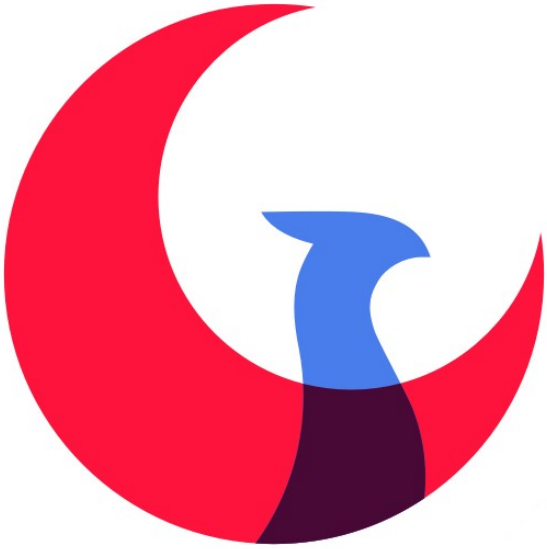


TKEYCOIN DAO
ПРОЕКТНО-ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ

ВЕРСИЯ 1.2 ДАТА ПУБЛИКАЦИИ 26 СЕНТЯБРЯ 2018 ГОДА

TKEY DMCC
support@tkeycoin.com
tkeycoin.com

Эта проектно-техническая документация Tkeycoin V. 1.2 предназначена только для информационных целей. TKEY DMCC не гарантирует точность сведений и выводов, сделанных в этом техническом документе, и этот технический документ предоставляется «как есть». TKEY DMCC не делает никаких заявлений и гарантий, а также прямо отказывается от всех подразумеваемых, гарантий и выражений, установленных законом или иным образом, включая, но не ограничиваясь: (i) гарантии товарности, пригодности для определенной цели, использования названий или ненарушения каких-либо прав; (ii) что содержание этой белой книги не содержит ошибок; и (iii) что такое содержание не будет нарушать права третьих лиц. TKEY DMCC и его аффилированные лица не несут ответственности за любые убытки, возникшие в результате использования этого технического документа, предоставления ссылки на него третьим лицам, а также использования любого содержащегося в нем содержания, даже если поступит сообщение о возможности такого ущерба.

Проектно-техническая документация подготовлена на основе анализа различных концепций и проведенных исследований в области криптографии, p2p-сетей, блокчейн, протоколов наличности, искусственного интеллекта, машинного обучения, экономики и криптовалют.

Данная документация создана компанией TKEY DMCC для будущих пользователей сети Tkeycoin DAO, а также для разработчиков, it-компаний, сферы e-commerce и любых представителей гражданского общества, которые планируют перейти на новый уровень взаимовыгодных бизнес-процессов на основе одноранговых платежных систем.

На основе нововведений и доработок в проекте, которые указаны в долгосрочном плане развития, проектно-техническая документация будет постоянно обновляться и расширяться.

ВНИМАНИЕ: Эксклюзивные наработки с техническим обоснованием станут раскрыты после перехода сети в рабочий режим. Компания TKEY DMCC опубликует открытый код на интернет-ресурсе GitHub — https://github.com/tkeycoin, после чего, проектно-техническая документация будет преобразована в полностью научно-технический документ.

TKEY DMCC продолжит свое участие, как основной поставщик кода. При этом лицензирование коснётся отдельных элементов системы, распространяющихся на коммерческой основе, а также SAAS-решений для крупных компаний.

Open Source

В общем и целом, мы придерживаемся концепции "Open Source". Доступность исходного кода сделает решения управляемыми для пользователей, поскольку это гарантирует возможность поддержки и развития решений сторонними разработчиками и интеграторами. Работоспособность и корректность функционирования Open Source продуктов проверяется миллионами энтузиастов и профессионалов со всего мира.

Открытый код и свободная лицензия могут использоваться в любых областях социальной и деловой жизни. Концепция Tkeycoin DAO нацелена не только на программистов, но и на все гражданское общество. И, прежде всего, на ту его часть, для которой свобода выбора и принятия решений являются важными факторами повседневной жизни.

Цифровая культура, выстраиваемая в Tkeycoin DAO, будет способствовать выражению личности и коллективного творчества. Наши принципы не

позволят сделать из такого важного аспекта обыденное средство выживания через наемный труд.

Содержание

Прерывание функции Curve25519 (например, вычисление совместно используемого секретного ключа из двух открытых ключей) считается крайне сложным. Каждая известная атака является более дорогостоящей, чем перебор на типичном 128-битным шифром с секретным ключом шифрования и расшифрования. Над решением общей проблемы дискретных логарифмов эллиптической кривой трудились в течение двух десятилетий, но успехи были незначительными. Обобщенные алгоритмы дискретных логарифмов разбивают простые группы, которые не достаточно большие, но размер простой группы,

Введение

С момента появления Bitcoin прошло десять лет, технологии берет начало из далекого 1983 года, когда были предложены первые протоколы электронной наличности.

Однако даже спустя такое количество лет, существующие криптовалюты не обладают качествами реальных денег, необходимыми для привлечения массового потребителя. В связи с этим доля криптовалютного рынка ничтожно мала в сравнении с обычными валютами.

Проблема напрямую связана с отсутствием широкого рынка, пользователи криптовалюты вынуждены использовать ее только в качестве средства быстрого заработка.

Несмотря на фундаментальные проблемы рынка, развитие экономики на базе блокчейн приводит к появлению множества новых идей и новых применений ранее известных концепций.

Блокчейн предлагает пересмотр понятий, которые годами принимались, как должное, и не подвергались критическому анализу. Сюда относятся: денежные расчеты, валюта, имущество, правительство, суверенитет, интеллектуальная собственность. Подвергая сомнению базовые определения и находя новые смыслы в привычных терминах, блокчейн-технология способна совершить настоящую революцию сразу в нескольких различных сферах.

Глава 1.

1.1 Фиатные деньги

Фиатная валюта является законным платежным средством. При этом она не поддерживается каким-либо физическим товаром, таким как золото, стоимость которого определяется предложением, спросом и стабильностью правительства, выпустившего валюту.

1.1.1. Потеря популярности фиатных валют

Продолжая тенденцию практичности, характерную для XXI века, бумажные деньги постепенно исчезают из нашей жизни, уступая место более практичным цифровым хранилищам. Тем не менее, оцифрованный банкинг, который мы сейчас ежедневно используем, все еще далек от совершенства. Для начала, он полностью контролируется третьими лицами. Никто на самом деле не владеет цифрами, которые они видят на экране – контроль полностью принадлежит третьим сторонам, таким, как банки.

Деньги, которые у вас есть на банковском счету, можно считать виртуальной валютой, так как у она не имеет физической формы и существует только в банковской книге. Если они потеряют книгу, Ваши деньги просто исчезнут.

1.1.2. Деньги из воздуха

Банки создают деньги из воздуха, и ярким примером этого является кредит. Деньги больше не печатаются, когда кто-то берет овердрафт или ипотеку, – они просто создаются из ничего. Более того, эти банки взимают непропорционально высокую плату за услуги, которые они предоставляют, при этом услуги эти устарели и непрактичны сегодня. Например, непрактично платить комиссию, чтобы потратить свои деньги за границей, как и непрактично ждать несколько дней для проверки перевода небольшой суммы от вас вашему родственнику. Все это не имеет никакого смысла во взаимосвязанном и мгновенном мире, в котором мы с вами сегодня живем.

Таким образом, денежно-кредитная система перестала быть практичной, она заменяется более высокой формой хранения ценностей. В этом конкретном случае, ей на смену приходит более быстрая и безопасная система, которая исключает дорогостоящие операции и дает контроль человеку.

1.1.3. Инфляция в фиате

При долгосрочном хранении сбережений в национальной валюте, она постепенно теряют свою ценность, снижается покупательная способность сбережений. Инфляция традиционных денег происходит в основном из-за увеличение денежной массы в стране сверх потребностей товарного обращения. По сути, на сегодняшний день, рубли, доллары, евро - ничем не подкреплены, так как не обеспечены даже тем же золотом.

Создать определённое количество денег не составляет труда, ведь не нужно добывать золото, чтобы напечатать потом эквивалентное количество бумажных денег. В связи с этим, денежная масса легко увеличивается, если того требует правительство (причины могут быть разные — рост государственных расходов, массовое кредитование необеспеченной валютой).

Определить необходимое количество денежной массы в определённый момент трудно, поэтому в традиционной экономике почти повсеместно существует инфляция.

Криптовалюты изначально лишены даже возможности появления инфляции, т.к. их количество, как и количество золота на Земле, ограничено алгоритмом.

1.2. Анализ рынка

В процессе формирования концепции децентрализованной платформы Tkeycoin DAO нами были проанализированы общечеловеческие проблемы и их воздействие на экономику, исследованы системные и циклические глобально-экономические процессы. Особое внимание было уделено недавнему всемирному экономическому кризису.

Тема экономики будущего крайне актуальна, однако большинство идей о новом экономическом устройстве не выходят за пределы уже известных экономических парадигм.

Сегодня глобальные экономические проблемы влияют на все сферы жизни, затрагивают, в той или иной степени, все без исключения государства. Последствия кризисных явлений, как в экономике, так и в других сферах жизнедеятельности человечества (демография, проблема голода, экологический, энергетический и сырьевой кризисы, политическая нестабильность и пр.) – могут быть поистине катастрофическими.

1.2.1 Ограничения и государственное регулирование

Усиленное регулирование со стороны государственных и банковских структур провоцирует возникновение теневой экономики, препятствующей нормальному развитию гражданского общества и социально-экономических отношений между людьми.

Нынешние централизованные системы контролируются отдельными структурами, которые могут оказывать негативное влияние на экономику в целом.

Более того, корпорации, пользующиеся своим монопольным положением на рынке, просто замораживают процесс получения новых знаний. С учётом этих обстоятельств, полная передача прикладной фазы инновационного процесса в коммерческую сферу становится недопустимой.

Как свидетельствует мировая практика, реализация инновационной политики с опорой только на научные организации и бизнес-сферы, финансируемые крупными корпорациями, в принципе невозможна, поскольку главные интересы корпораций и их цели функционирования, во многих случаях, не совпадают с интересами и целями общества.

Принимая во внимание вышеперечисленные факты, необходимо создание децентрализованных экосистем, способных обеспечить решение приоритетных национальных задач развития и инноваций.

К сожалению, в большинстве стран мира такая экосистема практически отсутствует. Это не позволяет полностью раскрыть ее функциональное назначение, как системы обеспечения реализации социально-экономических задач в области научно-инновационного развития и гаранта благополучия в гражданском обществе.

1.2.2 Криптовалютный рынок

Взгляды различных слоев общества на финансовую политику до сих пор существенно рознятся, в силу чего образуются разногласия между разработчиками и обществом. Результат – дробление технологии на множество проектов и изменения протокола. Такая политика негативно влияет на адаптацию криптовалют в обществе.

Подавляющая часть проектов не несет реальных социально-экономических функций и не вносит свой вклад в развитие комфортной жизни для гражданского социума. Хотя новые технологии, действительно, имеют огромный потенциал для общества и массового потребителя, но ни одна из них этот потенциал пока не реализовала.

Главная проблема заключается в отсутствии долгосрочной стратегии и полного понимания всех возможностей новых цифровых технологий.

На практике сегодня чаще всего создаётся масса узкоспециальных токенов, цель которых - быстрое обогащение. Это дискредитирует по-настоящему интересные проекты. Кроме того, значительная часть разработчиков использует старые технологии, не задумываясь о создании чего-то нового.

Все это – следствие неравновесного состояния как экономики, так и всех остальных сфер человеческой деятельности. Глубина и серьезность проблем неуклонно нарастала в течение Нового и Новейшего времени.

Выявлено, что, если рассматривать глобальные экономические и общечеловеческие проблемы, как неравновесные состояния динамических систем, можно увидеть не только трудности и тупиковые ситуации, но и векторы оптимального выхода из данных ситуаций и даже возможности перевода системы мировой экономики на новый эволюционный уровень.

1.2.3. Новые технологии, как путь к успешному развитию

Именно новые технологии способствуют успешному развитию социально-экономической деятельности в обществе. Создание высокотехнологичных разработок требует повышения общего уровня образования, создания технической элиты в обществе, совершенствования инфраструктуры и смежных областей. Это неизбежно ведет к повышению общего уровня развития населения, появлению у него новых запросов, развитию внутреннего рынка и повышению благосостояния.

Эффективный аппарат самоуправления в децентрализованной сети, построенный на принципах сотрудничества, взаимной помощи, самоотдачи, справедливости и солидарности, создает справедливую экономику для всего сообщества.

Справедливое распределение финансовых потоков сохраняет пропорцию между средствами, полученными пользователями на потребление ресурсов, и средствами, полученными фондом на развитие каждого производственного цикла. Поэтому спрос и предложение в рамках замкнутой экономики любого масштаба всегда сбалансированы.

Преимущество данного подхода в том, что он не затрагивает ничьих прав собственности и может быть реализован на обычном законодательном уровне.

Учитывая уровень нынешней глобальной экономики, переход на новую «справедливую экономику» является наиболее безболезненным для всего

общества. Создание надежной экономической системы, уравновешивающей долгосрочные интересы инвесторов, разработчиков, бизнеса и всех пользователей, является ключевой задачей Tkeycoin DAO.

1.3. Экосистема Tkeycoin DAO

В октябре 2017 года официально зародилась идея создания экосистемы Tkeycoin DAO. Проект был запущен с целью перейти на качественно новый уровень экономики за счет создания и использования одноранговой системы платежей на базе блокчейн технологии.

1.3.1. Мотивы

Мы выступаем за глобальное масштабирование рынка криптовалют и их массовое распространение в экономической сфере. Нами движет большая идея и новые технологии. Мы хотим привлечь внимание широкой аудитории и масштабировать проект на международном уровне., присоединив к проекту мировых исследователей, разработчиков, миллионы пользователей и тысячи компаний.

Сама технология может повлиять не только на все то, что связано с денежными рынками, платежными системами, финансовыми услугами и экономикой, но и на все остальные индустрии. Разрабатываемая нами технология обеспечивает универсальность и глобальный масштаб, ранее немыслимый неосвоенный. Это своего рода новая парадигма, позволяющая организовывать деятельность с меньшими усилиями, при этом более эффективно и масштабно, чем другие существующие парадигмы.

Цифровые технологии и блокчейн способны изменить мир в лучшую сторону путем создания безопасных и справедливых систем самоуправления. Если развивать данные системы правильно, то в течение пяти лет гражданское общество привыкнет к цифровым валютам и новым технологиям. Они станут реалиями повседневной жизни.

1.3.2. О концепции

Современные платёжные системы несовершенны и зависят от воли высокопоставленных чиновников. Децентрализованные системы, основанные на технологиях P2P, являются более справедливым средством взаимных расчётов пользователей.

Мы подошли к рынку с экономической и научной точки зрения, заимствуя лучшее из Bitcoin , Ethereum, DASH и других альтернативных валют, смешивая современные концепции и основываясь на мировом опыте IBM, Microsoft и целового ряда других компаний и научных исследований.

Bitcoin стал первой успешной реализацией распределенной криптовалюты, частично описанной в 1998 году Вэй Даем в списке рассылки cypherpunks.

Концепция Bitcoin легла в основу всех присутствующих на рынке альткоинов и токенов. Если в концепцию цифровой наличности внести изменения, то, на практике, мы получим цифровую валюту, которая отвечает нуждам и потребностям своих пользователей, которая полностью заменит бумажные деньги.

1.3.3. Цели проекта:

Основная цель – сделать криптовалюту широкодоступным платёжным средством, вложив в нее все свойства обычной валюты. При этом пользователи смогут производить платежи онлайн и оффлайн в любых сферах экономической деятельности, без ограничений со стороны третьих лиц и финансовых институтов.

Цели включают в себя (но не ограничиваются):

- Создание платежной системы, основанной не на доверии, а на криптографии, которая позволила бы любым двум участникам осуществить перевод средств напрямую, без участия посредника.

- Пользователи имеют полный контроль над своими платежами, банковскими карточками, виртуальными счетами, денежными переводами и обменом криптовалюты.

- Создание сбалансированной экосистемы между криптовалютами и обществом.

- Контроль экономических и социальных процессов внутри глобальной сети, делая операции законными и справедливыми.

- Отсутствие превышения полномочий или злоупотребления специальными правами, тем самым исключая мошенничество, коррупцию или рейдерские действия.

- Создание и развитие технологических решений на базе протокола для банковской, финансовой, государственной сфер деятельности.

- Рабочая среда на основе открытого кода для всех пользователей.

- Интеграция с платформой Tkeycoin такая же простая, как и с документированными SDK и плагинами для всех основных платформ и языков.

- Доступные интерфейсы и мобильные платежи.

- Полная децентрализация платформы.

Глава 2. Техническая документация — Tkeycoin Core

TKEYCOIN DAO - это криптографическая децентрализованная платформа, основанная на новом ядре blockchain. На основе нового протокола будет выпущена криптовалюта Tkeycoin (TCD), которая является мировым платежным средством, с помощью которого вы можете оплачивать независимо от местоположения и валюты магазина или организации.

Cryptocurrency Tkeycoin - это новая одноранговая платежная система, основанная на принципах полной децентрализации, где регулирование внутренних процессов третьими сторонами полностью отсутствует

2.1 Децентрализованная система, основанная на технологии P2P:

Децентрализованные системы, основанные на технологиях P2P, являются более справедливым средством взаимных расчётов пользователей.

Одноранговые сети (P2P) – это масштабируемый способ распространения информации среди широкой аудитории. Для одноранговой сети, одним из основных показателей производительности является среднее время, необходимое для доставки определенного файла всем одноранговым узлам которое в целом зависит от топологии сети и планирования передач.

Основное преимущество одноранговых архитектур по сравнению с классическими архитектурами клиент-сервер - их масштабируемость. Поскольку каждый узел является, и клиентом, и сервером одновременно, P2P-сети могут передавать данные большему количеству узлов за гораздо более короткий промежуток времени.

2.2. Общие положения

В данном разделе рассматривается классическая ситуация, в которой файл распределяться как можно быстрее известному набору узлов. Это может быть использовано в качестве базовой модели для многих сценариев, включая платежные системы.

Это также стандартная модель, используемая для иллюстрации масштабируемости P2P сетей, в которых можно рассчитать сумму времени, необходимого для распространения файла определенного размера всем узлам, как в P2P, так и в архитектуре клиент-сервер.

Расчет обычно выполняется с использованием последней временной отметки доставки, которая определяется как временной период, когда последний узел получает полный файл. Другой естественной фундаментальной метрикой является среднее время доставки, что является суммой времени доставки всех узлов, деленной на количество узлов. Однако, его уменьшение приносит более аналитические задачи, и данный раздел посвящен поиску точной процедуры планирования для достижения оптимального среднего время доставки в полностью подключенной P2P сети.

Основная сложность разработки оптимальных алгоритмов распределения файлов заключается в необходимости отслеживания идентичности данных. Другими словами, нужен целый файл, а не только тот объем данных, который может включать в себя много дублирования. Это усложняет проблему того, как узел должен выбрать отправку части данных из “кому больше всего нужен этот объем данных? Кому больше всего нужны именно эти данные?"

Игнорирование этого ограничения значительно снижает сложность задачи, но приводит к нереалистичным результатам. В целом, то, как общая сеть извлекает выгоду из решения отправить определенную часть данных к

определенному узлу, зависит от критерия оптимальности, а также физических ограничений вовлеченных узлов.

Раздел посвящен проблеме разработки явных алгоритмов планирования распространения файлов, которые минимизируют среднее время доставки.

Чтобы преодолеть вышеупомянутую трудность, наша общая стратегия заключается в использовании промежуточного шага путем введения другой концепции (*min-min time*), которая имеет присущую индуктивную структуру, которая облегчает разработку алгоритма.

