быть кворумом в $<V,Q>_{B1}$.

Теперь рассмотрим, что $U_a\backslash B_1$ и $U_a\backslash B_2$ не могут быть пустыми одновременно; в противном случае $U_a\backslash B=U_a$ было бы пустым. Следовательно, либо $U_a\backslash B_1$ является кворумом в $<V,Q>_{B1}$, либо $U_a\backslash B_2$ является кворумом в $<V,Q>_{B2}$, либо и то, и другое. В первом случае обратите внимание, что если $U_a\backslash B_1$ является кворумом в $<V,Q>_{B1}$, то при пересечении кворума в $<V,Q>_{B1}$, $(U_a\backslash B_1) \cap U \neq \emptyset$. Поскольку $(U_a\backslash B_1) \cap U=(U_a\backslash B_1)\backslash B2$, из этого следует, что $U_a\backslash B_2 \neq \emptyset$, что делает $U_a\backslash B_2$ кворумом в $<V,Q>_{B2}$. Согласно аналогичному аргументу, $U_b\backslash B_2$ также должен быть кворумом в $<V,Q>_{B2}$. Однако, несмотря на существование B_2, пересечение кворумов гарантирует, что $(U_a\backslash B_2) \cap (U_b\backslash B_2) \neq \emptyset$, что возможно только в том случае, если $U_a \cap U_b \neq \emptyset$.

Теорема 3. В FBAS с пересечением кворумов набор вскрытых узлов является DSet.

Доказательство: Пусть B_{min} - это пересечение всех DSet, содержащих все некорректно работающие узлы. Согласно определению целостности, узел v является неповрежденным тогда и только тогда, когда $v \notin B_{min}$. Следовательно, B_{min} — это как раз набор вскрытых узлов. Поскольку DSet закрыты при пересечении, B_{min} также является DSet.

1.6. Основные преимущества SwiftBlock Coin

1.6.1. Преимущества SCP

SwiftBlock Coin обеспечивает прозрачный, безопасный, удобный для пользователя и эффективный метод транзакций с помощью протокола консенсуса SwiftBlock, предлагая очень гибкую децентрализованную экосистему.

- **Прозрачность**: Все записи о транзакциях поддаются публичной проверке, что гарантирует пользователям доступ к информации реестра в любое время.
- **Эффективность**: Оптимизируя время подтверждения блокировки и сокращая задержку в сети, SCP обеспечивает более быструю обработку транзакций.
- **Безопасность**: Механизм совместной работы распределенных узлов снижает риск возникновения отдельных точек сбоя и вредоносных атак.
- **Удобство в использовании**: Более низкие транзакционные издержки позволяют частным лицам и предприятиям участвовать в блокчейн-экосистеме с меньшими затратами.
- **Децентрализованная экосистема**: Позволяет пользователям свободно участвовать в управлении, обеспечивая справедливость и устойчивость экосистемы.

1.6.2. Преимущества механизма SCP

Протокол консенсуса SwiftBlock (SCP) — это усовершенствованный механизм консенсуса, который сочетает в себе Византийскую отказоустойчивость (PBFT) и Доказательство доли (PoS), известное как Доказательство согласия (PoA), с целью повышения безопасности и эффективности транзакций.

- **Высокая отказоустойчивость**: Благодаря интеграции BFT сеть остается стабильной, даже если некоторые узлы выходят из строя или действуют злонамеренно.
- **Низкое энергопотребление**: По сравнению с Доказательством работы (PoW), SCP использует проверку, ускоренную с помощью доли, что устраняет необходимость в значительных вычислительных ресурсах и снижает энергопотребление сети.
- **Высокая пропускная способность**: Оптимизированный алгоритм консенсуса сокращает количество ненужных шагов по подтверждению транзакций, значительно увеличивая TPS (количество транзакций в секунду).
- **Децентрализация**: Устраняет контроль со стороны традиционных централизованных институтов, обеспечивая справедливость и автономию сети.

1.6.3. Преимущества экосистемы SwiftBlock Coin

Токенизация трафика позволяет всему трафику стать доступным для торговли, а SwiftBlock Coin служит основным средством для транзакций на свободном рынке.