Раздел организован следующим образом:

Раздел 2.3 рассматривает решение, которое достигает оптимального последнего времени доставки и затем формулирует проблемы min-min и среднего времени доставки. После этого мы представляем два основных результата.

Первый из них находится в разделе 2.4, где дается полное решение для достижения оптимального минимального времени, а также пояснение метода распределения нагрузки.

Второй - в разделе 2.5, где мы утверждаем, что достижение min-min потенциально может свести к минимуму среднее время доставки.

Мы заканчиваем в разделе 2.6 и рассматривает подтверждение.

2.3. Формулировка проблемы

2.3.1 Модель и обозначения

Рассмотрим один узел, называемый сервером, который должен распространять файл размера | F | на N одноранговых узлов. Предполагается, что система свободна , поскольку одноранговые узлы не приходят и не уходят. Мы предполагаем, что нет никаких топологических ограничений; каждый узел, включая сервер, может взаимодействовать друг с другом без каких-либо препятсвий, кроме ограничений загрузки узлов. Наконец, файл может быть разбит на бесконечно малые части; таким образом, нет никакой задержки пересылки, и узел может немедленно передать то, что он получает к другому узлу.

Мы используем следующие обозначения:

- $|F|$: размер файла
- $F_i(t)$: часть файла, который узел i имеет в момент времени t
- $|F_i(t)|$: размер части
- N : общее количество одноранговых узлов (не включая сервер)
- C_0: мощность загрузки сервера
- C_i: мощность загрузки узла i, $C_1 \geq C_2 \geq \ldots \geq C_N$.
- $C = C_0 + \Sigma^n_{i=1} C_i$: общая мощность системы
- $R_{ij}(t,t+\tau)$: данные, отправленные из узла i в узел j в интервале $(t,t+\tau)$.
- $r_{ij}(t) = \frac{d}{dt} |R_{ij}(0,t)|$: скорость, с которой узел i отправляет на узел j в интервале t
- Время окончания t_i для узла i : наименьшее t с $|F_i(t)| = |F|$

- $|F| / C_0$ – время затора: время, необходимое одному узлу для прямого получения всего файла с сервера, и нижняя граница времени для получения файла всеми узлами. Мы рассматриваем сценарий с ограничением загрузки, в котором каждый узел может получать информацию с неограниченной скоростью передачи данных, но суммарная скорость любых загрузок с каждого узла не должна превышать заданную мощность загрузки этого узла. С математической точки зрения,

$$\sum_{j=1}^{N} r_{ij}(t) \leqslant C_i \; \forall_i, t.$$

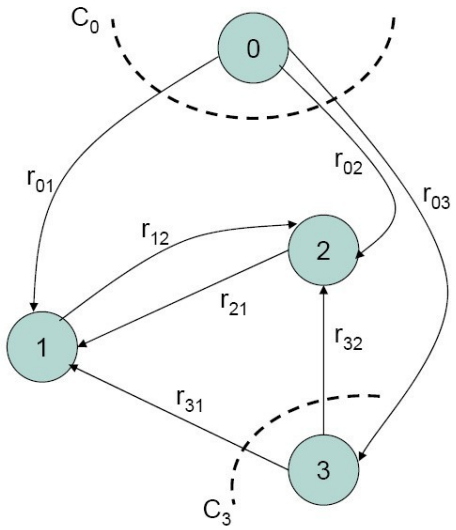


Рисунок 1. Схема, показывающая ограничения на связь между узлами в конфигурации 3 узла плюс сервер. Пунктирные линии представляют ограничения суммарной скорости $\sum_{j=0}^{n} r_{ij}(t) \leqslant C_i \forall_i$

Трудность с «целостностью данных» можно выразить, как

- $R_{ij}(t,t+\tau) \subseteq F_i(t+\tau)$ (ограничение полученных данных; может отправлять только уже полученные данные)
- $R_{ij}(t,t+\tau) \cap F_j(t) = \varnothing$ (только получать новые данные)
- $R_{ij}(t,t+\tau) \cap R_{kj}(t,t+\tau) = \varnothing \; \forall_i \neq k$ *(только получать неповторяющиеся данные)*
- $r_{ii}(t) = 0$ (узел не может отправлять данные сам себе)
- $F_j(t) = \mathrm{U}^{N}_{i=0} R_{ij}(0,t)$, *откуда*
- $\frac{d}{dt} |F_j(t)| = \Sigma^{N}_{l=0} r_{ij}(t) \quad \forall_j, t.$

2.3.2 Средние время доставки

Сначала мы кратко рассмотрим проблему минимизации последнего времени доставки (время, когда все узлы в сети получают весь файл). Очевидно, что на этот раз T^{*}_{L} не может быть меньше $|F| / C_0$, это время, необходимое серверу для отправки файла одному получателю или меньше времени, необходимого

для совместного использования файла со всеми узлами если каждый узел сети был полностью использован на все время, $N \mid F \mid / C$. Формально,

$$T_L^* \geq \max(\mid F \mid /C_0, N \mid F \mid /C) \quad (1)$$

эта нижняя граница является жесткой, рассматривая следующие два варинта.

1) Ситуация 1 – Быстрый сервер: Когда $C_0 \geqslant \sum_{i=1}^{N} C_i/(N-1)$, каждому одноранговому узлу назначается пропускная способность сервера C_i / (N - 1), и каждый одноранговый узел может затем повторно загружать оставшиеся N – 1 узлы со скоростью C_i / (N — 1). Избыточная емкость распределяется поровну. Это приводит к тому, что каждый одноранговый узел получает общую мощность C/N на временном интервале $(0, T_L^*)$.

2) Ситуация 2 – Медленный сервер: Когда $C_0 \leqslant \sum_{i=1}^{N} C_i/(N-1)$, сервер может выделять каждому одноранговому узлу i скорость загрузки

$$\frac{C_i C_0}{\sum_{j=1}^{N} C_j}$$

который не превышает загрузочную способность этого узла. Каждый узел может пересылать то, что он получает, каждому другому узлу; таким образом, каждый узел эффективно получает со скоростью C_0 с сервера.

Оказывается, что принуждение всех узлов к завершению приема файла в T_L^* может искусственно ограничивать производительность сети другими значениями. Другими словами, допустив небольшое увеличение $T_L > T_L^*$, мы можем потенциально существенно уменьшить среднее время доставки T_A и таким образом улучшить общую производительность сети. Это иллюстрируется следующим простым численным примером.

Пример 1: *Потенциальное улучшение по сравнению с минимальным временем доставки.*

Пусть N = 4, а C_0 = 12, C_1 = 6, C_2 = 4, C_3 = 2, C_4 = 1, и $\mid F \mid$ = 144. Мы вычисляем оптимальное время доставки T_L^* и оптимальное среднее время доставки, T_A. (Будет ясно, как рассчитать это в последующих разделах.)

Результаты показаны на рисунке 2. Если допустить небольшой сдвиг в сторону времени доставки t_4, могут быть достигнуты существенные улучшения в другом времени доставки. Например, при выбранном наборе загрузочных мощностей и заданном размере файла среднее сокращение времени доставки на 28,9% соответствует увеличению на 0,91% последнего времени доставки. Теперь понятно, что среднее время доставки является важным показателем производительности. Формально, мы имеем

$$T_A = \frac{\sum_{i=1}^{N} t_i}{N} \quad (2)$$

В общем, чтобы свести к минимуму среднее время доставки, мы хотим максимизировать скорость обмена информацией в сети и попытаться минимизировать время доставки узлов с высокой пропускной способностью

как можно быстрее. Однако из-за комбинаторной структуры проблемы и особенно из-за ограничения обнаруженияя данных трудно даже записать задачу оптимизации для общего примера. Следующий пример иллюстрирует эту трудность в очень простой двухсерверной сети.

Пример 2: *Прямая минимизация среднего времени доставки для двух одноранговых сетей.*

Рассмотрим двухсерверный случай, мы можем настроить линейную программу, которая оптимизирует среднее время доставки, регулируя размеры блоков данных, которые узлы отправляют друг другу в каждом временном интервале в рамках ограничений задачи.

$$
\begin{array}{ll}
\boldsymbol{min} & \\
R_{01},R_{02},R_{12} & t_1+t_2 \\
\textit{при условии} & t_1=|R_{01}(0,t_1)|/(\lambda C_0) \\
& t_2=t_1+(|R_{01}(0,t_1)|-|R_{12}(0,t_1)|) \\
& \dfrac{|R_{01}(0,t_1)\cap R_{02}(0,t_1)|}{C_1+C_0} \\
& \lambda=(|R_{01}(0,t_1)|-|R_{01}(0,t_1)|+|R_{02}(0,t_1)|) \\
& |R_{21}(0,t_1)|/C_2\leqslant t_1 \\
& |R_{21}(0,t_1)|=|R_{02}(0,t_1)\setminus R_{01}(0,t_1)| \\
& |R_{12}(0,t_1)|/C_1\leqslant t_1 \\
& |R_{01}(0,t_1)\cup R_{02}(0,t_1)|=|F| \\
& |R_{01}(0,t_1)|+|R_{02}(0,t_1)|=C_0 t_1 \\
& |R_{12}(0,t_1)|\leqslant |R_{01}(0,t_1)|
\end{array}
$$

Ограничение идентификации данных заставляет нас отслеживать размеры многих различных фрагментов данных, даже если N = 2 (последние шесть ограничений в приведенной выше оптимизации).

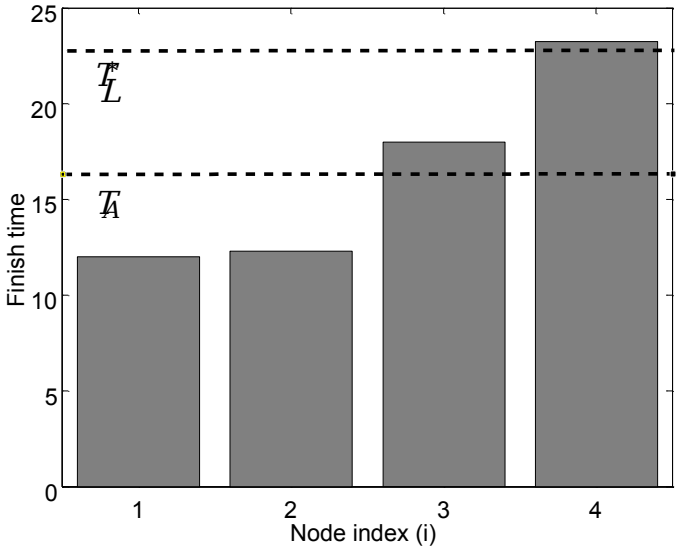


Рисунок 2.
Результаты для случая. N = 4, с C_0 = 12, C_1 = 6, C_2 = 4, C_3 =3, C_4 = 1, и | F | = 144. T_A - *среднее время доставки.* T^*_L - *является оптимальным последним временем доставки.*

В целом аналогичные оптимизации могут быть записаны для большего N, но число переменных и ограничений растет экспоненциально с размером проблемы. Эта трудность мотивирует нас искать индуктивные структуры, которые позволяют нам не оптимизировать все части данных одновременно. Это должна быть min-min times, которая будет представлена в разделе 2.2.3.

2.3.3. Min-Min Times

Min-min time последовательно минимизирует время индивидуальной обработки. Помимо его отношения к оптимальному среднему времени доставки, он также представляет самостоятельный интерес, так как минимизация времени доставки ранних потоков повышает устойчивость к отключению сети.

Формально, пусть t^s_i - время доставки узла i по схеме тарифов s.

- Пусть S_1 - множество схем, которые минимизируют время t_1.
- Пусть S_{i+1} - множество схем, которые минимизируют время i + 1-го доставки, при условии, что все предыдущие грани доставки минимизированы.

Говорят, что схема в $s \in S_N$ достигает *min-min times*, и время t^s_i называют min-min times.

Индуктивная структура, навязанная последовательной минимизацией, позволяет найти явный план для достижения min-min. Это будет показано в разделе 2.4. Прежде чем углубиться в наши основные результаты, введем полезное понятие множественности (*multiplicity*), которое будет использоваться для классификации проблем. Определим множественность, M, как максимальное количество узлов, которые могут получить файлы с размером | F | с препятсвием | F | / C_0.

Лемма 1. *Пусть M - наибольшее значение K такое, что существует график с*

$$F_i(\frac{|F|}{C_0}) = F, \forall_i \leqslant K.$$

Тогда M - наибольшее целое число такое, что

$$C_0 \leqslant \sum_{i=1}^{M} \frac{C_i}{M-1} + \sum_{M+1}^{N} \frac{C_i}{M} \quad . (3)$$

2.4. Планирование достижения минимального времени (Min-Min Times)

Когда кратность $M = N$, все узлы могут закончить | F | / C_0, используя расписание, рассмотренное в разделе 1.2.2 теперь мы изучаем оптимальные расписания для остальных случаев ($M < N$).

Основная трудность в достижении минимального времени, когда мы пытаемся минимизировать t_i, как использовать дополнительные возможности некоторых узлов. Будет показано, что им нужно только минимизировать t_i+1. Другими словами, планирование более чем на один шаг вперед не рационально. Еще одна трудность заключается в том, как запланировать все узлы, чтобы минимизировать t_i, учитывая, что все они имеют разные мощности C_i. Для определения оптимального расписания для всех узлов будет использоваться метод распределения нагрузки. Потенциальные вклады законченных узлов к следующему узлу можно рассматривать как "заполнение водой", и данные запланированные быть поделенным другими узлами формируют неровный пол.

Точнее, см. Рис. 3 (a) для иллюстрации. Во время интервала (t_{i-1}, t_i) j-й столбец имеет ширину C_j и площадь $F_j(t_1) \setminus F_i(t_{i-1})$. Таким образом, глубина является минимальным временем, которое потребовалось бы для узла j для загрузки всех данных, которые он мог бы на узел i.

Заметим, что множества $F_j(t_1) \setminus F_i(t_{i-1})$ не пересекаются при $j > i$. (Это будет гарантировано нашим алгоритмом планирования.) Таким образом, область в столбцах $j > i$ - это точно данные, которые должны быть переданы узлу i в интервале (t_{i-1}, t_i), и вопрос в том, кто должен передавать то, что для минимизации этого интервала. Если сервер и завершенные узлы не отправили какие-либо дополнительные данные в узел i, максимальная глубина - это минимально возможное значение t_i-t_{i-1} (столбец N на рисунке 3). Оптимальным способом является то, что узлы $0 \leq j < i$ отправляют затененные данные на рисунке 3 (b), выравнивая время доставки $t_i - t_{i-1} = |F_j(t_1) \setminus F_i(t_{i-1})| \, C_j$.

Единственный оставшийся вопрос - это то, что должен делать узел, когда другие загружают данные на него. Из-за ограничения «достоверности данных» $r_{ii}(t) = 0$ и, следовательно, он не может передавать данные самому себе. Оптимальным способом является использование пропускной способности узла i для передачи данных в $i + 1$. Точные данные для отправки будут определяться «пустотой». для следующего временного интервала (t_{i-1}, t_i), в том, что данные U_{ij} были бы в столбце j, если бы он не был отправлен на узел $i + 1$ в интервале (t_{i-1}, t_i), но вместо этого он открывается сверху столбцов, пропорциональный их возможностям (рис.3 (b)). В последующих доказательствах мы предоставим специализированные данные по распределению нагрузки для разных случаев (рис. 4 и 6). Фактический алгоритм планирования изложен в алгоритме 1. Он использует верхнюю границу диапазона C_0 для конкретной заданной кратности M, для которой ровно один набор оптимальных значений $F_i(|F|/C_0)$, $\forall_i > M + 1$, способен достичь первого $M + 1$ минимального времени (min-min times).

$$\frac{M\left(C_0^* - \frac{C_{m+1}}{M} - \sum_{i=1}^{M} \frac{C_i}{M-1}\right) + C_0^*}{C_0^* + \sum_{i=1}^{M} C_i} \quad (4)$$

$$= \frac{(M+1)\left(C_0^* - \frac{C_{M+1}}{M} - \sum_{i=1}^{M} \frac{C_i}{M-1}\right)}{\sum_{i=M+2}^{N} C_i}$$

Где $C_0 > C_0^*$, могут быть несколько наборов $F_i(|F|/C_0)$, $\forall_i > M + 1$, которые достигают первого $M + 1$ минимального времени.

Затем в алгоритме 1 также используется следующая линейная программа для выбора единственного набора значений $F_i(|F|/C_0)$, который позволяет достичь всех минимальных времен.

$$max \sum_{i=M+2}^{N} (n-i)\lambda_i (5)$$

s.t.

$$\frac{C_i}{M+1} < \lambda_l \leqslant \frac{C_i}{M} \forall_i \geqslant M+2$$

$$\sum_{i=M+2}^{N} \lambda_i = C_0 - \frac{C_{M+1}}{M} - \sum_{i=1}^{M} \frac{C_i}{M-1}$$

$$\frac{(M+1)\lambda_i - C_i}{C_i} \geqslant$$

$$\frac{\frac{1}{M-1}\sum_{i=1}^{M} C_i + (M+1)\sum_{i=M+2}^{N} \lambda_i - \sum_{i=M+2}^{N} C_i}{C - C_{M+1}}$$

Следующая теорема характеризует алгоритм 1

Алгоритм 1 Оптимальное планирование для достижения минимального времени (min-min times)

- На $(0,t_1)$, пусть $r_{0i} = \lambda_i$, $r_{i1} = \min(\lambda_i, c_i)$, $r_{i2} = c_i - \min(\lambda_i, c_i)$, где λ_i удовлетворяет
- $$\sum_{i=1}^{N} \lambda_i = C_0 \qquad \lambda_2 = C_2 \qquad \frac{2\lambda_i}{C_i} = \frac{\lambda_i + C_0}{C_0 + C_1}, \forall_i > 2 \quad (6)$$
- На (t_{i-1},t_i), $2 \le i < N$: $r_{ji}(t) = Cj \quad \forall j \neq i$, с $R_{ji}(t_{i-1},t_i) \cap R_{ki}(t_{i-1},t_i) = \emptyset$ и $(R_{ji}(t_{i-1},t_i) \cup R_{ki}(t_{i-1},t_i)) \cap F_i(t_{i-1}) \neq \emptyset$ для всех $k = j$. Узел i передает данные U_{ij} узлу $i+1$ с $r_{i,i+1}(t) = C_i$ такие, что

1. $U_{ij} \cap U_{ik} = \emptyset$ для всех $k \neq j$ (данные не пересекаются)
2. $U_{ij} \in F_j(t_i - 1)$ (данные хранятся на t_{i-1} узлом j)
3. Для всех $j \geq 0$, $j \neq i+1$,

$$\frac{|U_{ij}|}{t_i - t_{i-1}} = \gamma_{ij} = \frac{C_j}{\left(\sum_{k=0, k\neq i+1}^{N} C_k\right)} C_i$$

Кроме

- Если $M = N - 1$ or $C_0 \le C^*_0$, (4), то пусть λ_i решает

$$\sum_{i=1}^{N} \lambda_i = C_0$$

$$\underbrace{\begin{matrix} \lambda_1 / C_1 & if\ i \leqslant M \\ 1/M & if\ i = M+1 \\ \lambda_{M+2} / C_{M+2} & if\ i \geqslant M+2 \end{matrix}}_{\frac{\lambda_i}{C_i}}$$

$$\frac{\lambda_{M+2}}{C_{M+2}} = \frac{M + \sum_{i=1}^{M} \lambda_i + C_0}{(M+1)\sum_{i=0}^{M} C_i}$$

Иначе пусть $\lambda_i = C_i / (M-1)$, $\forall_i \le M$, и λ_i *для* $i \geq M+2$ удовлетворяют LP (5)

В конце если

- На $(0,t_M)$:

$r_{0i} = \lambda_i, \forall i$; $r_{ij}(t) = \lambda_i$ для $j \le M, j \ne i$; $r_{ij}(t)=0$

для $j > M+1$; $r_{i,M+1}(t) = C_i - \Sigma_{j \ne M+1} r_{ij}(t)$.

- На (t_i-1, t_i) для $M+1 \le i < N$:

$r_{ji}(t) = C_j$ для $j \ne i$, такой, что

$|R_{0i}(t_i-1, t_i) \cap F_j(t_i-1)| = \mu_{ji}(t_i - t_{i-1})$,

$$\mu_j, M+1 = \frac{C_0}{(\sum_{k=1}^{N} C_K) - C_{M+1}} C_j, \qquad \forall_j \ne M+1,$$

$$\mu_{j,i} = \lambda_i t_i - C_i \frac{(C_0 - \lambda_i) t_i}{C_0 + C - C_i} \qquad \forall_j \ne i \ne M+1.$$

для $j < i$, где

$r_{i,i+1}(t) = C_i$, такой что $|R_{i,i+1}(t_i-1, t_i) \cap F_j(t_i-1)| = \gamma_{ji}$ где

$$\gamma_{ji} = \frac{C_i C_j}{(\sum_{k=1}^{N} C_k) - C_{i+1}}, \qquad \forall_j \ne i+1$$

конец, если

На (t_N-1, t_N), $r_{iN}(t) = C_i$ *для* $i < N$, и $r_{ik}(t) = 0 \quad \forall k$.

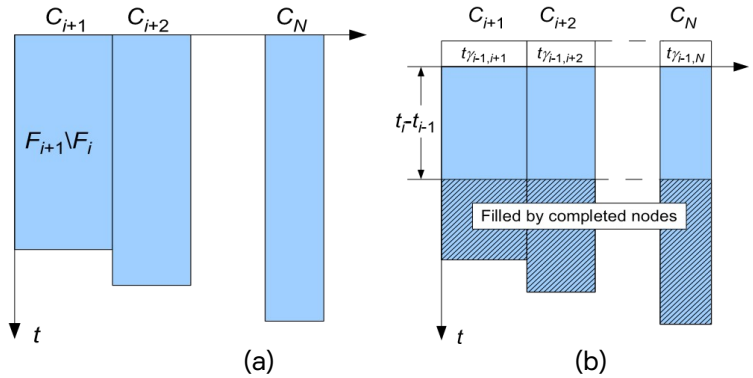


Рисунок 3. Распределение нагрузки. Ширина столбца j равна емкости C_j, а глубина - время передачи $F_j(t_1) \setminus F_i(t_{i-1})$ со скоростью C_j. В (a) узел N занимает больше времени для передачи своей информации. В (b) сервер «*заполнен водой*», сокращая время для всех, чтобы завершить передачу на узел i, и разрешить полное использование для этого интервала. Заполнение гелия $(t_i - t_{i-1})\ \gamma_{ij}$ в интервале (t_{i-1}, t_i) равномерно уменьшает высоты всех столбцов.

Теорема 1. *Алгоритм 1 достигает минимального времени.*

Доказательство. *Утвердим здесь доказательство для M = 1; доказательство для 1 <M <N можно найти в Приложении A.*

Заметим сначала, что алгоритм 1 выполним. В частности, до момента времени t_i все узлы $j > i$ имеют непересекающиеся данные, а узлы $j < i$ - все данные. Аналогично, $U_{i,j}$ может быть переадресован i, поскольку он получен от узла j, поскольку $\gamma_{i,j} \le C_j$, позволяя выполнить три заявленных условия.

Остается установить оптимальность. Пусть t_i обозначает минимальное время доставки узла i. Доказательство оптимальности сначала устанавливает нижние

оценки на $\underline{t}_1$ и $\underline{t}_2$ и показывает, что алгоритм 1 достигает этих времен и что λ_i - это уникальные значения, которые могут достичь этого. Затем он индуктивно показывает, что последующие времена сведены к минимуму.

Пусть $C' = \sum_{i=3}^{N} C_i$. Это можно рассматривать как мощность «виртуального узла», состоящего из узлов 3, ..., *N*. Количество информации, которая может попасть в узлы 1 и 2 на (0, t_2) ограничена сверху как

$$F_1(t_2) + F_2(t_2) \le (C_0 + C_1)\, t_2 + C_{2t1} + \frac{C'}{2}\, t_2 . \quad (7)$$

Первые условия показывают, что сервер и узел 1 могут вносить вклад на всем временном интервале. Второй термин отражает передачу узла 2 узлу 1 на (0, t_1); на (t_1, t_2), он не может вносить вклад, поскольку он не может загружать в себя, а на (t_1, t_2) узел 1 уже получил весь файл. Термин $t_2C'/2$ возникает следующим образом. Узел $i \ge 3$ может отправлять информацию, полученную им до момента времени t_2, на оба узла 1 и 2, но он не может превышать собственную мощность загрузки и не может загружать данные t_1, которые у него отсутствуют до t_1. Таким образом, его загрузка в 1 и 2 ограничена сверху min $\{C_i t_2, F_i(t_1) + F_i(t_2)\}$. Однако данные, полученные узлом i с сервера, происходят за счет данных, которые сервер мог отправить непосредственно на узел 1 или 2, что дает

$$\min\{C_i t_2, F_i(t_1) + F_i(t_2)\} - F_i(t_2). \quad (8)$$

заметим, что

$$min\left\{ C_i t_2, F_i(t_1) + F_i(t_2) \leqslant \frac{C_i t_2 + 2F_i(t_2)}{2} \right\} \quad (9)$$

с равенством, только если

$$2F_i(t_1) = 2F_i(t_2) = C_i t_2. \quad (10)$$

Подставляя (9) в (8) и суммируя по $i \ge 3$, получаем $C'T_2/2$, устанавливая (7).

Нижняя оценка по $\underline{t}_2$ получается из подстановки $F_1(t_2) + F_2(t_2) = 2F$ *в (7) и* подставляя известное значение $\underline{t}_1 = |F| / C_0$, давая

$$t_2 \geqslant \frac{2|F| - C_2|F|/C_0}{C_0 + C_1 + C'/2} \quad . \quad (11)$$

Это достигается с помощью алгоритма 1.

Чтобы убедиться, что выбор λ_i является единственным, достигающим $\underline{t}_2$, заметим, что (10) является необходимым условием для всех $i \ge 3$.

Разделение на $C_i t_1$ и подстановка $\lambda_i = |F_i(t_1)| / t_1$ дает

$$\frac{2\lambda_i}{C_i} = \frac{\underline{t}_2}{\underline{t}_1} \quad . \quad (12)$$

для всех $i \ge 3$. Аналогично, данные, известные только узлу 1 и серверу t_1, из которых есть количество $(\lambda_1 - C_1)\, t_1$, также должны быть доставлены со скоростью $C_1 + C_0$ за время $t_2 - t_1$. Разделение на t_1 и добавление 1 дает

$$\frac{\lambda_1 + C_0}{C_0 + C_1} = \frac{\underline{t}_2}{\underline{t}_1} \quad .(13)$$

Сочетание (12) и (13) показывает, что λ_i, $i > 2$, должно удовлетворять (6) для достижения $\underline{t}_2$. Таким образом, алгоритм 1 достигает $\underline{t}_1$ и $\underline{t}_2$, и (6) необходимы для любой схемы, которая делает.

Учитывая, что (6) должно выполняться для достижения $\underline{t}_1$ и $\underline{t}_2$, индукцией по i может быть показано, что: (a) узел i не получает данных в интервале (t_1, t_{i-2}) и (b) t_i равен плотно ограниченный ниже

$$\underline{t}_i \geqslant \frac{|F| - \lambda_i \underline{t}_1 - C_{i-1}(\underline{t}_{i-1} - \underline{t}_{i-2})}{C - C_i} + \underline{t}_{i-1} \quad . (14)$$

Термин $\lambda_i \underline{t}_1$ представляет собой количество данных, полученных узлом i от сервера в течение первого интервала (0, $\underline{t}_1$), а термин C_{i-1} ($\underline{t}_{i-1} - \underline{t}_{i-2}$) - это данные, полученные от узла i–1 в

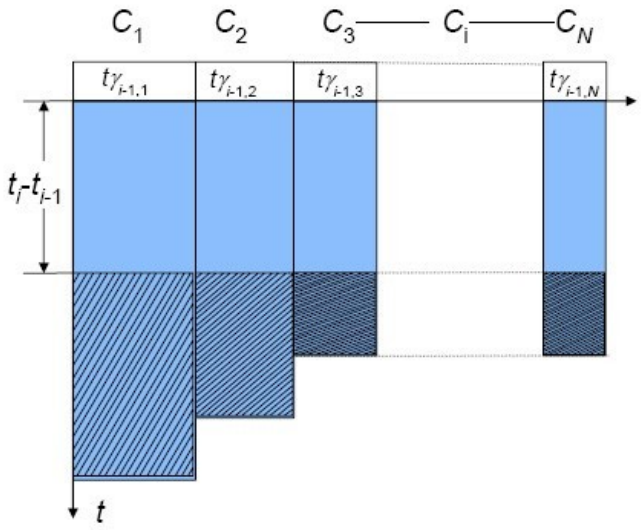


Рисунок 4. Визуальное описание алгоритма заполнения воды (waterfilling) для случая, когда $M = 1$. Заметим, что днища столбцов $M + 2$, ..., N являются уровнями.

интервал, который отмечает ($\underline{t}_{i-2} - \underline{t}_{i-1}$). Для минимизации последнего условия требуется, чтобы узел $i + 1$ не получал данных в интервале (t_1,t_{i-1}). Алгоритм 1 удовлетворяет этому и, следовательно, устанавливает индуктивный шаг.

2.5. Достижение минимального времени означает уменьшение среднего минимального времени доставки

Мы теперь утверждаем, что результат минимального времени, достигнутый Алгоритмом 1 в разделе 1.3, также потенциально минимизирует среднее время доставки. Это согласуется с интуицией аппроксимирования «расписания самого короткого задания», но усложняется наличием нескольких серверов.

Утверждение **1.** *Минимальное время минимизирует среднее время доставки.*

Предположим $C_0 \leqslant \overset{*}{C}_0$ (аргумент для случая с $C_0 > C_0^*$ приведен в Приложении B). Пусть множество A - узлы 1, ..., M, M +1. Максимальный объем информации, которая может

А по времени t_{M+1} можно записать как

$$(\sum_{i=0}^{M} C_i)\, t_{M+1} + C_{M+1}\, t_M - \sum_{i=M+2}^{N} F_i(t_{M+1}) \quad (15)$$

$$+ \quad min\left(t_{M+1} \sum_{i=M+2}^{N} C_{i,}(M+1) \sum_{i=M+2}^{N} F_i(t_{M+1})\right)$$

Это выражение максимизируется для любых t_M и t_{M+1}, когда

$$t_{M+1} \sum_{i=M+2}^{N} C_i = (M+1) \sum_{i=M+2}^{N} F_i(t_{M+1})$$

С параметром $\sum_{i=M+2}^{N} F_i(t_{M+1})$

Для достижения t_{M+1} (15) должно быть больше $(M + 1)\,|F|$. Установив неравенство и решение для t_{M+1}, найдем оценку на t_{M+1} из

$$t_{M+1} \geqslant \frac{(M+1)|F| - C_{M+1}\, t_M}{C - C_{M+1} - \sum_{i=,+2}^{N} \frac{C_i}{M+1}} \quad (16)$$

Заметим также, что из теоремы о множественности достижимая нижняя оценка на $\sum_{i=1}^{M} t_i$ является $M|F|C_0$. Это достигается с помощью Алгоритма 1 с особой множественностью M.

Теперь мы утверждаем, что минимизация необходимая для минимизации $\sum_{i=1}^{N} t_i$. Сначала рассмотрим возможность передачи данных узлам 1, ..., $M + 1$ на $(0, t_{M+1})$ вместо узлов $M + 1$, ..., N на том же интервале. Для данных размера$\in$ это приведет к увеличению $\sum_{i=1}^{M+1} t_i$ по крайней мере $\in/(C - C_{M+2})$ и снижение до $\sum_{i=M+2}^{N} t_i$ не более $\in/(C - C_{M+2})$ игнорируя любые каскадные эффекты в будущем. (Полученная структура от минимизации $\sum_{i=1}^{M+1} t_i$ позволяет любому оставшемуся узлу обслуживаться с полной скоростью в любое время ... таким образом, создавая эти корректирующие цифры.)

Затем обратите внимание, что для того, чтобы узлы $M + 2$, ..., N делились

$$t_{M+1} \sum_{i=M+2}^{N} t_i C_i$$

данные с узлами 1,...,$M + 1$ on $(0, t_{M+1})$ пока только удерживая

$$\sum_{i=M+2}^{N} \frac{C_i}{M+1} t_{M+1}$$

данные, каждый узел j должен передавать со скоростью C_j для всего интервала $(0, t_{M+1})$. Таким образом, существует только одно распределение данных среди узлов $M + 2$, ..., N, что позволяет достичь нижней границы на $\sum_{i=1}^{M+1} t_i$.

Наконец, рассмотрим, могут ли изменения в этих распределениях уменьшаться $\sum_{i=M+2}^{N} t_i$ больше, чем они увеличиваются $\sum_{i=1}^{M+1} t_i$. Рассмотрим снова сдвиг

данных между двумя узлами в множестве M + 2, ..., N до времени t_{M+1}. в В лучшем случае это приведет к увеличению $\sum_{i=1}^{M+1} t_i$ по крайней мере

$$\frac{\epsilon}{C_0 + C_{M+2} + \sum_{i=1}^{M} C_i} \quad (17)$$

так как только узлы S, 1, ..., M и один узел в множестве {M + 2, ..., N} будут иметь необходимые данные для завершения узла M + 1. Уменьшение $\sum_{i=M+2}^{N} t_i$ не более

$$\frac{\epsilon}{C_0 + C_{M+2}} - \frac{\epsilon}{C + C_N} \quad (18)$$

где первый термин происходит от добавления информации к узлу, который обслуживается с наименьшей скоростью, а отрицательный термин исходит из удаления информации с узла, который обслуживается с максимальной скоростью.

Так как (17) больше (18), то никакого уменьшения результата $\sum_{i=1}^{N} t_i$ может быть достигнуто сдвигами до времени t_{M+1}. Поскольку в этот момент любой узел j может обслуживаться со скоростью C - C_j, последовательная минимизация является оптимальной в среднем смысле.

Перед заключением мы демонстрируем применение наших результатов, изучая, как гетерогенность одноранговых мощностей влияет на минимальное среднее время доставки. Алгоритм 1 (теперь оказался оптимальным) используется для вычисления минимального среднего времени доставки.

Пример 3: *Неоднородность повышает производительность.*
Используется та же сеть в примере 1. Суммарная емкость одноранговых узлов фиксирована $(\sum_{i=1}^{4} C_i = 50)$ мощность сервера 100, а размер файла - 10000. Согласно (1), последнее время доставки не изменяется, так как суммарная емкость фиксирована. Однако среднее время доставки изменяется, когда меняется неоднородность одноранговых мощностей. На рисунке 5, среднее время доставки рассчитывается против затора пропускной способности одноранговой сети. Интересно отметить, что увеличение гетерогенности (большая дисперсия) в целом уменьшает среднее время окончания (более высокая производительность). Причина в том, что пропускная способность, доступная для отправки определенного фрагмента файла, растет быстрее, когда отправляется узлу большой мощности, а не разбивается и отправляется на несколько узлов с меньшей пропускной способностью. Этот эффект перевешивает уменьшенную скорость, с которой узлы с низкой пропускной способностью могут отправлять полученную информацию.

ПРИЛОЖЕНИЕ А Доказательство теоремы 1 с 1 <M <N

Доказательство. Доказательство начинается с установления условий для соответствующих значений λ. Затем он находит точные значения λ и min-min раз $\underline{t}_1$, ..., $\underline{t}_{M+1}$ и применяет концепцию вода / гелий, чтобы установить все оставшиеся min-min times.

Чтобы достичь минимального $\underline{t}_1 \ldots \underline{t}_M$, каждый узел должен передать все, что он получает от сервера, на $(0, \underline{t}_M)$ до узлов $i \in \{1, \ldots, M\}$. Таким образом, верхняя граница того, что каждый узел может получать с сервера на $(0, t_M)$, является

$$\lambda_i \frac{C_i}{M-1} \qquad \forall_i \leqslant M \quad (19)$$

$$\lambda_i \frac{C_i}{M} \qquad \forall_i > M \quad (20)$$

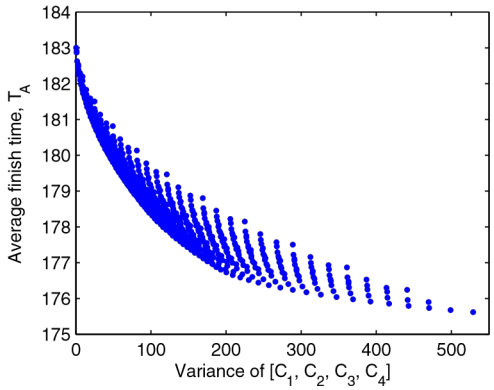


Рисунок 5. Оптимальное среднее время доставки для, $N=4, C_0=100, \sum_{i=1}^{4} C_i=50$ и $|F|$ = 10000, с значениями C_i, ограниченными целыми числами > 1.

2.6. Подтверждение

Поскольку алгоритм 1 сохраняет значения λ_i в этих диапазонах и передает все серверные потоки в узлы $\{1, \ldots, M\}$, времена $\underline{t}_1, \ldots, t_M = | F | / C_0$ минимизируются.

Чтобы установить нижнюю границу на t_{M+1}, сначала рассмотрите, сколько узлов данных $M + 1$ может принимать на $(0, t_M)$, от сервера, узлов $\{1, \ldots, M\}$, и узлов $\{M + 2, \ldots, N\}$:

$$|R_{0,M+1}(0,t_M)| = \lambda_{M+1} t_M$$

$$|\bigcup_{i=1}^{M} R_{i,M+1}(0,t_M)| \leqslant (\sum_{i=1}^{M} C_i - (M-1)\sum_{i=1}^{M} \lambda_i)\, t_M$$

$$|R_{M+1,M+1}(0,t_M)| = 0$$

$$|\bigcup_{i=M+2}^{N} R_{i,M+1}(0,t_M)| \leqslant (\sum_{i=M+2}^{N} C_i - M\sum_{i=M+2}^{N} \lambda_i)\, t_M$$

На (t_M, t_{M+1}) каждый узел $i \in \{0,1, \ldots, M\}$ мог бы отправить $M + 1$ со скоростью r_i, $M + 1\,(t) = C_i$, давая

$$|\bigcup_{i=1}^{M} R_{i,M+1}(t_M, t_{M+1})| \leqslant (t_{M+1} - t_M)\sum_{i=0}^{M} C_i \quad (21)$$

Вклад $U^{n}_{i=M+2} R_{i,M+1}\,(t_M, t_{M+1})$ узлов $\{M + 2, \ldots, N\}$ ограничен, как суммарной загрузочной способностью, $\sum_{i=M+2}^{N} C_i$, и количеством информации, полученной ими на $(0, t_M)$. Таким образом

$$|\bigcup_{i=1}^{M} R_{i,M+1}(t_M, t_{M+1})| \leqslant \min\Big(\sum_{i=M+2}^{N} C_i(t_{M+1}-t_M), \Big[\sum_{i=M+2}^{N}\lambda_i t_i - \sum_{i=M+2}^{N}\lambda_i M\Big] t_m\Big). \quad (22)$$

Они объединяются, чтобы сформировать верхнюю границу объема информации, которая может быть получена узлом M + 1 по времени t_{M+1}, показанного в (23). Также обратите внимание, что по определению, $F_{M+1}(t_{M+1}) = F$.