- **Экосистема SwiftFi**: SwiftBlock Coin поддерживает внедрение смарт-контрактов, позволяя пользователям участвовать в децентрализованном кредитовании, майнинге ликвидности, торговле стабильной криптовалютой и других финансовых операциях.
- **Маркетплейс NFT**: Используя механизм смарт-контрактов SwiftBlock Coin, пользователи могут создавать, торговать и выставлять на аукцион активы NFT, облегчая децентрализованные транзакции с цифровым искусством, внутриигровыми предметами и другими цифровыми активами.
- **Межсетевое взаимодействие**: SwiftBlock Coin использует технологию межсетевого моста для обмена активами с основными публичными сетями (например, Ethereum, BSC, Polkadot), повышая ликвидность.

1.6.4. Варианты использования

SwiftBlock Coin — это революционная криптовалюта, ориентированная на токенизацию трафика, предлагающая высокую гибкость и широкий спектр приложений.

- **Токенизация трафика**: Пользователи могут легко приобретать трафик с помощью SwiftBlock Coin, устанавливая его в качестве основной цифровой валюты для оплаты трафика.
- **Традиционная торговля бизнес-трафиком**: Компании могут покупать трафик с помощью SwiftBlock Coin, увеличивая объемы транзакций, не полагаясь на сторонних посредников (например, рекламные агентства, платежные каналы), снижая затраты и обеспечивая глобальный трафик и платежи без границ.
- **Финансовая деятельность**: Благодаря экосистеме SwiftFi SwiftBlock Coin обеспечивает децентрализованную финансовую деятельность, такую как кредитование, инвестиции в смарт-контракты и размещение ставок, снижая барьеры и повышая эффективность.
- **Маркетплейс NFT**: Пользователи могут создавать, торговать и выставлять на аукцион активы NFT, используя механизм смарт-контрактов SwiftBlock Coin.
- **Децентрализованный обмен трафиком (DAE)**: основанный на блокчейне рынок обмена рекламой без посредников, использующий SwiftBlock Coin в качестве основного цифрового актива, повышающий прозрачность, безопасность и ликвидность при одновременном снижении затрат на рекламу.

1.6.5. Безопасность активов

Протокол консенсуса SwiftBlock (SCP) объединяет множество алгоритмов и механизмов для достижения более высокой безопасности.

- **Децентрализованный консенсус**: Технология блокчейн обеспечивает безопасность цифровых активов благодаря децентрализованному консенсусу и быстрому завершению.
- **Гибкая модель доверия**: Гибкий механизм доверия SCP позволяет узлам выбирать надежных партнеров и валидаторов в соответствии со своими потребностями, повышая общую безопасность сети.
- **Отсутствие зависимости от суперузлов**: Даже при наличии нескольких вредоносных узлов поддерживается безопасность консенсуса, обеспечивающая работу сети, несмотря на DDoS-атаки, взломы или сбои узлов.
- **Низкая задержка**: Транзакции подтверждаются в течение нескольких секунд, не полагаясь на майнеров или вычислительные мощности, что позволяет избежать рисков для безопасности, связанных с задержками.
- **Отсутствие уязвимостей в смарт-контрактах**: SwiftBlock Coin не поддерживает полные по Тьюрингу смарт-контракты, что исключает такие распространенные атаки, такие как повторный вход или целочисленное переполнение.

1.6.6. Механизм вознаграждения SFB

SwiftBlock Coin создает децентрализованную, прозрачную и справедливую экосистему поощрений, вознаграждая пользователей за привлечение рекламы, создание контента и торговлю трафиком.

- **Пользователи**: Зарабатывают SwiftBlock Coin, просматривая рекламу, взаимодействуя и делясь информацией.
- **Рекламодатели**: Получают вознаграждение за анализ данных за ставки и размещение рекламы.
- **Поставщики трафика**: Зарабатывают SwiftBlock Coin за точные рекомендации.
- **Поставщики данных**: Извлекают выгоду из авторизованного обмена данными.
- **Награды сообщества**: Ускоренные выплаты за поддержку и рост сообщества.