Рассматривая каждый член min в (23) отдельно, и решение для t_{M+1} дает две нижние оценки на t_{M+1} в терминах $\sum_{i=1}^{M}\lambda_i, \lambda_{M+1}$ и $\sum_{i=M+2}^{N}\lambda_i$

Когда $$\sum_{i=M+2}^{N} C_i t_{M+1} \leqslant \sum_{i=M+1}^{N} (M+1)\lambda_i t_1,$$

$$\underline{t}_{M+1}(C - C_{M+1}) \geq t_M(M-1)\sum_{i=1}^{M}\lambda_i - \underline{t}_M\lambda_{M+1} \quad (24)$$

$$+\underline{t}_M C_0 + \underline{t}_M M\sum_{i=M+2}^{N}\lambda_i + |F|$$

и в обратном случае

$$\underline{t}_{M+1}\sum_{i=0}^{M} C_i \geqslant \underline{t}_M(M-1)\sum_{i=1}^{M}\lambda_i - \underline{t}_M\lambda_{M+1}$$

$$+\underline{t}_M C_0 - \underline{t}_M\sum_{i=M+2}^{N}\lambda_i + |F| \quad (25)$$

Заметим, что в обоих случаях нижняя граница уменьшается в λ_{M+1}, и поэтому минимизируется максимизацией λ_{M+1} путем установки

$$\lambda_{M+1} = \frac{C_{M+1}}{M} \quad (26)$$

$$|F_{M+1(t_{M+1})}| \leqslant \Big(\sum_{i=1}^{M} C_i - (M-1)\sum_{i=1}^{M}\lambda_i\Big)t_M - M\sum_{i=M+2}^{N}\lambda_i t_M + \lambda_M + 1\, t_M \quad (23)$$
$$+(t_{M+1}-t_M)\Big(C_0 + \sum_{i=1}^{M} C_i\Big) + min\Big(\sum_{i=M+2}^{N} C_i t_{M+1}, \sum_{i=M+2}^{N}(M+1)\lambda_i t_i\Big)$$

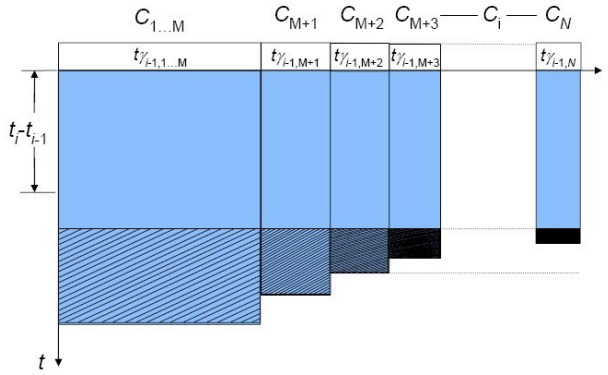


Рисунок 6. Визуальное описание аргумента водозаполнения для случая, когда 1 < M <N и $C_0 > C^*_0$. Обратите внимание на многоуровневую структуру столбцов для i>M.

Так как оценка (24) возрастает в $\sum_{i=M+2}^{N} \lambda_i$ и что заданное (25) убывает, min в (23) минимизируется для данного $C_0 = \sum_{i=1}^{N} \lambda_i$, когда две границы совпадают. Это дает фундаментальную нижнюю границу

$$\underline{t}_{M+1} \geqslant \frac{(M^2 C_0 - M^2 \lambda_{M+1} + 2M C_0 - M\lambda_{M+1} + C_0)|F|}{C_0((M+1)(\sum_{i=0}^{M} C_i) + M \sum_{i=M+2}^{N} C_i)} \quad (27)$$

Когда $C_0 > C^*_0$, значение $\sum_{i=1}^{M} \lambda_i$ необходимое для достижения этой границы, нарушает (19). В этом случае алгоритм устанавливает λ_i, $i<M$, к ее верхней границе $C_i/(M-1)$, и (22) становится

$$|\bigcup_{i=M+2}^{N} R_{i,M+1}(t_M, t_{M+1})| = \sum_{i=M+2}^{N} C_i t_{M+1}.$$

Когда $C_0 > C^*_0$, узлы $i \in \{M+2, \ldots, N\}$ не нужно загружать всю их информацию F_i (t_M) в узел M для достижения (22); достаточно, чтобы λ_i, $i \in \{M+2, \ldots, N\}$, было достаточно большой, чтобы $r_{i,M+2}(t) = C_i$ для всех $t \in (t_M, t_{M+1})$.

LP (5) гарантирует, что условие выполнено, при этом последовательно предоставляя как можно больше емкости сервера $(0, t_M)$ узлам $M+2, \ldots, N$.
В любом случае алгоритм 1 достигает нижней границы на t_{M+1}, сохраняя $t_1, \ldots, t_M = |F| / C_0$.

Наконец, мы заявляем после $\underline{t}_{M+1}$, каждый узел i получает со скоростью $C - C_i$ на своем интервале завершения и C_{i-1} на предыдущем интервале. Чтобы подтвердить, рассмотрим вымышленный временной интервал, когда другой узел $k \notin \{1,\ldots, N\}$ должен получить всю информацию, хранящуюся в узлах $\{1,\ldots, N\}$ (то есть, у нее нет части файла). В этом случае количество времени, которое требуется для передачи, если все узлы имеют доступ ко всему файлу, $|F| / C$, меньше количества времени, которое требуется для того, чтобы любой отдельный узел загрузил назначенную часть файла, $\lambda_i t_1 / C_i$.

В соответствии с алгоритмом 1,

$$\lambda_N \leq \lambda_i, \qquad \forall i \in \{1,\ldots,N\}, \qquad (28)$$

в том числе в том случае, если $C_0 > \overset{*}{C}_0$. Чтобы показать, что каждый узел имеет достаточно информации для полной передачи на любой временной интервал, достаточно показать, что

$$\frac{\sum_{i=1}^{M} \lambda_i}{C_0 + \sum_{i=1}^{N} C_i} \leqslant \frac{\lambda_N}{C_N} \quad (29)$$

которые могут быть $$\frac{C_0}{C} \leqslant \frac{\sum_{i=M+2}^{N} \lambda_i}{\sum_{i=M+2}^{N} C_i} \quad (30)$$

и приводит к $$C_0 \geqslant \frac{C_{M+1}(C - C_0)}{MC_{M+1} + \sum_{i=M+2}^{N} C_i} \quad .(31)$$

Эта нижняя грань на C_0 для выполняемого условия строго меньше нижней границы из-за ограничения множественности.

Таким образом, полное использование сохраняется для всех временных интервалов до (t_{N-1}, t_N) при выполнении предложенной оптимальной схемы.

ПРИЛОЖЕНИЕ В АРГУМЕНТ ПРЕТЕНЗИИ 2 $C_0 > \overset{*}{C}_0$

Максимальный объем информации, которая может перейти в набор A по времени t_{M+1}, равна

$$(C_0 + \sum_{i=1}^{M} C_i) t_{M+1} + C_{M+1} t_M - \sum_{i=M+2}^{N} F_i(t_{M+1})$$
$$+ min(t_{M+1} \sum_{i=M+2}^{N} C_i, (M+1) \sum_{i=M+2}^{N} F_i(t_{M+1})) \quad (32)$$

Как показано ранее, это выражение максимизируется относительно $\sum_{i=M+2}^{N} F_i(t_{M+1})$ для любых t_M и t_{M+1}, когда

$$t_{M+1} \sum_{i=M+2}^{N} C_i = (M+1) \sum_{i=M+2}^{N} F_i(t_{M+1})$$

Теперь рассмотрим вопрос об увеличении для$\in$ в $\sum_{i=M+2}^{N} F_i(t_{M+1})$ результате получается аналогично построенная оценка на t_{M+1} из

$$t_{M+1} \geqslant \frac{(M+1)|F| - C_{M+1} t_{M+\epsilon}}{C - C_{M+1} - \sum_{i=M+2}^{N} \frac{C_i}{M+1}}. \quad (33)$$

Когда $\sum_{i=1}^{M} t_i$ добавляется к обеим сторонам в (33), становится ясно, что соотношение между $\in$ *и* $t_1 \ldots t_M$ является ключом к минимизации суммы $\sum_{i=1}^{M} t_i$.

В частности, мы хотим построить функцию $f(\epsilon)$ такой, что

$$\sum_{i=1}^{M} t_i \geqslant f(\epsilon) \quad (34)$$

Пусть $B = (C_0 - \overset{*}{C}_0)|F|/C_0$ - это избыточные данные которые узел может отправить в узкое место, используя мощность выше C^*_0. Поскольку $\in$ соответствует дополнительной емкости, заданной узлам M+2,...N, мы можем рассмотреть B – $\in$ дополнительные данные, данные узлов 1,..., M + 1 от того, что осталось от этой избыточной емкости.

Из теоремы о множественности, не учитывая эффекта влияния $\in$ мы знаем, что плотная граница на $\sum_{i=1}^{M} t_i$ это $M|F|/C_0$. Теперь мы поддерживаем

$$t_{M+1} \sum_{i=M+2}^{N} C_i = (M+1) \sum_{i=M+2}^{N} F_i(t_M)$$

(Можно предположить $\sum_{i=M+2}^{N} F_i(t_M) = \sum_{i=M+2}^{N} (t_{M+1})$, что при любых изменениях в содержание узлов M + 2,...,N после времени t_M не имеет преимущества для завершения времени t_{M+1}).

Для $\in$ < B следует, что $B - \in$ емкость должна поглощаться непосредственно с сервера узлами 1, ..., M +1, поскольку $C_0 > \overset{*}{C}_0$ схема имеет все данные для $\sum_{i=M+2}^{N} F_i(t_M)$ приходя непосредственно с сервера.

Это означает, что узлы 1, ..., M + 1 будут перенасыщены, так что $\lambda_i > C_i / (M - 1)$ для хотя бы одного узла в множестве 1, ..., M или $\lambda_{M+1} > C_M + 1 / M$. В качестве прямого результата $t_M > | F | / C_0$, так как каждый из этих узлов не может отправлять все M узлы.

Объем информации, которая будет оставлена для отправки в набор, будет по меньшей мере $B - \in$ и может удерживаться только членами набора. Время отправки этой информации узлам в наборе которые остаются до конца, будут по крайней мере $(B - \in) / \Sigma_{i=0}^{M-1} C_i$. Пусть

$$f(\epsilon, k) = \frac{(B-\epsilon)}{\sum_{i=0}^{M-1} C_i} + \frac{k|F|}{C_0}.$$

Тогда (34) выполняется $f(\epsilon) = f(\epsilon, M)$. Подставляя эту нижнюю оценку для $\sum_{i=1}^{M} t_i$ в (33) ($\sum_{i=1}^{M} t_i$ обеих сторон), рассмотрев наихудший сценарий $t_M = f(\epsilon, 1)$, дает нижнюю оценку на $\sum_{i=1}^{M+1} t_i$ с точки зрения. Производная по отношению к $\in$

$$\frac{1}{\sum_{i=0}^{M-1} C_i + C_{M+1}} \left(\frac{C_{M+1}}{\sum_{i=0}^{M} C_i + \frac{M}{M+1} \sum_{i=M+2}^{N} C_i} - 1 \right)$$

$$+ \frac{1}{\sum_{i=0}^{M} C_i + \frac{M}{M+1} \sum_{i=M+2}^{N} C_i},$$

отрицательная, поскольку

$$C_{M+1} + \sum_{i+0}^{M-1} C_i + C_{M+1} - \left(\sum_{i+0}^{M} C_i + \frac{M}{M+1} \sum_{i=M+2}^{N} C_i \right)$$

$$= 2C_{M+1} - \left(C_M + \frac{M}{M+1} \sum_{i=M+2}^{N} C_i \right) < 0$$

Остается остаток $C_0 \le C^*_0$.

3. Влияние выбора узла на производительность

3.1. Структурированная оверлейная конструкция

При построении структурированных одноранговых сетей каждый узел выбирает соседей, которые удовлетворяют ограничениям логического идентификатора (например, префиксному соответствию или диапазону идентификаторов) и строит направленные ссылки к ним. Эти ограничения являются гибкими, так что несколько узлов являются возможными соседями для каждой записи таблицы маршрутизации. Интеллектуальный выбор соседей из множества возможных соседних узлов значительно влияет на характеристики наложения, устойчивости и балансировки нагрузки.

Проблема выбора соседства может быть сведена к задаче обобщенной минимизации затрат. Мы представляем здесь обобщенную модель затрат, которая фиксирует общие характеристики узла и линии связи при выборе соседа. В идеале оптимизация выбора соседа для узла *i* означает минимизацию суммы затрат *i* на все остальные узлы. Стоимость от и до состоит из двух факторов: затрат, связанных с промежуточными оверлейными узлами (стоимость узла: c_n) и стоимостью, налагаемой оверлейными сетевыми ссылками (стоимость края: c_e). Пусть *N* будет размер сети. Стоимость узла *i* (C_i) это:

$$C_i = \sum_{j=1}^{N} t(i,j)\, c_p(i,j) \quad \text{где}$$

$$c_p(i,j) = \sum_{n \in V(i,j)} c_n(n) + \sum_{e \in P(i,j)} c_e(e) \quad (1)$$

где $t(i,j)$ трафик от *i* до *j*, $c_p(i,j)$ это стоимость пути от *i* и до *j*, $P(i,j)$ это путь (набор ребер) от *i* и до *j*, $V(i,j)$ является набор промежуточных оверлейных узлов в пути $P(i,j)$ (он не включает *i* и *j*) *e* является краем пути $P(i,j)$, *n является узлом* $V(i,j)$, $c_n(n)$

это стоимость узла *n*, и $c_e(e)$ это стоимость края *e*. Если $t(i,j)$=0, нет стимула для узла оптимизировать

Model	**Cost (C_i)**
Random	None
Dist	$\sum_{b=1}^{N_b} c_e(i, n_b)$
Cap	$\sum_{b=1}^{N_b} c_n(i, n_b)$
CapDist	$\sum_{b=1}^{N_b} \left[c_n(n_b) + c_e(i, n_b) \right]$

Таблица 1: Изучены изученные функции затрат. $c_n(i)$ представляет задержку обработки в узле *i*. Это уменьшающая функция пропускной способности узла *i*. $c_e(i,n_b)$ представляет собой прямую задержку связи между *i* узлом и узлом n_b.

путь от *i* и *j* до. В этой модели, c_n фиксируется гетерогенность пропускной способности узла, которая является функцией полосы пропускания, мощности вычислений, времени доступа к диску и т. д. c_e фиксирует близость сети.

Для структурированных сетей, таких как Chord, Pastry и Tapestry, функция стоимости определяется как:

$$C_i = \sum_{p=1}^{N_b} \sum_{j \in R_b} t(i,j)\, c_p(i,j) \quad где$$

$$c_p(i,j) = c_n(n_b) + \sum_{n \in V(n_b,j)} c_n(n) + c_e(i,n_b) + \sum_{e \in P(n_b,j)} C_e(e)$$

где b соседний индекс, n_b является соседом, индексированным по b,N_b является число соседей, R_b представляет собой набор адресатов, направляемых через соседа n_b, $c_n(i)$ является значением стоимости узла i, $c_e(k,l)$ является стоимостью края между двумя узлами k и l, c_e (e) является краевой стоимостью e.

В зависимости от цели оптимизации, мы можем выбирать разные показатели для c_n *и* c_e ,в том числе задержек, пропускной способности, надежности, доступности, денежной стоимости или любой их комбинации. Например, выбор узлов большой мощности, поскольку соседи могут уменьшить задержку поиска и увеличить общую вычислительную мощность системы. С другой стороны, использование доступности как показателя создает более стабильную сеть.

Обратите внимание, что наша идеализированная функция стоимости предполагает полное знание сетевых компонентов и поэтому на практике не представляется возможным. Поскольку большинство протоколов одноранговой сети ориентированы на оптимизацию соседних таблиц локально, мы сосредоточимся на применении нашей функции затрат на стоимости первого оверлейного хопа. Поэтому мы фокусируемся на соседних выборах, которые рассматривают первый прыжок и оптимизируют латентность при едином трафике (t (i,j) = 1, $\forall i,j$).

В таблице 1 показаны четыре функции выбора соседей. *Random* выбирает соседей. *Dist* выбирает ближайших соседей в сети, чтобы адаптироваться к базовой топологии сети. В настоящее время *Bamboo*, *Pastry* и *Tapestry* используют этот механизм. *Cap* выбирает соседей, которые имеют наименьшую задержку обработки. *CapDist* выбирает соседей, которые дают наименьшую совокупную задержку, которая является суммой задержки обработки узла и задержки наложения наложения.

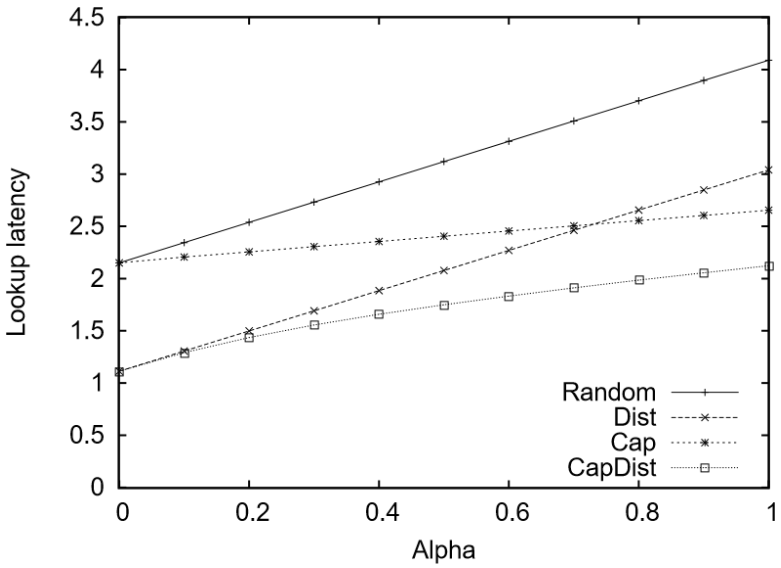


Рисунок 1. Среднее время ожидания для равномерного распределения задержки обработки.

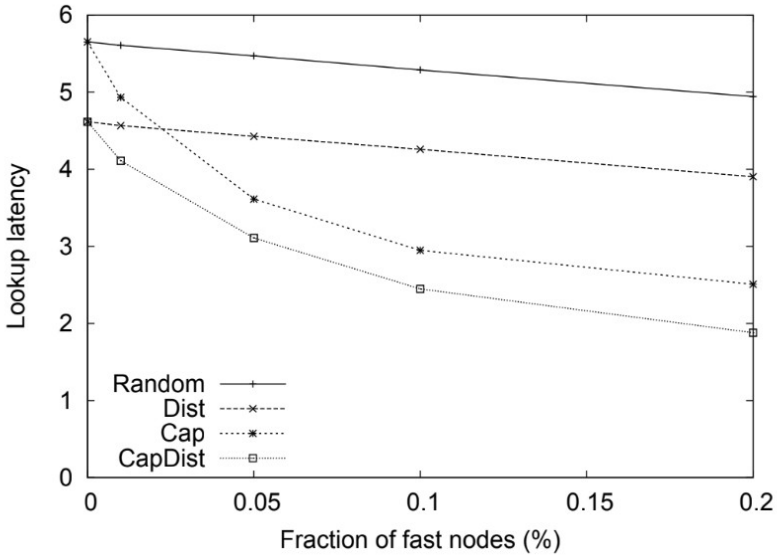


Рисунок 2. Средняя латентность ожидания для распределения задержки бимодальной обработки.

3.2. Симуляция результатов

В этом разделе мы представляем результаты моделирования, которые определяют количественные оценки эффективности использования интеллектуальных алгоритмов выбора соседей. Мы также изучаем влияние таких алгоритмов на статическую устойчивость полученного наложения к рандомизированным отказам и целенаправленным атакам.