1.6.7. Распределение монет SwiftBlock Coin

Категория	Сумма (монеты)	Процент	Цель
Фаза внутренней сети	300 000 000	10%	Глобальное распределение, сокращение вдвое вознаграждения для стимулирования первых пользователей и роста.
Фаза основной сети	1 950 000 000	65%	Глобальное распределение для стимулирования роста и развития экосистемы.
Ликвидность и рыночные операции	300 000 000	10%	Поддержка рыночного оборота.
Исследовательская команда	150 000 000	5%	Финансирование технических исследований.
Команда разработчиков	150 000 000	5%	Покрытие расходов на разработку.
Программы мотивации	150 000 000	5%	Поощрение деятельности и развития сообщества.
Общий объем	**3 000 000 000**	**100%**	

1.6.7.1. Общий объем

SwiftBlock Coin имеет фиксированный общий запас в 3 000 000 000 монет, что обеспечивает дефицит и долгосрочную ценность. Распределение является прозрачным и справедливым, что способствует устойчивому росту. Механизм сжигания уменьшает общее количество токенов в обороте, сохраняя стабильность стоимости.

Фаза внутренней сети

В течение фазы внутренней сети SwiftBlock Coin использует глобально распределенную модель выпуска, чтобы обеспечить честное распределение токенов. Целью этого этапа является стимулирование участия пользователей с помощью механизмов раннего стимулирования, способствующих первоначальному росту проекта и развитию экосистемы. Постепенно снижая вознаграждение, проект стремится поддержать энтузиазм первых пользователей, одновременно предотвращая чрезмерную инфляцию. Эти токены могут быть использованы для вознаграждения активных пользователей, поддержки управления сообществом и обеспечения необходимой поддержки экосистемы. Кроме того, часть средств будет направлена на маркетинговые меры, направленные на повышение видимости SwiftBlock Coin, привлечение большего числа пользователей к пониманию проекта и участию в нем, тем самым закладывая основу для последующего развития основной сети. Чтобы способствовать здоровому обращению токенов, SwiftBlock Coin

внедрила механизм сжигания, при котором часть токенов сжигается во время регистрации каждого узла или транзакции в цепочке, тем самым сокращая общее количество токенов в обращении и поддерживая дефицит токена и стабильность его стоимости в долгосрочной перспективе.

Алгоритм консенсуса SwiftBlock Coin основан на надежных узлах, что делает его крайне важным для стимулирования первых участников к созданию кругов личной безопасности. Это означает, что до запуска основной сети сеть Swift Block будет предлагать существенные вознаграждения пользователям высокого уровня, которые помогают поддерживать сообщество или вносят свой вклад в его развитие. Поддержание долгосрочных стимулов сети также имеет важное значение, поэтому сеть использует уникальный механизм, при котором скорость выпуска уменьшается вдвое всякий раз, когда количество монет, получаемых пользователем от выпуска узлов, удваивает его первоначальные инвестиции, и этот процесс повторяется до тех пор, пока не будет достигнут предел.

Фаза основной сети

Фаза основной сети является критическим периодом для развития экосистемы SwiftBlock Coin. Основной целью токенов на этом этапе является содействие долгосрочному росту всей сети, расширение разнообразия сценариев децентрализованных приложений и привлечение большего числа пользователей и разработчиков к участию с помощью механизмов стимулирования. SwiftBlock Coin обеспечивает справедливое распределение благодаря глобальной модели выпуска, оптимизируя стратегии стимулирования для укрепления долгосрочных намерений пользователей. По мере стабилизации основной сети SwiftBlock Coin будет широко использоваться в различных приложениях для децентрализованного финансирования (SwiftFi), платежных системах, платформах смарт-контрактов и многом другом. Кроме того, эти токены будут поддерживать экосистему разработчиков, вознаграждая участников и финансируя инновационные проекты, что еще больше улучшит экосистему Swift Block. SwiftBlock Coin также реализует механизм "сжигания транзакций", при котором часть токенов сжигается при каждом переводе по цепочке, чтобы уменьшить предложение в обращении и обеспечить устойчивый рост рыночной стоимости.

Ликвидность и рыночные операции

Ликвидность имеет решающее значение для стабильности и устойчивого роста SwiftBlock Coin на рынке. Таким образом, часть токенов выделяется для повышения ликвидности рынка, поддержки работы торговой платформы и проведения необходимых мероприятий по

продвижению на рынок. Хорошая ликвидность помогает снизить волатильность рынка, позволяя инвесторам и пользователям торговать более плавно, одновременно повышая конкурентоспособность SwiftBlock Coin на мировом рынке. Что касается рыночных операций, то эти средства будут использованы для создания бренда, продвижения в сообществе и установления партнерских отношений. Например, сотрудничая с основными биржами, SwiftBlock Coin стремится увеличить глубину и доступность торговли, что делает ее более распространенной. Кроме того, для дальнейшей стабилизации рынка механизм сжигания SwiftBlock Coin будет играть важную роль в управлении ликвидностью, когда часть токенов постоянно сжигается во время рыночных транзакций, чтобы оптимизировать спрос и предложение, повышая долгосрочную ценность токена.