3.2.1 Настройки моделирования

Мы моделируем протоколы Tapestry и Chord в качестве представителей их соответствующих геометрий (дерева и кольца). Когда каждый узел оптимизирует свою функцию стоимости, он выполняет произвольную выборку для выбора соседей и выбора наилучшего среди выборок. В наших экспериментах мы используем 32 образца для каждого уровня маршрутизации в Гобелене или каждом пальце в аккорде.

В наших симуляциях используются топологии транзитных заглушек 5100, сгенерированные с использованием библиотеки GT-ITM. Мы строим накладки из аккордов и гобеленов из 4096 узлов, помещая оверлейные узлы в случайные физические местоположения. Мы собираем результаты с 9 различными конфигурациями для GTITM, генерируем 3 топологии транзитных заглушек и выбираем

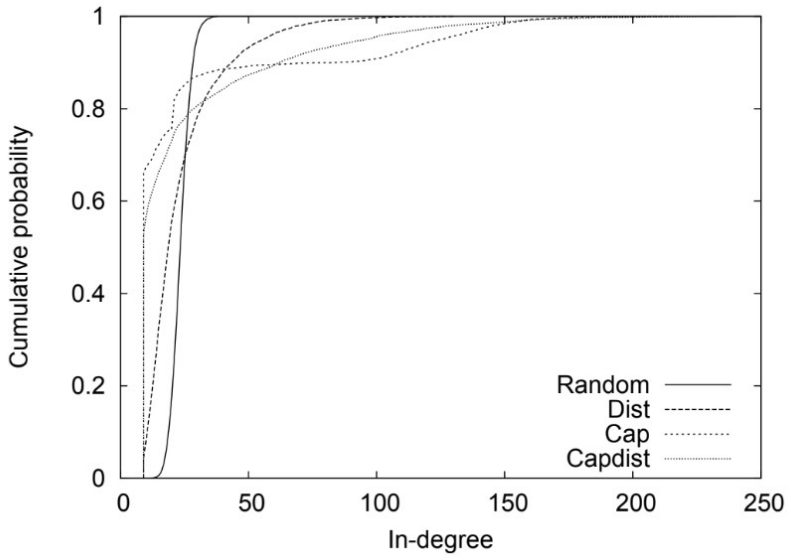


Рисунок 3. CDF числа входящих ребер для равномерного распределения задержки обработки.

3 места размещения наложения на каждую топологию.

Для наших экспериментов Chord каждый узел пересылает сообщения ближайшему соседству, ближайшему к месту назначения в пространстве идентификатора. Поиск завершается с ошибкой, если все соседи до места назначения в пространстве имен сбой. Для Tapestry каждый узел пересылает сообщения первому живому соседу, сопоставляющему еще одну цифру префикса. Если все первичные и резервные ссылки в записи маршрутизации терпят неудачу, поиск не выполняется.

3.2.2 Производительность

Начнем с количественного определения эффектов алгоритмов выбора соседства на производительность. Мы рассматриваем два разных распределения: uniform и bimodal. Поскольку результаты Tapestry и Chord в обоих случаях одинаковы, мы будем показывать только результаты Tapestry.

Начнем с назначения задержки обработки узла из грубо-зернистого равномерного распределения. Мы равномерно распределяем задержку обработки . Это является максимальной задержкой обработки $\frac{\alpha}{10}, \frac{2\alpha}{10}, \frac{3\alpha}{10}, \ldots, \alpha$. На рисунке 1 показана средняя латентность поиска по всем парам узлов в Tapestry. Используя уникальность сети и гетерогенную емкость, *CapDist* обеспечивает наилучшую производительность поиска. Когда изменение задержки обработки велико (α=1c), CapDist выполняет 30% лучше, чем *Dist* и 48% лучше, чем *Random*. Если вариации не существует (т.е. = 0c), *Dist* и *CapDist* используют сетевую близость, чтобы превзойти *Random* и *Cap*.

Теперь мы рассмотрим бимодальную модель для пропускной способности, где узлы являются быстрыми или медленными. Быстрые узлы обрабатывают 100 поисковых сообщений в секунду, а медленные узлы обрабатывают 1 сообщение в секунду. Рисунок 2 показывает, что по мере того, как мы меняем долю быстрых узлов от 0% до 20%, выбор соседства с использованием емкости (*Cap* и *CapDist*) способствует маршрутам через быстрые узлы и обеспечивает лучшую производительность. В тех случаях, когда разница в мощности обработки чрезвычайно высока, мы ожидаем, что использование мощностей на быстрых узлах будет ограничено ограничениями маршрутизации протокола и

развертывание виртуальных узлов необходимо для полного использования избыточной вычислительной мощности.

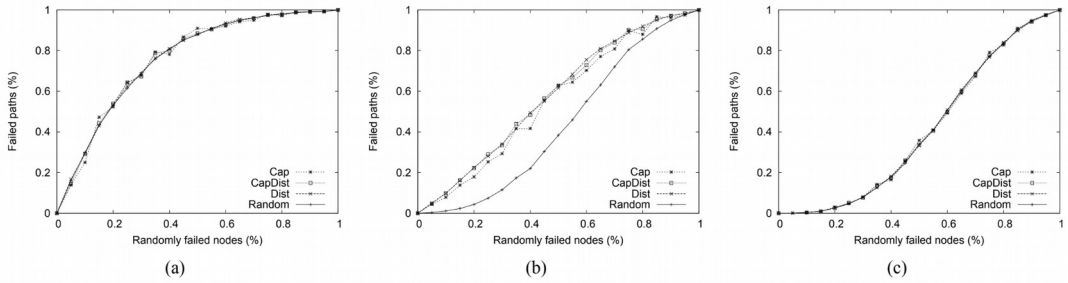


Рисунок 4. Tapestry при неудачных ошибках узла. (a) Выбор гобеленов с разными соседями по одной первичной ссылке (например, *Dist*: первичная ссылка, выбранная для оптимизации *Dist* функции стоимости), (*b*) Выбор Tapestry с разными соседями по одной первичной ссылке и двум резервным ссылкам (например, *Dist*: все три ссылки, выбранные для оптимизации *Dist* функция стоимости), (*c*) Выбор Tapestry с разными соседями по одной первичной ссылке и случайное выбор двух резервных ссылок (например, *Dist*: первичная ссылка, выбранная для оптимизации *Dist* функции затрат и двух резервных ссылок, выбранных случайным образом).

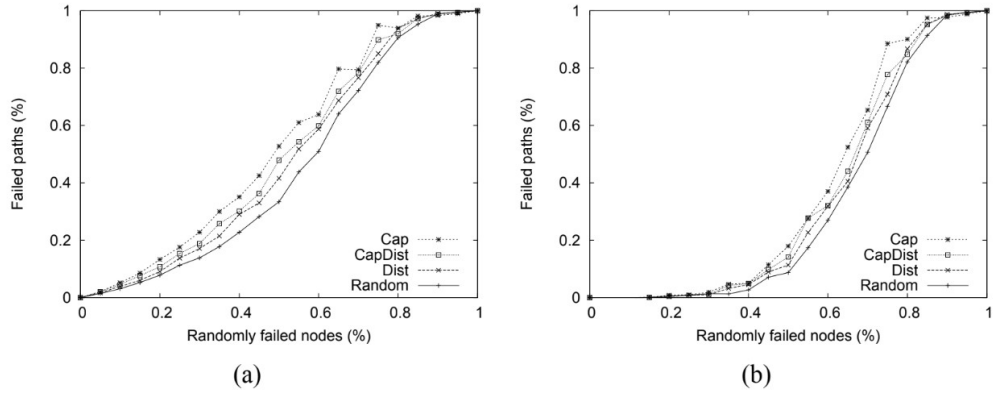


Рисунок 5. Chord при неудачных ошибках узла. (a) Выбор Chord по выбору пальца на столе для пальцев, (b) Chord варьирующего finger на finger table и имеющих 4 последовательных соседей.)

Использование оптимизации задержки создает уровни входящих узлов. Узлы вблизи центра сети (т. е. транзитные домены) и узлы с большой пропускной способностью являются предпочтительными и минимизируют задержку пути за счет использования каналов с низкой задержкой или низкой задержки обработки. На рисунке 3 показана кумулятивная функция распределения (CDF) узлов в градусах в сетях Tapestry с различными алгоритмами выбора соседства. В отличие от *Random*, результаты оптимизированных по цене наложений показывают медленные переходы и длинные хвосты. Мы также отмечаем, что CDF узлов в транзитных доменах более искажен и имеет более длинные хвосты, чем узлы в доменах-заглушках.

3.2.3. Статическая устойчивость

В этом разделе мы исследуем с помощью моделирования влияние, которое алгоритмы выбора соседей оказывают на статическую устойчивость.

Мы измеряем устойчивость как долю всех пар *live* конечных точек, которые все еще могут пересекаться друг с другом через оверлей после внешнего события, либо случайные сбои узлов, либо целенаправленные атаки. Мы предполагаем, что атаки направлены на удаление узлов с наивысшей степенью, чтобы максимально увеличить общую доступность сети. Для этих экспериментов мы

предполагаем, что узлы имеют равномерное распределение задержки обработки α = 0,5 с.

Для Tapestry мы изучаем устойчивость базового протокола, базового протокола и дополнительных маршрутов резервного копирования (все они выбраны с использованием ряда алгоритмов выбора соседей) и базового протокола и резервных маршрутов, выбранных случайным образом. Обратите внимание, что добавление двух резервных ссылок увеличивает число соседей. Для Chord мы изучаем базовый протокол и базовый протокол плюс последовательные соседи.

3.2.4. Случайные сбои узла

Сначала мы рассмотрим влияние рандомизированных отказов узлов. В общем, мы ожидаем, что использование алгоритмов выбора, которые предпочитают узлы с высокой пропускной способностью, приводит к большей иерархии в сети, где многие более слабые узлы соединены высоко связанными узлами большой мощности. В таких случаях мы ожидаем, что случайные сбои будут отключать более узкие узлы от сети, но оказывая относительно низкое влияние на общую связь.

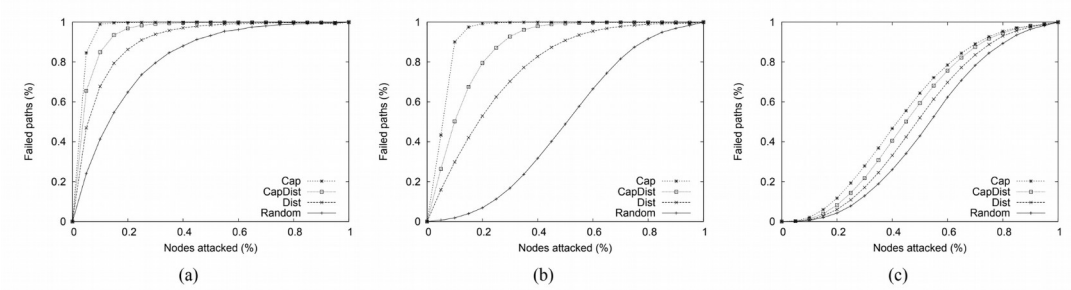


Рисунок 6. Узлы Tapestry под атакой. (a) Выбор Tapestry с разными соседями по одной первичной ссылке (например, *Dist*: первичная ссылка, выбранная для оптимизации *Dist* функции стоимости), (b) Tapestry, изменяющий выбор соседа по одной первичной ссылке и двум резервным ссылкам (например, *Dist*: все три ссылки, выбранные для оптимизации *Dist* функции стоимости), (c) Tapestry, изменяющий выбор соседа по одной первичной ссылке, и случайный выбор двух резервных ссылок (например, *Dist*: первичная ссылка, выбранная для оптимизации *Dist* функции затрат и двух резервных ссылок, выбранных случайным образом).

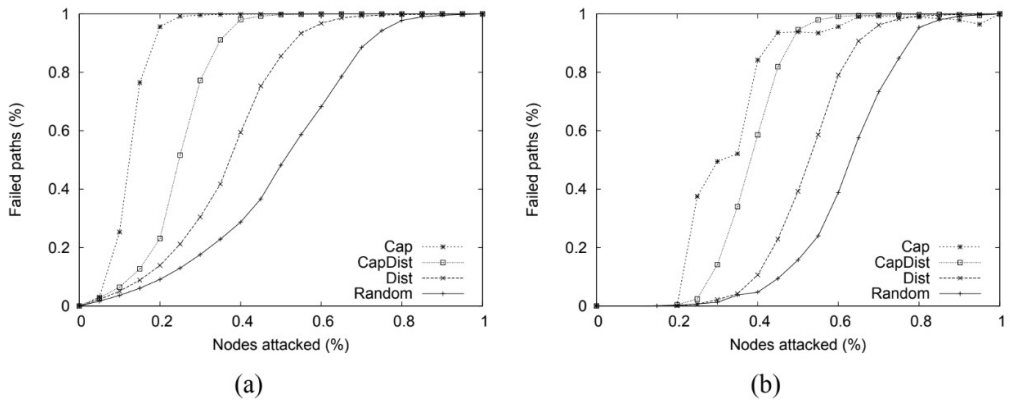


Рисунок 7. Узлы Chord под атакой. (a) Выбор Chord по выбору finger на finger table, (b) Выбор Chord на finger и выбор из 4 последовательных соседей.

На рисунках 4 и 5 показаны отказоустойчивость Tapestry и Chord, соответственно. Удивительно, но мы видим, что отклонение отказов немного

зависит от соседних настроек. Более жесткие ограничения на исходящие каналы структурированных одноранговых сетей допускают меньшее изменение в результирующей топологии, чем неструктурированные сети. Каждый узел имеет как минимум 0 (log*N*) исходящие ссылки, а рандомизированное именование также сглаживает распределение исходящих ребер. Поскольку каждый поиск 0 (log*N*) принимает переходы независимо от функций стоимости выбора соседа, вероятность встретить случайно неудавшиеся узлы в поиске будет аналогичной.

Добавление резервных ссылок в Tapestry и последовательных соседей в Chord значительно улучшает отказоустойчивость. Обратите внимание, что в Tapestry кривые отказа изменяются от экспоненциально растущей кривой до гладкой S-образной кривой из-за избыточности пути, когда резервные ссылки добавлены.

3.3. Атаки целевого узла

Хотя структурированные одноранговые протоколы определяют минимальное количество исходящих ссылок на узел, количество входящих ссылок узла неограничено. Это означает, что алгоритмы выбора соседства с учетом мощности будут искажать сеть, так что мощные узлы имеют значительно более высокие значения в градусах, чем более слабые узлы. Это означает, что, как и неструктурированные сети, так и структурированные одноранговые оверлеи, которые рассматривают возможность выбора соседей, уязвимы для атак.

Как показано на рисунках 6 и 7, атакующие узлы с высокой степенью серьезности сильно влияют на сетевое подключение. *Random* показывает лучший перенос атаки между соседними выборами. У *CapDist* хуже атака, чем у *Dist*, хотя он имеет лучшую производительность, как показано в разделе 4.2.2. Этот результат демонстрирует фундаментальный компромисс между производительностью и устойчивостью к атакам при построении структурированного оверлея. Повышение производительности от алгоритмов выбора соседей увеличивает изменчивость индексов среди узлов. Узлы с большой пропускной способностью или узлы, расположенные вблизи центра сети, заканчиваются высокими степенями и имеют непропорционально большое влияние на сетевое соединение, когда они терпят неудачу.

3.4. Анализ дополнительной избыточности

Мы наблюдаем, что добавление резервных ссылок или последовательных соседей может значительно увеличить уровень допуска. Случайный выбор резервных ссылок в Tapestry и последовательных соседей в Chord позволяет избежать маршрутизации горячих точек, которые уязвимы для целенаправленных атак. Например, в «Tapestry» стопоритизированные резервные ссылки менее эффективны для повышения устойчивости к атакам, чем для случайных резервных ссылок. Уплата дополнительных затрат на поддержку дополнительных ссылок повышает статическую устойчивость к целенаправленным атакам.

Другим методом повышения толерантности является наложение максимальной степени на каждый узел, но это ограничение увеличивает задержку поиска. Ограничение степени наложения узлов может быть использовано в качестве защиты от атак Eclipse.

Мы представляем обобщенную модель для выбора соседей, которая включает в себя показатели для близости сети и доступных ресурсов (мощности) и показывает, что при учете этих факторов может привести к значительным успехам в производительности маршрутизации, эти выгоды приносят их

связанные с этим издержки. Мы находим, что выбор алгоритма выбора соседей управляет компромиссом между производительностью и устойчивостью к атакам.

Оптимизированные структурированные накладки имеют несбалансированные структуры. Эти оверлеи не связывают количество входящих ссылок на узел. Таким образом, центральные узлы в сети или узлы с большим количеством ресурсов будут иметь гораздо более высокую степень независимости, чем другие. Если атакуются узлы с высокой степенью вероятности, воздействие на сетевое соединение очень тяжелое. С другой стороны, минимальная степень отклонения означает даже для наложений, которые оптимизируются в направлении близости или доступных ресурсов, большинство узлов достигают достаточной устойчивости к случайным ошибкам.

4. Информационная теория сложных систем

Сложные системы часто довольно слабо определяются как системы, состоящие из множества различных компонентов в сильном взаимодействии и демонстрирующие некоторую самоорганизацию и возникающее структурированное коллективное поведение. Тем не менее, термин "сложность" так часто используется без квалификации многими авторами, как научными, так и ненаучными, что, к сожалению, почти утратил свое значение.

Нет ни краткого определения сложной системы, ни консенсусного набора определенных свойств, связанных с понятием сложной системы, такие как нелинейность, хаотическое поведение, наличие обратной связи, робастность, возникающий порядок и иерархическая организация, среди прочего, обычно упоминаются в литературе по мере необходимости, но недостаточно.

Широко известно, что для многих систем, далеких от термодинамического равновесия, вблизи фазовых переходов, несколько термодинамических величин, таких как удельная теплоемкость, сжимаемость и магнитная восприимчивость, представляют собой сингулярное поведение в критической области с асимптотическими различиями, характеризующимися критическими показателями, которые определяют „степенные законы". Это наблюдение побудило физиков приравнивать степенные законы как таковые к сложности и утверждать, что некоторая система сложна, потому что она демонстрирует распределение степенных законов размеров событий.

Первым применением этой концепции стало масштабирование поведения "толстых хвостов", наблюдаемое в динамике индекса Standard & Poor " s 500 (Mantegna and Stanley 1995). Несколько лет назад, в поисковых запросах Google по компонентам Dow Jones Industrial Average (DJIA) (Kristoufek 2015) наблюдалась корреляция степенного закона с показателем между 0,8 и 1,1. Аналогичным образом, «жирный хвост» с показателями около 2 был обнаружен в распределении колебаний валютных курсов в международной торговле на валютном рынке (FOREX) (Chakraborty, et al. 2016) и в волатильности цены Bitcoin.

Однако, несмотря на свою популярность, никто никогда не демонстрировал какой-либо связи между законами власти и любой формальной мерой сложности. Кроме того, хотя и верно, что в статистической механике равновесия, когда система не сложна по обычным меркам, не найдено никаких степенных законов, уже полвека известно, что существует множество способов генерации степенных законов без сложности. (Шализи 2006, 61). Что еще хуже, Шализи показывает, что большинство вещей, которые претендовали на власть законы на самом деле и другие виды распределение с тяжелым хвостом, ошибочно принятый как таковой из-за неадекватных статистических оценок

соответствия данных в силу закона, видимо, забыв, что любая гладкая кривая выглядит как прямая линия, если внимание ограничивается достаточно на небольшой области, которая, по некоторым не-степенным законам распределения может продлить на распространяться на кратные порядки величины.

Некоторые авторы предполагают, что осложненные системы являются сильными кандидатами на сложные системы, или, по крайней мере, что осложненные системы должны быть сложными. Однако осложненность и сложность имеют совершенно разные черты: например, атомные часы - это очень сложное устройство, но очень предсказуемое и, следовательно, совсем не сложное; напротив, плоский двойной маятник - это механическая система, которая проявляет сложное, хаотичное поведение (Stachowiak и Okada 2006), несмотря на свою простую конструкцию. С другой стороны, сильно упорядоченные системы имеют лишь небольшой диапазон допустимого поведения и не могут быть сложными, в то время как сильно неупорядоченные системы имеют независимые, слабо взаимодействующие части и не являются сложными.

Эта связь между сложностью и порядком (измеряется энтропией) является изображенное на Рис. 1. В нижней крайности упорядоченные системы, такие как идеальный кристалл, имеют низкую энтропию и низкую сложность. В высшей крайности совершенно случайные системы, такие как справедливое подбрасывание монеты, имеют высокую энтропию, но низкую сложность. Таким образом, в сообществе сложных систем существует единое мнение, что любая удовлетворительная мера сложности должна быть минимальной как для полностью случайных, так и для упорядоченных систем, поскольку они допускают краткие описания, и присваивать наивысшее значение сложности системам, которые не являются ни полностью случайными, ни полностью упорядоченными, лежащими где-то посередине (Ladyman).

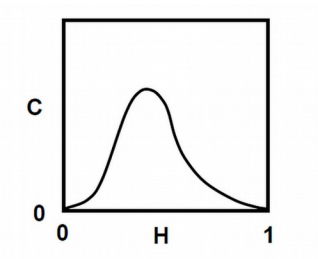


Рисунок 1. Интуитивное соотношение сложности и энтропии.

Ladyman, и другие проанализированы различные показатели сложности, научная литература показала, что многие из них не являются вычислимыми, в то время как другие не являются подходящими мерами сложности физической системы, поскольку они в конечном счете являются мерами случайности, а не сложности и дают максимальные значения совершенно случайным процессам обработки данных (Ladyman и другие соавторы). Эти авторы пришли к выводу, что мера, которая лучше всего отражает качественное понятие сложности системы - это статистическая сложность, представил Джеймс Кратчфилд и Карл Янг (1989).

Соответственно, здесь мы будем использовать этот альтернативный подход к сложности, так как он больше подходит для физических систем (Crutchfield 2011). Он адаптирует и расширяет идеи от теории дискретных вычислений до вывода сложности в динамических системах, а именно через понятие "машины" - устройства для кодирования структур в дискретных процессах (Crutchfield 1994). После того, как мы реконструировали машину, мы можем сказать, что мы понимаем структуру процесса и говорить о структуре

исходного процесса относительно сложности реконструированной машины (Crutchfield 1994).

Вход для вычисления задается начальной физической конфигурацией системы; выполнение вычисления соответствует временной последовательности изменений во внутреннем состоянии системы, и результат вычисления отсчитывается, наконец, в состоянии, в котором система расслабляется (Crutchfield 1994). Эти вычислительные модели, реконструированные по наблюдениям данного процесса, называются ϵ-машинами, чтобы подчеркнуть их зависимость от ограниченной точности ϵ измерительного прибора (Crutchfield and Young 1989).

Три теоремы оптимальности гарантируют нам, что ϵ-машинное представление процесса захватывает все свойства процесса: будучи его оптимальным предиктором и минимальным представлением по сравнению со всеми другими оптимальными предикторами, и являясь любым другим минимальным оптимальным предиктором, эквивалентным ϵ-машине.

Система может характеризоваться набором наблюдаемых случайных величин, имеющих в каждый момент времени t в прошлом $X^- = ..., X_{t-2}, X_{t-1}$ и в будущем $X^+ = ..., X_{t+1}, X_{t+2}, ...$. Возможно существуют , различные истории измерений x^-_i , из которых строится прогноз, то есть условное распределение вероятностей $P(X^+|x^-_i)$ к будущим значениям X^+ с помощью которого любой может предсказать все, что предсказуемо в системе. Две истории x^-_1 *и* x^-_2 считаются эквивалентными, если $P(X^+|x^-_1) = P(X^+|x^-_2)$; могут быть построены классы σ эквивалентных историй, и можно перестать различать истории, которые делают одни и те же прогнозы. Эти классы эквивалентности называются «причинными состояниями» процесса, поскольку они сохраняют соответствующую информацию для прогнозирования будущего.

С учетом этих рамок статистическую сложность можно рассчитать следующим образом:

$$C\mu = - \sum_{\sigma \in S} P(\sigma) \log_2 P(\sigma)$$

где S является множеством всех причинных состояний и $P(\sigma)$ является распределением вероятности по ним.

Здесь может быть полезно применить эту формулу к предыдущим двум примерам, следуя расчетам Crutchfield.

Кристалл имеет одно единственное состояние (его неизменное характерное пространственное расположение его составляющих атомов, распространяющихся во всех направлениях) и, следовательно, существует только один класс σ эквивалентных историй, $P(\sigma) = 1$, и статистическая сложность $C\mu = 0$ по желанию.

Для другого примера, в любой момент времени будущие наблюдения X^+ за справедливой монеты непредсказуемы, все разные истории измерений эквивалентны в их неспособности предсказать, существует только один класс σ, $P(\sigma) = 1$, и еще $C\mu = 0$.

Представляя третий пример, рассмотрим процесс периода-2, такой как идеально периодические часы маятника, которые после ввода в действие обнаруживают два повторяющихся причинных состояния, *тик* и *так,* теперь, есть две разные истории, ведущие к каждому из этих состояний $P(\sigma_{tick}) = P(\sigma_{tack}) = 1/2$, и, в отличие от первых двух примеров, он показывает себя довольно удивительно, как структурно сложный процесс с $C\mu = 1$.

Эти результаты показывают, что статистическая сложность Crutchfield и Young удовлетворяет интуитивным требованиям к мере сложности, рассмотренной выше (Рис. 1) и поддержать наш методологический выбор ИТ для этой работы. Теперь мы приступим к измерению статистической сложности потока блоков, предоставляемых блокчейном, чтобы сделать вывод о том, можно ли считать экосистему блокчейна сложной системой.

4. Доказательство эффективности алгоритма POW.

Хотя концепция децентрализованной цифровой валюты, а также альтернативные приложения, такие как реестры собственности, существуют уже несколько десятилетий, консенсусный алгоритм Сатоши Накамото, известный как «доказательство работы» (PoW), был прорывом, поскольку он одновременно обеспечивал:

1. Простой и умеренно эффективный консенсусный алгоритм для коллективного соглашения об активах обновлений состояния журнала Bitcoin .
2. Механизм, который позволил свободно входить в консенсусный процесс и предотвращать атаки Sybil.

С технической точки зрения любая криптовалюта на основе PoW, такая как Bitcoin или Ethereum, может рассматриваться как система перехода состояния, где есть «состояние», состоящее из статуса собственности всех существующих токенов и «функции перехода состояния», которая принимает предыдущее состояние σ_{t-1} и действительную транзакцию T и выводит новое состояние σ_t в результат.

$$\sigma_t \equiv \Upsilon(\sigma_{t-1}, T)$$

где Υ - функция перехода состояния.

Цепочку блоков Bitcoin можно рассматривать, как ϵ-машину-генератор бесконечной струны (См. Рис. 1), которая колеблется между двумя состояниями примерно каждые 10 минут:

1. σ_m (состояние добычи). Новый блок был просто включен в блокчейн, а машина начинает добычу нового блока, который включает большинство ожидающих транзакций, собранных со всего мира, в пул транзакций. Хэши генерируются и тестируются в соответствии с целевым уровнем сложности сети.

2. σb (состояние вещания): Найден ноль, который приводит к хешу, меньшему, чем цель,
проверенный блок передается в сеть P2P для включения в блок-цепь. Если происходит разветвление (форк) цепочки блоков, то глобальная сеть Bitcoin в конечном счете сойдется к новому согласованному состоянию добычи.

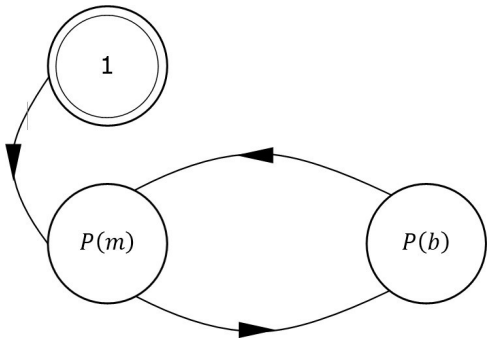


Рисунок 1. Цепочка блоков Bitcoin, рассматриваемая, как ϵ-машина-генератор бесконечной струны, которая колеблется между двумя состояниями σ_m *и* σ_b. Вписанная окружность указывает начальное состояние, соответствующее генерации начального блока.

Статистическая мера сложности цепочки блоков Bitcoin ранее оценивалась используя обычное «число ведущих нулевых байтов в хэше», аппроксимации, для тогдашней текущей скорости производства сети 4,265,775.24 Tera (10^{12}) хешей в секунду, как $C_\mu \cong 1.56 \times 10^{-20}$.

Более простая и еще более точная процедура состоит в том, чтобы рассмотреть, что новые блоки создаются примерно каждые 10 минут, т. е. 600 секунд, и, следовательно, для одного и того же значения сетевой производительности это занимает в среднем $(4.27 \times 10^{18}) \times 600 \cong 2.56 \times 10^{21}$ хэшей, чтобы найти тот, который ниже цели*.

**Цель рассчитывается путем деления максимальной цели, используемой SHA-256 (которые должны логически соответствовать 256 1 двоичные цифры, но, поскольку Bitcoin сохраняет цель как тип с плавающей запятой, она усекается примерно до 2^{224} который может быть представлен, как 64-х шестнадцатеричный хэш 0x00000000FFFF0000000000000000000000000000000000000000000000000000, «0x» - обычный префикс для обозначения шестнадцатеричных цифр) по сложности.*

Поэтому, вероятность состояния вещания это $P(\sigma_b) = 1 / (2.56 \times 10^{21}) \cong 3.9 \times 10^{-22}$, в то время как состояние добычи σ_m это $P(\sigma_m) = 1 - P(\sigma_b) \cong 1 - 3.9 \times 10^{-22}$, статистическая сложность C_μ в результате $C_\mu = -((1 - 3.9 \times 10^{-22})\log_2(1 - 3.9 \times 10^{-22}) + (3.9 \times 10^{-22})\log_2 (3.9 \times 10^{-22}))$

$$C^\mu \cong 2.83 \times 10^{-20},$$

значение, которое почти в два раза больше полученного ранее, но имеет тот же порядок, и приводит к такому же выводу, что цепочка блоков Bitcoin вряд ли может считаться сложной системой.

Прямолинейно применяя эту процедуру к нескольким другим криптовалютам PoW, мы получаем результаты, показанные в Таблице 1.

Валюта	**Bitcoin**	**Ethereum**	**Bitcoin Cash**	**Bitcoin Gold**	**Litecoin**	**Dash**	**Monero**	**Ethereum Classic**
Время блока	10 min	0,25 min	10 min	10 min	2,5 min	2,5 min	2 min	0,25 min
Hashrate (hash/s)	$2{,}78\times10^{19}$	$2{,}77\times10^{+14}$	$4{,}62\times10^{+18}$	$3{,}50\times10^{+07}$	$2{,}98\times10^{+14}$	$1{,}80\times10^{+15}$	$4{,}32\times10^{+08}$	$7{,}70\times10^{+12}$
C_μ	$4{,}51\times10^{-21}$	$1{,}28\times10^{-14}$	$2{,}62\times10^{-20}$	$1{,}70\times10^{-09}$	$1{,}27\times10^{-15}$	$2{,}19\times10^{-16}$	$7{,}15\times10^{-10}$	$4{,}17\times10^{-13}$

Валюта	**Zcash**	**Vertcoin**	**Dogecoin**	**Feathercoin**	**BlackCoin**	**Namecoin**	**Auroracoin**
Время блока	10 min	0,25 min	10 min	10 min	2,5 min	2,5 min	2 min
Hashrate (hash/s)	$2{,}78\times10^{19}$	$2{,}77\times10^{+14}$	$4{,}62\times10^{+18}$	$3{,}50\times10^{+07}$	$2{,}98\times10^{+14}$	$1{,}80\times10^{+15}$	$4{,}32\times10^{+08}$
C_μ	$4{,}51\times10^{-21}$	$1{,}28\times10^{-14}$	$2{,}62\times10^{-20}$	$1{,}70\times10^{-09}$	$1{,}27\times10^{-15}$	$2{,}19\times10^{-16}$	$7{,}15\times10^{-10}$

Таблица 1. Статистическая мера сложности C_μ, рассчитанная для нескольких криптовалют использующих PoW.

Эти чрезвычайно низкие статистические сложности приводят нас к выводу, что цепочки блоков на основе PoW, в общем, вряд ли могут считаться сложными, подтверждая и распространяя на все эти криптовалюты - заявление Накамото о Bitcoin, что «сеть устойчива по своей неструктурированной простоте» . Функционирование этих цепей может рассматриваться как алгоритмически сложное, но не комплексное.

PoW, несомненно, сыграл решающую роль в рождении главного прорыва Накамото. Однако его вычислительно-энергоемкий характер означает, что криптовалюты PoW зависят от потребления энергии, что приводит к значительным издержкам в работе сетей, которые несут пользователи посредством комбинации ставок инфляции и транзакций. По мере уменьшения в сети Bitcoin ставки вознаграждения, которое получают майнеры, повышается уровень комиссионных сборов, для подддержания уровня безопасности сети.

По этой причине недавно появился всплеск популярности криптовыделения, который использовал альтернативный алгоритм, известный как «Proof-of-Stake» (PoS) для выбора создателей блоков. В отличие от криптовалют на основе PoW, где майнеры должны решать сложные криптографические головоломки, чтобы иметь возможность создавать блоки и получать за это вознаграждение, в криптовалютах на основе PoS создатель следующего блока выбирается детерминированным (псевдослучайным) способом и вероятность того, что выбран аккаунт, зависит от его богатства (ставки). Другими словами, PoS вычисляет вес узла как пропорциональный его валютнымому резерву, а не его вычислительным ресурсам. По этой причине в случае криптовалют на основе PoS, блоки обычно называются «кованными» (от от англ. Forging (фожинг) — ковка) или процесс называют «чеканка монет (минтинг)».

Чистые криптовалюты POS, такие как Nxt, выбирают учетную запись, которая имеет право генерировать следующий блок в соответствии с количеством монет в учетной записи; чем богаче аккаунт, тем выше вероятность того, что он сможет сгенерировать следующий блок и получить соответствующие транзакционные сборы. Привычно предположить, что эта вероятность должна быть точно пропорциональна балансу счета, хотя это не совсем верно для Nxt

В Nxt, чтобы участвовать в процессе форжинга блоков, каждая активная учетная запись k извлекает первые 8 байтов результата применения 8 раз в последовательности хеширования SHA256 к его открытому ключу и генерирующей сигнатуре текущего блока, значение, которое упоминается, как эта конкретная учетная запись, которое имеет значение H_k . Поскольку это зависит от открытого ключа учетной записи, это зависит от учетной записи, пытающейся подделать поверх определенного блока. Даже если сюда не задействованы генераторы псевдослучайных чисел, поскольку результат хэш-функции практически непредсказуем тем не менее, все же разумно рассматривать хиты H_k как *i.i.d. случайные величины* с равномерным распределением на этом интервале.

Кроме того, каждая учетная запись k вычисляет свое целевое значение на основе ее текущего эффективного баланса как

$$T_k = T_b \times S \times B_k$$

где:

T_k - новое целевое значение для k^{th} учетной записи. Он растет с каждой секундой, которая проходит с момента времени предыдущего блока, ограниченным проверкой, жестко закодированной в протоколе Nxt, до максимального значения $2^{64} / (2 \times 60) = 1.53722867310^{17}$ и минимум половину последнего целевого значения базовой базы блока.

T_b - является базовым целевым значением, обычно выражаемым в процентах от базовой цели блока генезиса (153722867,3), изменяется от блока к блоку и выводится из предыдущей целевой базы базы T_p , S', среднее время, которое требовалось для генерации последних 3 блоков и трех констант *maxratio, minratio* и γ, используя формулу, которая плавно увеличивает или уменьшает новую базовую цель в зависимости от предыдущего блока, заняв меньше или более одной минуты, которая должна быть сгенерирована, поэтому обеспечение генерации время блокировки между блоками по 60 секунд в среднем.

S - это время, прошедшее через секунды, так как последний блок был сгенерирован. Это одинаково для всех учетных записей.

B_k - является эффективным балансом (долей) k^{th} счета. Чтобы избежать перетасовки атак, на этот баланс рассчитывается только количество не менее 1000 NXT, которое было подтверждено не менее 1440 раз за последние 24 часа.

Чтобы заслужить право на «форжинг» (генерацию) блока, каждая активная учетная запись Nxt *k* «конкурирует», ожидая, что *S* фактор в целевой формуле выше увеличивается с каждой секундой, что никакой блок не генерируется, пока его конкретное целевое значение T_k не превысит его собственное значение случайного попадания H_k. Обратите внимание, что чем больше ставка B_k, тем выше и быстрорастущая будет ее целевая T_k, что упростит превзойти ее хит H_k. Другими словами, в Nxt «случайность» учетной записи для создания блока зависит только от текущей «ставки» (которая является собственностью каждой учетной записи), времени с момента последнего блока (который разделяется всеми поддельными учетными записями) и базовое целевое значение (которое также разделяется всеми учетными записями).

В отличие от Bitcoin, вместо глобальной цели, против которой узлы продолжают добывать свои *nonces,* пока не будет найден тот, который меньше цели, в Nxt отдельные хиты вычисляются заранее, а новые целевые увеличенные значения генерируются каждую секунду, до тех пор, пока не будет найдено одно удовлетворяющее значение условию *hit* < *target* .

Поэтому блокчейн на основе PoS также можно рассматривать как машину с бесконечной цепочкой, которая каждую минуту колеблется между двумя состояниями:

1. σ_t (*состояние таргетинга*): новый блок был включен в блок-цепочку. Каждый активный форжинг счет *k* генерирует свое собственное значение случайного попадания H_k и начинает генерировать новое, увеличивая индивидуальное целевое значение T_k каждую секунду, когда не генерируется ни один блок, пока некоторые из них не превысят свои собственные значения хита.

2. σ_b (*состояние вещания*): несколько учетных записей попали в их собственную цель и выиграли право форматировать блоки-кандидаты. Каждый из них связывает до 255 неподтвержденных транзакций в новый блок вместе со всеми его необходимыми параметрами и передает его в сеть в качестве кандидата для цепочки блоков. Если было создано несколько блоков-кандидатов, блок с наивысшим накопленным значением сложности в конечном счете выиграет и будет вставлен в верхней части цепочки блоков.

Следовательно, применяя к Nxt одну и ту же процедуру для оценки ее статистической сложности *C*μ, поскольку блок подделывается примерно каждые 60 секунд, вероятность узла, вычисляющего индивидуальную цель, которая больше, чем ее попадание, $P(\sigma_t) = 1 / 60 \cong 1.67 \times 10^{-2}$ и $C\mu \cong 0.122$,

значение показателя статистической сложности, которое на 10-20 порядков больше, чем значение, указанное в таблице 1, и, следовательно, вызвает серьезную обеспокоенность в связи с тем, что Nxt может войти в хаотический режим в любое время без предварительного уведомления.

Существуют также другие протоколы PoS с концептуально различными реализациями. Например, вероятность форжинга может также зависеть от времени, в течение которого монеты находились на счете, без передачи (так называемый возраст монеты). Возраст монеты, можно просто определить как период времени в валюте. Если Боб получает 10 монет от Алисы и удерживает их в течение 90 дней, можно сказать, что Боб накапливает 900 монетных дней из возраста монеты. Кроме того, когда Боб переводит 10 монет, которые он получил от Алисы, то возраст монеты Боба, накопленный этими 10 монетами потребляется (или уничтожается).

Концепция «возраста монеты» была изветна Накомото еще в 2010 году и использовалась в Bitcoin , чтобы приоритизировать транзакции, например, хотя в нынешней модели безопасности Bitcoin она не играла существенной роли.