Исследовательская команда

Непрерывное развитие экосистемы SwiftBlock Coin опирается на мощную техническую исследовательскую поддержку. Таким образом, часть токенов выделяется для финансирования долгосрочных усилий исследовательской команды по внедрению инноваций в основные технологии блокчейна. Основными задачами исследовательской группы являются оптимизация блокчейн-протоколов, изучение безопасности смарт-контрактов и разработка новых механизмов консенсуса. Благодаря постоянным техническим исследованиям SwiftBlock Coin стремится повысить безопасность сети, масштабируемость и эффективность, чтобы соответствовать требованиям будущих децентрализованных приложений. Кроме того, эти средства пойдут на поддержку академического сотрудничества с университетами и исследовательскими институтами для изучения развития технологии блокчейн и продвижения отраслевых стандартов. Чтобы оптимизировать экономическую модель SwiftBlock Coin, исследовательская группа усовершенствует механизм сжигания, чтобы достичь наилучшего баланса между стабильностью сети и стимулами, обеспечивая устойчивое развитие экосистемы.

Техническая команда

Техническая команда отвечает за разработку, обслуживание и оптимизацию сети SwiftBlock Coin. Часть токенов направляется на выплату зарплат разработчикам, строительство инфраструктуры и техническую модернизацию. Являясь новатором в области децентрализованных финансов (SwiftFi) и блокчейн-технологий, SwiftBlock Coin должна поддерживать высокие технические стандарты для обеспечения стабильности и безопасности сети. Основные задачи технической команды включают оптимизацию сетевой архитектуры, управление узлами, обновление смарт-контрактов и улучшение пользовательского опыта.

Кроме того, эти средства поддержат развитие технического сообщества, привлекая больше разработчиков присоединиться к экосистеме SwiftBlock Coin и способствуя росту проекта. Примечательно, что SwiftBlock Coin использует механизм динамического сжигания для регистрации узлов, при котором определенное количество токенов сжигается при каждом присоединении нового узла, обеспечивая справедливое распределение ресурсов и увеличивая долгосрочный дефицит токена.

Механизмы стимулирования

Часть токенов выделяется на продвижение на рынке, общественные мероприятия и программы мотивации для повышения вовлеченности пользователей и влияния бренда. Организуя глобальные маркетинговые кампании, такие как онлайн- и оффлайн-промо-акции, маркетинг в социальных сетях и программы поощрения сообщества, SwiftBlock Coin стремится привлечь больше пользователей и компаний к внедрению своей сети. Кроме того, эти средства будут использованы для проведения хакатонов, конкурсов разработчиков и других мероприятий, призванных побудить разработчиков создавать инновационные приложения на SwiftBlock Coin. Проект также внедрит механизм "burn-to-earn" в мотивационных мероприятиях, при котором часть маркетинговых расходов будет использоваться для покупки и сжигания токенов во время определенных кампаний, контролируя предложение на рынке и повышая потенциал роста стоимости токена.

Механизм сжигания

Чтобы обеспечить долгосрочную стабильность экосистемы SwiftBlock Coin и дефицит токенов, SwiftBlock Coin реализует механизм автоматического сжигания, при котором определенное количество токенов сжигается во время каждой транзакции в цепочке или регистрации узла. Основной целью этого механизма является контроль предложения на рынке, снижение инфляции и повышение рыночной стоимости SwiftBlock Coin, что делает его по-настоящему устойчивой криптовалютой.

Механизм сжигания при транзакции

При каждом переводе SwiftBlock Coin по цепочке сжигается часть токенов. Это означает, что всякий раз, когда пользователи совершают транзакции, оплачивают, обменивают или переводят токены в сети, система автоматически сжигает часть токенов, постоянно сокращая их оборот. Этот механизм не только стабилизирует цену токена, но и стимулирует пользователей хранить токены в долгосрочной перспективе, снижая давление продаж на рынке и обеспечивая устойчивый рост стоимости.