Скотт Надаль и Санни Кинг (псевдоним) самостоятельно заново открыли концепции PoS и «возраста монеты» в октябре 2011 года, в результате чего, понимая, что PoS действительно может заменить большинство функций PoW с тщательной реорганизацией модели минтинга и безопасности Bitcoin .

Другим примером вариации PoS является доказательство скорости Reddcoin of Velocity (PoSV), которое предполагает «поощрять как, за владение (Stake), так и за активность (Velocity)», которые непосредственно соответствуют двум основным функциям Reddcoin как реальной валюты: хранилище ценности и средство обмена».

Существуют также гибридные реализации PoW + PoS, в которых PoW майнинг работает как устойчивый канал распространения криптовалюты, так и механизм защиты от сбоев в сети. Поскольку вознаграждения блока PoW снижаются с течением времени, протокол PoS имеет достаточно времени для перехода в центр внимания.

Например, в архитектору Peercoin Кинга и Надаля для блоков PoS вводится новый процесс минтинга в дополнение к Bitcoin PoW, а блоки разделяются на два разных типа: блоки PoW и блоки PoS. PoS в новом типе блоков - это специальная транзакция, называемая *coinstake* (названная в честь специальной тразнакции в Bitcoin — *coinbase*). В транзакции *coinstake* владелец блока оплачивает себя, тем самым потребляя свои монеты, получая при этом привилегию генерации блока для сети и минтинга Для PoS. Первый ввод монет называется *kernel* и необходим для выполнения определенного протокола хэширования, что делает генерацию POS-блоков стохастическим процессом, подобным PoW-блокам. Однако существенное различие заключается в том, что операция хэширования выполняется в ограниченном пространстве поиска (точнее, один хэш на неизрасходованный кошелек в секунду) вместо неограниченного пространства поиска, как в PoW. Таким образом достигается отсутствие значительного потребления энергии.

Цель хэша, которую ядро ставки должно выполнить, - это цель на единицу возраста монеты (*день монеты*), потребляемая в ядре (в отличие от цели PoW Bitcoin , которая является фиксированным целевым значением, применяемым к каждому узлу). Таким образом, чем больше монет потребляется в ядре, тем легче достичь целевого протокола хэша. Например, если у Боба есть кошелек-вывод, который накопил 100 монетных лет и ожидает, что он сгенерирует ядро за 2 дня, то Алиса может примерно ожидать, что ее 200-летний кошелек-вывод будет генерировать ядро за 1 день. В дизайне Peercoin как цель хэша PoW, так

и цель хэша PoS регулируются непрерывно, а не двухнедельным интервалом регулировки Bitcoin, чтобы избежать внезапного скачка скорости генерации сети.

В таблице 2 представлены результаты применения этой процедуры к нескольким разным PoS и криптовалютам с использованием гибридной системы.

Валюта	**NXT**	**Reddcoin**	**Peercoin**	**BlackCoin**	**NovaCoin**
Время блока	60 s	min	10 min	60 s	10 min
Hashrate (hash/s)	1,0	$1{,}30\times10^{+10}$	$3{,}32\times10^{+16}$	$1{,}06\times10^{+14}$	$4{,}42\times10^{+11}$
C_μ	0,122	$5{,}26\times10^{-11}$	$3{,}29\times10^{-18}$	$8{,}52\times10^{-15}$	$1{,}86\times10^{-13}$

Таблица 2. Статистическая мера сложности C_μ рассчитана для нескольких PoS и гибридных криптовалют.

Заметно, как высокая сложность и, следовательно, необходимая более высокая хэш-скорость PoW-части протокола способствует более низкому значению меры сложности по сравнению с протоколом NXT PoS по причинам, рассмотренным выше. Поскольку интервал времени между форжингом блоков в Nxt составляет около 60 секунд, вероятность выбора любого узла значительно выше, чем у других валют.

В отличие от PoW, согласование на основе подтверждения доли не является объективным, новые пользователи не могут точно определить состояние PoS-системы на основе исключительно правил протокола и данных, полученных от узлов системы. Для того чтобы исключить возможность длинных форков блокчейна, PoS система должна быть слабо субъективной, дополняя правила протокола безопасностью, обеспечиваемой социально. Социальный компонент систем, использующий подтверждение доли, ослабляет их децентрализацию и математическую обоснованность.

Выводы:

Данное исследование показывает, что что статистическая сложность может быть использована в качестве эффективного инструмента анализа для оценки жизнеспособности предложенных высокопроизводительных сетевых криптографических методов на основе имеющихся количественных данных.

В то время, как консенсус PoW в соответствии со стандартом PoW в целом показал, что он очень не сложный, протокол консенсуса Nxt PoS показывает чрезвычайно высокую степень сложности.

Эта функция нежелательна, поскольку она вводит излишнюю сложность в то, что должно быть простой вычислительной системой, и поэтому предлагаемые методы PoS кажутся более запутанными и сложными, чтобы быть глобально масштабируемыми, применимыми и устойчивыми в качестве модели децентрализованных сетевых вычислений. Эта высокая сложность, похоже, не вытекает из концепции PoS как таковой, а из более мелкой конкуренции среди фальсификаторов, полученной из значительно меньшего количества испытаний в секунду, производимых в Nxt для выбора следующего фальсификатора.

При внесение изменений и при правильном построение нового алгоритма на базе концепции PoW, мы можем добиться значительных результатов в получение нового вида алгоритма для масштабируемой системы. Который позволит майнерам использовать значительно меньшее количество

электроэнергии и создаст полную децентрализацию и независимость от крупных пулов.

6. Tkeycoin и квантовый компьютер

6.1. Общие положения

Уже давно ставится под сомнение, какое влияние квантовые компьютеры окажут на Bitcoin и криптовалюты в целом. Мы проанализируем три основных направления, на которые могут повлиять квантовые компьютеры: майнинг, безопасность и форки.

Наши специалисты находят, что в ближайшей перспективе влияние квантовых компьютеров оказывается довольно малым для всех трех направлений. Воздействие квантовых компьютеров потребует значительно большего числа кубитов и прорывов в квантовых алгоритмах для обращения существующих хэш-функций.

Скачок цен на Bitcoin в конце 2017 года привлек внимание общественности к криптовалютам и их месту в будущем. Многие из привлекательных особенностей криптовалюты отсутствие третьих лиц, контролирующие валюты, обеспечивая такие преимущества, как более низкие операционные издержки, скорость и безопасности. В то время, как его использование в качестве средства платежа остается достаточно ограниченным. Рост стоимости Bitcoin помог привлечь к нему большой интерес, что сторонники Bitcoin предполагают, является первым шагом к общественному признанию.

В то же время в последние годы наблюдается растущая волна оптимизма по поводу квантовых вычислений. Сейчас не редкость слышать комментарии о том, что технология квантовых вычислений достигла такого уровня, когда масштабируемость была в пределах досягаемости, это было что-то неслыханное даже пять лет назад. Хорошо известно, что сердцем Bitcoin является криптография, и квантовые компьютеры особенно хороши в таких сложных задачах, как поиск и взлом кода. Это открывает множество вопросов относительно того, как квантовые компьютеры повлияют на Bitcoin.

Можно ли использовать квантовый компьютер для добычи Bitcoin? Могут ли квантовые компьютеры скомпрометировать систему Bitcoin? Можно ли использовать квантовый компьютер в чужих руках, чтобы украсть Bitcoin? В течение некоторого времени это было проблемой в сообществе Bitcoin, хотя это теоретическая проблема на данном этом этапе. В этой разделе мы раскрываем некоторые из этих проблем и точно видим, что подразумевает последствия постквантового компьютерного мира для Bitcoin и других криптовалют.

6.2. Майнинг

Одной из технологий, на которых основан Bitcoin, является SHA-256, криптографическая хеширующая функция, которая превращает произвольные входные данные в 256-битную строку («хэш»). Это односторонняя функция, так что легко найти хэш из ввода, но не наоборот. Bitcoin mining состоит из задачи поиска ввода («nonce») в сочетании с информацией о последнем блоке, который генерирует хэш, который меньше целевого значения *T*, максимальное число, приемлемое для того, чтобы считаться допустимым Bitcoin хэш. Целевое значение постоянно корректируется так, что среднее время между блоками составляет 10 минут. (на момент исследования цель приблизительно, равна $T=8.9x10^{11}$ гораздо меньше, чем $2^{256} = 1{,}2x10^{77}$.

Если бы можно было найти квантовый алгоритм для эффективного преобразования SHA-256, то мы могли бы действительно легко разбить Bitcoin

или любую другую криптовалюту. Однако ценность Bitcoin исходит из трудности нахождения таких решений, что дает ему «доказательство работы» (PoW). В настоящее время считается, что нет эффективного алгоритма, классического или квантового, который может инвертировать SHA-256. Следовательно, единственный способ - поиск грубой силы, который классически означает попытки ввода разных входов до тех пор, пока не будет найдено удовлетворительное решение.

В квантовой механике мы имеем поиск Гровера, который, кажется, является идеальным решением этой проблемы и имеет квадратичное квантовое ускорение. Давайте посмотрим, насколько хорошо эта стратегия работает, сравнивая ее с майнингом на классическом компьютере. Классически вероятность успеха добычи блока с догадками дается формулой $T_{rt} / 2^{256}$, где r это hashrate (количество догадок в секунду), и t время в секундах. Для квантового майнера, выполняющего алгоритм Гровера, вероятность успеха равна $\sin^2 (2r_q t \sqrt{T/2^{256}})$, где r_q это число итераций Гровера в секунду, которое мы можем назвать «квантовой скоростью хэша».

Теперь между классическим и квантовым майнером существует другая динамика, потому что Bitcoin предназначен для поиска нового блока в среднем каждые 10 минут (=600 секунд), и, следовательно, характер проблемы поиска меняется в это время. Для того, чтобы процедура Гровера давала высокую вероятность успеха, квантовый майнер должен запустить свой алгоритм на некоторое время t до изменения проблемы, а затем произвести измерение. Между тем, классический майнер в это время переберал, как можно больше nonces. Таким образом, квантовый майнер надеется, что ни один из классических майнеров не нашел решения еще во время эволюции Гровера. Поскольку интервал между блоками следует за экспоненциальным распределением, вероятность того, что блок все еще доступен для добычи, определяется $e^{-t/600}$. Предполагая постоянную стоимость запуска квантового компьютера в течение заданного времени, прибыльность квантового Bitcoin-майнинга тогда

$$Re^{-t/600} \sin^2 (2r_q t \sqrt{T/2^{256}}) - Ct$$

где, R является вознаграждением (на момент исследования вознаграждение = 12,5 Bitcoin плюс транзакционные сборы), а C - стоимость запуска квантового компьютера. Оценим теперь некоторые правдоподобные числа, чтобы выяснить, выгодна ли квантовая добыча Bitcoin. Мы предположим, что квантовый компьютер стоит так же, как и классический компьютер, и используем сегодняшнюю цену биткоина, вознаграждение за блок и сложность майнинга. Мы оцениваем, что квантовая добыча Bitcoin становится прибыльной при скорости квантового хэша 48 *kilo-hashes/s*. По сравнению с нынешним лучшим классическим оборудованием для майнинга Bitcoin с хэш-скоростью 125 *kilo-hashes/s,* эти цифры могут показаться многообещающими, но мы должны иметь в виду, что классические майнеры Bitcoin могут достичь огромных ставок хеша, потому что алгоритм случайного добычи может быть довольно легко распараллелен. Проблема в том, что квантовое преимущество не превышает $\sqrt{T/2^{256}}$ сколько бы ни было кубитов. Таким образом, хотя существует квантовое преимущество, оно не является достаточно непреодолимым, чтобы классическая распараллеливание не могла победить его. Для квантового компьютера с меньшей скоростью хэширования, чем минимально выгодные *48 kilo-hashes/s,* тогда нужно было бы прибегнуть к классическому распараллеливанию квантовых компьютеров.

Например, если квантовая хэш-скорость составляет 3 *kilo-hashes/s*, то потребуется 1300 квантовых компьютеров, чтобы быть наравне с классическим лучшим оборудованием для майнинга, которое можно купить сегодня. Таким образом, для того, чтобы квантовая добыча была прибыльной, требовались бы

довольно быстрые квантовые hashrates и / или значительно более значительное квантовое ускорение. Это может произойти в будущем, но пока классический майнинг PoW является сложным.

6.3. Безопасность

Чтобы гарантировать, что Bitcoin тратится только их законными владельцами, используется алгоритм цифровой подписи эллиптической кривой (ECDSA). Он основан на криптографии с открытым ключом, где владельцы Bitcoin могут подписывать транзакции, используя свой приватный ключ (private key), а другие могут убедиться, что он подлинный, используя свой публичный ключ. Криптография с эллиптическими кривыми уязвима для квантовых вычислений, так как алгоритм Шора может быть легко модифицирован для расшифровки сообщений, отправляемых с помощью эллиптических кривых, т.е. квантовый компьютер можно было бы использовать, чтобы найти закрытый ключ из открытого ключа. Это, по-видимому,подвергает уязвимость, но на самом деле в Bitcoin встроены несколько защит, которые предотвращают это. Во-первых, открытые ключи не отображаются по вашему адресу. Протокол Bitcoin генерирует адреса, помещая открытый ключ через SHA-256, а затем через RIPEMD-160. Поскольку открытый ключ обнаруживается только при расходовании Bitcoin, он становится уязвимым для атаки квантового компьютера только после того, как открытый ключ обнаружен в транзакции. Эта ситуация легко устраняется путем создания нового адреса после каждой транзакции (как и в случае наилучшей практики). Как только квантовые компьютеры станут обычным явлением, большинство Bitcoin-клиентов переключатся на автоматическую генерацию ключей после каждой транзакции. Это может снизить удобство некоторых приложений. Например, вы не сможете распечатать Bitcoin адрес в QR-код и использовать его постоянно в качестве кассового аппарата. К сожалению (как мы скоро увидим)это исправление является временным.

Другая возможность заключается в том, что как только открытый ключ обнаружен в ожидающей транзакции, злоумышленник Ева с квантовым компьютером может украсть Bitcoin до завершения транзакции. В принципе, у Евы есть только 10 минут, чтобы найти секретный ключ до завершения транзакции. На практике Bitcoin-транзакции часто проводятся в ожидающем пуле (“mem-pool”) в течение часа или больше. Для 256 бит ECDSA требуется около 1500 кубитов и $6x10^9$ добавлений одного кубита (каждое добавление одного кубита занимает 9 квантовых ворот). Таким образом, для выполнения такого типа атаки в течение часа квантовому компьютеру необходимо выполнить скорость операций ворот около 660 МГц. В недавних исследованиях Roetteler, указано, что требуется 2330 кубитов и 1.26 * 10^11 требуются операции ворот Toffoli (Примечание: предполагается, что ворота без Toffoli занимают незначительное время в этой работе). По этой оценке, несмотря на то, что требуется больше кубитов, квантовому компьютеру нужно будет работать только на 350 МГц, чтобы снять атаку. В любом случае требования к количеству кубитов и скорости делают эту атаку невозможной для ранних поколений квантовых компьютеров.

Предполагая, что вы можете сломать оба SHA-256 и RIPEMD-160, но тогда с существующим адресом было бы легко взять деньги из чужих резервов. Однако обратите внимание, что эта проблема по существу такая же, как проблема поиска хеш-коллизий с целью добавления блоков в блокчейн (т. е. майнинг).

Теория гласит, что пока вычислительная мощность, необходимая для кражи Bitcoin, намного выше, чем вычислительная мощность, необходимая для добычи Bitcoin, любой, кто мог бы украсть, вместо этого будет его добывать. Квантовые вычисления не очень сильно меняют эту логику.

Выполнение атаки столкновения хэша на классическом компьютере требует исчерпывающей генерации кошельков, в то время как в пост квантовом мире необработанные входы в хэш-функцию можно проверить, а затем распознать закрытые ключи. Таким образом, получается постоянный коэффициент ускорения при столкновении атак. Помимо этого постоянного фактора сохраняется существующая безопасность Bitcoin против атак на столкновение с хешем. Может ли существовать какая-то другая уязвимость? Хотя нет никаких доказательств того, что другие уязвимости не существуют, в настоящее время нет никаких оснований полагать, что они есть. Единственная гарантия, которую мы имеем, что Bitcoin существует уже много лет, и никто его не взламывает, несмотря на огромные финансовые стимулы для этого. Хотя мы не исключаем возможности изобретения новых квантовых алгоритмов с целью взлома или майнинга, по меньшей мере, это кажется весьма нетривиальным.

6.4. Форки

В мире с квантовыми компьютерами безопасность Bitcoin целиком лежала бы в хэш-функциях SHA-256 и RIPEMD-160 и в медленных тактовых циклах ранних квантовых компьютеров. (например, SHA-1 и MD5). По этой причине многие люди увидели бы здравый смысл в изменении системы подписи открытого ключа Bitcoin на то, что не уязвимо для квантового компьютера.

Существуют классические криптографические примитивы, которые, как известно, не уязвимы для квантовых компьютеров. Сказав это, в целом нет доказательств того, что любой из пост-квантовых методов действительно безопасен против квантового компьютера. Например, криптография на основе задачи кратчайшего вектора (SVP) в настоящее время считается безопасной. Однако SVP (как и все решетчатые задачи) имеет много симметрии и периодичности. Это делает его проблемой, которая хорошо подходит для квантовых вычислений. Можно предположить, что квантовый алгоритм для решения SVP существует, но просто еще не найден. В какой-то момент может начаться гонка вооружений между открытием квантовых алгоритмов и изобретением неортодоксальных способов классической криптографии.

Bitcoin может быть пойман в середине этой гонки вооружений. Результатом может стать ряд жестких форков (совместные изменения протокола), поскольку протокол Bitcoin старается идти в ногу с последними разработками. Как мы видели в недавнем форке Bitcoin Cash, такие изменения в протоколе всегда противоречивы и опасны для криптовалюты. Однако, в отличие от спорных моментов обсуждения такого размера блока, никто не находится под иллюзией, что ECDSA является неотъемлемой частью цели или архитектуры Bitcoin. Замена другого метода не должна влиять на масштабирование или децентрализацию. В этом смысле вполне вероятно, что квантовые связанные форки будут менее спорными, чем недавние «точечные» изменения, и будут широко приняты. Отметим, что Виталик Бутерин (изобретатель Ethereum) предложил использовать подписи Лампорта, которые основаны на хэш-функциях и, таким образом, квантово устойчивы. В этом смысле вопрос квантовой безопасности криптовалют уже сегодня начинает восприниматься всерьез.

6.5. Выводы:

Мы обнаружили, что в ближайшем будущем влияние квантовых компьютеров будет, скорее всего, довольно небольшим для Bitcoin. Для квантового майнинга квантовое преимущество ограничено, поэтому потребуются очень быстрые скорости квантового хеширования. В целях безопасности существует уязвимость для отложенных транзакций из-за использования криптографии с эллиптическими кривыми, которая может быть нарушена вариантом алгоритма Шора. Однако, из-за временных ограничений во времени ожидания

транзакции, которое составляет около 10 минут, это создает довольно жесткие требования к возможностям квантового компьютера. Наконец, простая возможность существования квантового компьютера может дестабилизировать сам Bitcoin с помощью серии форков. Однако мы полагаем, что такие изменения в протоколе, вероятно, будут менее спорными, поскольку они не являются фундаментальным изменением в его архитектуре.

Заглядывая в будущее, можно представить себе гонку квантовых вооружений между экзотическими формами криптографии и квантовыми алгоритмами их разрушения. В таком сценарии разработка квантовых версий криптовалюты, основанных на квантовой механике, станет реальной необходимостью. В будущем, когда квантовые технологии станут обычным явлением, будет иметь смысл заново выпустисть новый блокчейн протокол и транзакционную цепочку методами квантовой криптографии. В зависимости от того, какой будет будущая квантовая вычислительная инфраструктура.