Механизм сжигания при регистрации узлов

Чтобы обеспечить справедливость и децентрализацию в сети SwiftBlock Coin, при регистрации каждого нового узла также запускается механизм сжигания. Новые узлы должны заплатить определенное количество монет SwiftBlock в качестве регистрационного взноса, часть которого сжигается. Этот механизм предотвращает злоупотребление сетевыми ресурсами, гарантируя, что только пользователи, искренне приверженные сетевой безопасности и стабильности, могут участвовать в работе узла, при этом еще больше сокращая оборотные средства и повышая долгосрочную ценность токена.

Долгосрочный дефляционный эффект

Механизм сжигания SwiftBlock Coin — это непрерывный, динамичный процесс. По мере роста пользователей сети и объемов транзакций количество сожженных токенов будет постепенно увеличиваться, создавая долгосрочный дефляционный эффект. Этот механизм гарантирует, что SwiftBlock Coin со временем избежит проблем с инфляцией, позволяя ее стоимости неуклонно расти. Кроме того, SwiftBlock Coin может внедрять стратегии периодического обратного выкупа и сжигания для дальнейшей оптимизации спроса и предложения, поддерживая рыночную стоимость токена.

Механизм сжигания является важнейшим компонентом экономической модели SwiftBlock Coin. Благодаря двойным механизмам "сжигания транзакции" и "сжигании узлов" предложение токенов постепенно сокращается, что повышает их долгосрочную ценность и стабильность рынка. Этот механизм не только укрепляет привлекательность SwiftBlock Coin как децентрализованного финансового актива, но и обеспечивает прочную экономическую основу для устойчивого развития экосистемы. По мере расширения сети SwiftBlock Coin механизм сжигания будет играть все более важную роль в поддержании дефицита токенов и стабильности, что сделает ее по-настоящему эффективной, безопасной и децентрализованной блокчейн-экосистемой.

Общий объем

SwiftBlock Coin имеет фиксированный общий запас в 3 000 000 000 монет, что обеспечивает дефицит и долгосрочную ценность. Все распределения токенов осуществляются в соответствии с прозрачными и справедливыми принципами для поддержки устойчивого развития проекта. Благодаря строгой экономической модели и научным механизмам выпуска токенов SwiftBlock Coin позволяет избежать инфляции и волатильности цен, обеспечивая стабильный рост рыночной стоимости. Механизм сжигания

играет жизненно важную роль в экосистеме SwiftBlock Coin, включая сжигание токенов во время транзакций по цепочке, регистрации узлов и рыночных операций. Этот механизм обеспечивает долгосрочную стабильность SwiftBlock Coin, сокращает предложение в обращении, поддерживает баланс спроса и предложения и повышает рыночную стоимость токена. SwiftBlock Coin стремится к созданию безопасной, эффективной и децентрализованной блокчейн-экосистемы, что делает ее надежным цифровым активом для пользователей по всему миру. В будущем SwiftBlock Coin продолжит оптимизировать свою экономическую модель и механизмы управления, чтобы обеспечить устойчивый рост стоимости и способствовать широкому внедрению децентрализованных финансовых услуг по всему миру.

1.6.8. Будущее развитие

SwiftBlock Coin нацелен на повышение сетевой безопасности, оптимизацию взаимодействия с пользователями, расширение вариантов использования и повышение стабильности рынка. Это позволит модернизировать блокчейн-инфраструктуру, поддерживать сложные DApps и способствовать развитию глобальных партнерств. Механизм сжигания будет продолжать создавать дефляционное давление, обеспечивая долгосрочный рост стоимости.

1.6.9. Вывод

SwiftBlock Coin — это децентрализованная криптовалюта, построенная на технологии блокчейн, предлагающая эффективную, безопасную и прозрачную цифровую финансовую экосистему. Разработанный SCP SwiftBlock протокол консенсуса обеспечивает высокую эффективность транзакций и безопасность сети. Благодаря глобальному распространению и инновационным стимулам SwiftBlock Coin способствует росту экосистемы. Обладая надежной экономической моделью и механизмом сжигания, SwiftBlock Coin стремится стать ведущим мировым цифровым активом, предоставляющим безопасные, эффективные и прозрачные финансовые услуги в по-настоящему открытой и децентрализованной блокчейн-экосистеме.