До сих пор неясно, будет ли среднестатистический пользователь иметь квантовый компьютер или же квантовые компьютеры будут существовать как облачный сервис, а пользовательские машины останутся классическими. Возможно, в будущем среднестатистический пользователь будет иметь квантовую связь, но полноценные квантовые вычисления будут выполняться на сервере.

7. Цифровые подписи

7.1. Общие положения

Выбирая цифровую подпись только с одной схемой подписей, Вы рискуете в определенный моменты быть взломанными, также некоторые организации не смогут использовать Ваш протокол из-за государственных стандартов.

Поэтому, необходимо создать гибкое решение отвечающее двум параметрам: безопасности и масштабируемости. Мы понимаем, что криптовалюты физически не могут использовать абсолютно все схемы и алгоритмы шифрования. Необходимо положиться на опыт крупных компаний и исследований новых концепций.

Разработка будет вестись с учетом поддержки в будущем других криптографических схем, которые можно будет добавлять по средствам софтфорка. Данные изменения будут внедрятся в моменты крупных обновлений.

Анализ, исследования и предупреждения National Security Agency (NSA) доказывают, что многие из используемых сегодня алгоритмов должны быть модернизированы. Наш выбор Curve25519 и Ed25519.

Curve25519 - это эллиптическая кривая и набор параметров к ней подобранных таким образом, чтобы обеспечить более высокое быстродействие (в среднем, 20-25%) и избавиться от некоторых проблем с безопасностью у традиционного ECDH.

Curve25519 используется, как обмен ключами по умолчанию в OpenSSH, I2p, Tor, Tox и даже в IOS, WhatsApp, Viber, OpenBSD, OpenSSH в протоколах ZCASH, TOX, библиотеках Libsodium, OpenSSL, LibreSSL, NaCl, NSS, Crypto ++

На веб-сайте SafeCurves представлены оценки безопасности различных конкретных кривых. «Безопасный» в следующей таблице означает, что кривая удовлетворяет всем требованиям SafeCurves. Смотрите описание Curve25519 в таблице - http://safecurves.cr.yp.to/

Работа представленная на сайте SafeCurves была поддержана Национальным научным фондом США под грантом 1018836 и Нидерландской организацией научных исследований (NWO) в рамках гранта 639.073.005.

7.2. Ed25519: высокоскоростные высоконадежные сигнатуры

Подписи Ed25519 являются сигналами эллиптической кривой, тщательно спроектированными на нескольких уровнях архитектуры и реализации для достижения очень высоких скоростей без ущерба для безопасности.

Эта система имеет цель безопасности 2 ^ 128 его разрыв имеет схожую сложность с нарушением NIST P-256, RSA с ~ 3000-битными ключами, сильными 128-битными блочными шифрами и т. д. Наилучшие атаки, которые, как известно, в среднем стоят более 2 ^ 140 бит операций и деградируют квадратично в результате успеха вероятность того, что количество операций бит прекратится.

Эллиптическая кривая Диффи Хеллман, эллиптическая кривая генерации подписи и эллиптическая кривая Подтверждение подписи полагается на скалярные умножения, они обычно реализуются как различные типы скалярного умножения как по соображениям безопасности, так и по эффективности.

7.3. Анализ Curve25519

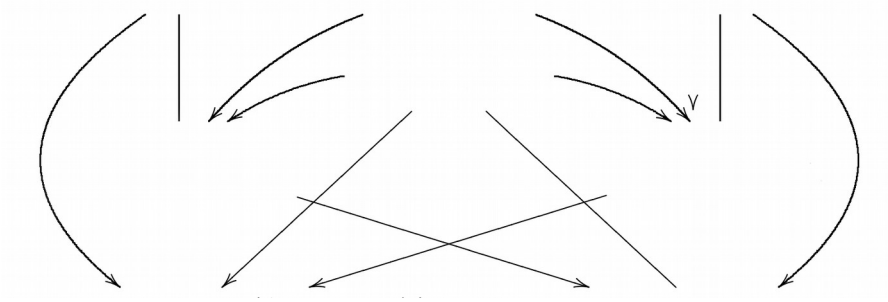
Данная глава посвящена рассмотрению и анализу Curve25519 — это эллиптическая кривая функции Диффи-Хеллмана, которая совместима со многими криптографическими приложениями. Здесь Curve25519 применяется для получения новых скоростных рекордов высокобезопасных вычислений функции Диффи-Хеллмана.

Высокоуровневый вид Curve25519: Каждый пользователь Curve25519 обладает 32 - байтным секретным ключом и 32-байтным открытым ключом. Каждое объединение двух пользователей Curve25519 обладает 32-байтным совместно используемым секретным ключом, который используется для аутентификации и шифрования сообщений между двумя пользователями.

Среднеуровневый вид: На рисунке ниже, показан поток данных из секретных ключей через открытые ключи к совместно используемому секретному ключу.

Секретный ключ Алисы *a* Секретный ключ Боба *b*

Открытая строка 9



Общий секретный ключ {Алисы, Боба} = Общий секретный ключ {Боба, Алисы}
Curve25519 (*a*, Curve25519 (*b*, 9)) Curve25519 (*b*, Curve25519 (*a*, 9))

Хэш общего секретного ключа Curve25519 (*a*, Curve25519 (*b*, 9)) используется в качестве ключа для системы аутентификации с секретным ключом (для аутентификации сообщений) или в качестве ключа для системы аутентифицированного шифрования с секретным ключом (для одновременной аутентификации и шифрования сообщений).

Низкоуровневый вид: Функция Curve25519 представляет собой скаляр Fpв сжатой форме координаты x умноженный на E ($\mathbf{F}p^2$), где p - это простое число $2^{255} - 19$ и E - это эллиптическая кривая $y^2 = x^3 + 486662x^2 + x$.

7.4. Предполагаемый уровень безопасности Curve25519

Прерывание функции Curve25519 (например, вычисление совместно используемого секретного ключа из двух открытых ключей) считается крайне сложным. Каждая известная атака является более дорогостоящей, чем перебор на типичном 128-битным шифром с секретным ключом шифрования и расшифрования. Над решением общей проблемы дискретных логарифмов эллиптической кривой трудились в течение двух десятилетий, но успехи были незначительными. Обобщенные алгоритмы дискретных логарифмов разбивают простые группы, которые не достаточно большие, но размер простой группы, рассматриваемой нами, превышает 2^{252}. Эллиптические кривые с некоторыми специальными алгебраическими структурами можно Information theory of complex systems разбить намного быстрее с применением необобщенных алгоритмов, но E (Fp^2) не обладает этими структурами.

Спецификация

ВНИМАНИЕ: Если вы не знаете о кольцах, полях и эллиптических кривых, Вы должны ознакомиться со значениями данных понятий в Wikipedia и доказательством Теоремы, которая привидена ниже:

Теорема №1

Пусть, p - это простое число при p ≥ 5. Пусть, A - это целочисленная переменна так, что $A^2 - 4$ не является квадратом по модулю p. Определим E какэллиптическую кривую $y^2 = x^3 + Ax^2 + x$ над полем $\mathbf{F}p$*. Определим X_0: E(*$\mathbf{F}p^2$*) →* $\mathbf{F}_p{}^2$*: $X_0(\infty) = 0$; $X_0(x, y) = x$. Пусть, n - это целочисленная переменная. Пусть, q является элементом* $\mathbf{F}_p$*. Тогда существует только s ∈* $\mathbf{F}_p$*, так что $X_0(nQ) = s$ для всех Q ∈ E(* $\mathbf{F}p^2$ *), поэтому $X_0(Q) = q$.*

В частности, пусть $p = 2^{255}-19$. В Теореме В. 1 p - это простое число. Определим $\mathbf{F}_p$ как простое поле $\mathbf{Z}/p = \mathbf{Z}/(2^{255}-19)$. Обратите внимание, что число 2 не является квадратом $\mathbf{F}_p$; определим $\mathbf{F}_p{}^2$ как поле ($\mathbf{Z}/(2^{255} - 19)$) [√2]. Допустим, $A = 486662$. Обратите внимание, что значение $486662^2 - 4$ не является квадратом в $\mathbf{F}_p$. Определим E как эллиптическую кривую $y^2 = x^3 + Ax^2 + x$ над $\mathbf{F}_p$.Определим функцию $X_0 : E(\mathbf{F}_p{}^2) \to \mathbf{F}_p{}^2$ как: $X_0(\infty) = 0$; $X_0(x, y) = x$. Сейчас уже можно сказать, что заданная заданная $n \in 2^{254} + 8$ {0,1, 2, 3, . . . , $2^{251} - 1$ и $q \in$} $\mathbf{F}_p$, функция Curve25519 производит значение s Теореме 1. Однако, чтобы сопоставить криптографическую реальность и обнаружить типы ошибок проектирования, описанных Менезесом, я определяю вводы и выводы Curve25519 как байтовые последовательности.

По определению, набором байтов является {0, 1,..., 255}. Кодирование байта как последовательности битов не имеет отношения к теме нашей работы. Запишем $s \to \underline{s}$ для стандартной биекции из значения {0,1,...$2^{256} - 1$} для набора {0, 1,..., 255}^{32} 32- байтовых строк: то есть, для каждого целого числа $s \in$ {0, 1,..., $2^{256} - 1$}, $\mathbf{O}\underline{s} = (s \bmod 256, \lfloor s/256 \rfloor \bmod 256, ..., \lfloor s/256^{31} \rfloor \bmod 256)$.

Набором **открытых ключей** Curve25519, по определению, является {0, 1,..., 255}^{32}; т. е., $\{\underline{q} : q \in \{0, 1, ..., 2^{256} - 1\}\}$. Набором **секретных ключей** Curve25519,

поопределению, является {0, 8, 16, 24,... , 248} × {0, 1,..., 255}^{30} × {64, 65, 66,... , 127}; иными словами {$\underline{n}$: $n \in 2^{254} +$ {0,1, 2, 3,... , $2^{251} - 1$}}.

Теперь определим Curve25519: {секретные ключи Curve25519}×{открытые ключи Curve25519} → открытые ключи Curve25519}. Пусть, $q \in$ {0, 1,..., $2^{256} - 1$ и $n \in 2^{254} + 8$ {0, 1, 2, 3, . . , $2^{251} - 1$}. Согласно Теореме 1, существует уникальное число $s \in$ {0, 1, 2, ..., $2^{255} - 20$} при котором $s = X_0(nQ)$ $Q \in E(\mathbf{F}_p^2)$, *так что* $X_0(Q) = q \bmod 2^{255} - 19$. Наконец, Curve25519($\underline{n}$, $\underline{q}$) определяется как $\underline{s}$. Обратите внимание, что функция Curve25519 не является сюръективной: в частности, ее финальное значение выходного бита всегда равно 0, и его не нужно передавать.

8. Двойное расходование

Мы исправляем двойное расходование, приведенный в основополагающей статье Bitcoin Накамото, и даем формулу закрытой формы для вероятности успеха двойной атаки с использованием Регуляризованной неполной бета-функции. Приведем доказательство экспоненциального спада на число подтверждений, часто приводимых в литературе, и найдем асимптотическую формулу. Необходимо большее количество подтверждений по сравнению с данными Накамото. Мы также вычисляем вероятность, обусловленную известным временем проверки блоков. Это обеспечивает более точный анализ риска, чем классический.

Основной прорыв - это решение проблемы двойных расходов. До этого открытия, никто не знал, как избежать двойных расходов на электронную валютную единицу без надзора со стороны центрального органа. Это сделало Bitcoin - первой формой электронной одноранговой валюты.

Двойная атака может быть предпринята только с существенной долей hashrate, используемый в доказательстве работы сети Bitcoin. Злоумышленники начнут двойную гонку против остальной сети, чтобы заменить последние блоки блокчейна, тайно добывая альтернативную цепочку блоков. Последний раздел вычисляет вероятность злоумышленников догонать сеть.

Однако анализ Накамото не точен, так как он делает упрощающее предположение, что честные майнеры проверяют блоки с ожидаемой скоростью. Мы приведем правильный анализ и дадим формулу замкнутой формы для точной вероятности.

8.1. Анализ

Теорема 8.1.1.

Пусть $0 < q < 1/2$, относительная хэш-мощность группы из нападавших, а $p = 1-q$, соответственно честных майнеров. После того, как z блоков были проверены честными майнерами, вероятность успеха атакующих, есть:

$$P(z) = I_{4pq}(z, ½)$$

где $I_x(a, b)$ - *Регуляризованная неполная бета-функция*

$$i_x(a,b)=\frac{\Gamma(a+b)}{\Gamma(a)+\Gamma(b)}=\int_0^x t^{a-1}(1-t)^{b-1}\,dt\,.$$

В общем случае для $z \geq 2$ эти вероятности $P(z)$ больше, чем полученные $P_{SN}(z)$ Накамото. С точки зрения безопасности Bitcoin, это показывает, что необходимо большее время подтверждения z, по сравнению с теми z_{SN} данными, которые предоставлены Накамото. В частности, это происходит,

когда важна доля *hashrate q* атакующих. В следующей таблице показано число *z* ожидания по сравнению с z_{SN} *данными*, которые дал Накамото для атакующего *hashrate* 10% (или *q* = 0,1) и вероятность успеха атакующих составляет менее 0,1%.

q	0.10	0.15	0.20	0.25	0.30	0.15	0.40	0.15
z	6	9	13	20	32	58	133	539
z_{SN}	5	8	11	15	24	41	81	340

Накамото утверждает, что вероятность *P*(*z*) экспоненциально стремится к 0 при *z*. Этот результат интуитивно ожидаем и цитируется в целом, но мы не смогли найти доказательство в литературе. Мы приводим здесь строгое доказательство этого результата. Точнее, мы дадим точные асимптотики для P_{sn} (*z*) и *P*(*z*), которые показывают экспоненциальный спад.

Теорема 8.1.2.

При $z \to +\infty$ имеем, при $s = 4_{pq} < 1$,

$$P(z) \sim \frac{s^z}{\sqrt{\pi(1-s)z}}.$$

причем $\lambda = q/p$, а $c(\lambda) = \lambda - 1 - \log \lambda > 0$,

$$P_{SN}(z) \sim \frac{e^{-zc(\lambda)}}{2}$$

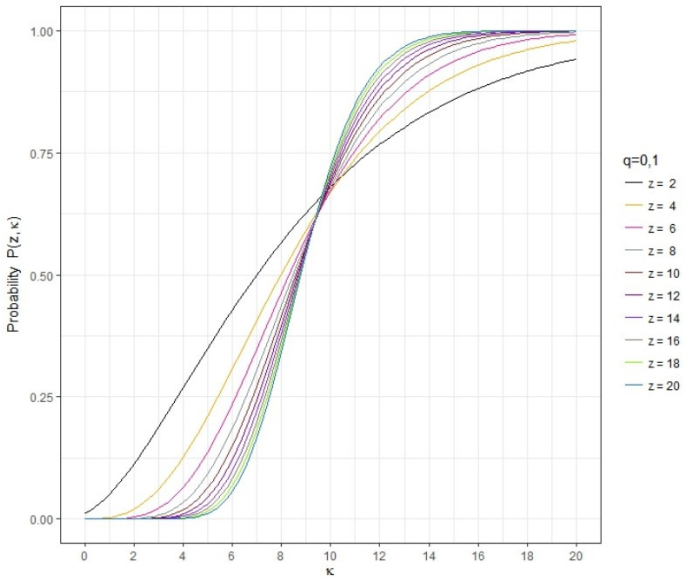
Мы можем проверить, что — log *s* > c (λ), что означает, что $P_{sn}(z) \prec$ P (*z*) для *z* большой.

8.2. Более точный анализ рисков.

Мы анализируем новый параметр риска двойной траты. Вероятность успеха атакующих увеличивается со временем τ_1, которое требуется для проверки транзакций *z*, так как у них есть больше времени, чтобы тайно добывать альтернативную цепочку блоков. С другой стороны, задача злоумышленников сложнее, если проверки происходят быстрее, чем ожидаемое время. Значение τ_1 известно, поэтому то, что действительно актуально, - это условная вероятность, предполагающая τ_1. Введем безразмерный параметр к, который измеряет отклонение от среднего времени:

$$k = \frac{\tau_1}{z\, t_0}$$

где t_0 среднее время валидации блока честными майнерами ($t_0 = \tau_0 / p$, где τ_0 = 10 мин для сети Bitcoin).



Вероятность успеха как функция к *при q = 0.1*

Изучается вероятность P (*z,* к) успеха атакующих. Мы можем восстановить предыдущие вероятности с помощью P (*z,* к), 0 < к < 1.

Теорема 8.2.1.

У нас есть,

$$P_{sn}(z) = P_{sn}(z,1)$$

и

$$P(z) = \int^{+\infty}_{0} P(z,k)\, dp_z(k),$$

с функцией плотности

$$dp_z(k) = \frac{z^z}{(z-1)!}\, k^{z-1} e^{-zk}\, dk .$$

Приведем форумулу к замкнутой формы для P (*z,k*).

Теорема 8.2.2.

У нас есть,

$$P(z,k) = 1 - Q(z, kzq/p) + \left(\frac{q}{p}\right)^z e^{kz\frac{p-q}{p}} Q(z,kz) .$$

Мы также находим асимптотику при $z \to +\infty$ при различных значениях к.

Теорема 8.2.3.

Мы имеем для $z \to +\infty$,

(1) *Для 0* < к < *1,*

$$P(z,k) \sim \frac{1}{1-k\lambda} \frac{1}{\sqrt{2\pi z}} e^{-zc(k\lambda)}$$

(2) *Для* к = *1,*

$$P(z,k) = P_{SN}(z) \sim \frac{1}{2} e^{-zc(\lambda)}$$

(3) *Для 1* < к < *p/q,*

$$P(z,k) \sim \frac{k(1-\lambda)}{(k-1)(1-k\lambda)} \frac{1}{\sqrt{2\pi z}} e^{-zc(k\lambda)}$$

(4) *Для* к = *p/q, P (z, p/q)* → *1|2 и*

$$P(z,p/q) - 1/2 \sim \frac{1}{2\pi z}\left(\frac{1}{3} + \frac{q}{p-q}\right)$$

(5) *Для p/q*, $P(z, k) \to 1$ и

$$1-P(z,k)\sim \frac{k(1-\lambda)}{(k-1)(k\lambda-1)}\frac{1}{\sqrt{2\pi z}}e^{-zc(k\lambda)}$$

Используя аргумент вогнутости, мы покажем, что $P(1) \le P_{SN}(1)$, но в целом для $z \ge z_0$ мы имеем $P_{SN}(z) \le P(z)$. Мы делаем вычислить явный, не резкий, значение Z_0 для которых это неравенство справедливо:

Теорема 8.2.4.

Пусть $z \in \mathbb{N}$. Достаточным условием наличия $P_{SN}(z) \le P(z)$ является $z \ge z_0$ с $z_0 = \lceil z^*_0 \rceil$ -наименьшее целое число, большее или равное:

$$z^*_0 = \quad max\left(\frac{2}{\pi(1-q/p)^2}, \frac{1}{2\sqrt{2}}-\frac{(1+\frac{1}{\sqrt{2}})}{2}\frac{\log(\frac{2\varphi(p)}{\pi})}{\varphi(p)}\right)$$

$$\text{где} \quad \varphi(p)=\frac{q}{p}-1-\log(\frac{q}{p})-\log(\frac{1}{4pq})>0.$$

Мы также предоставляем двойные таблицы входа $P(z,$ к) для разных значений ($q,$ к) для $z = 3$ и $z = 6$.

Для полного набора таблиц при $z = 1, 2, \ldots 9$.

8.3. Математика майнинга

Процесс bitcoin-майнинга состоит из вычисления хэшей заголовков блоков, изменяющих *nonce*, чтобы найти хэш ниже предопределенного порога, то есть *трудности*. При каждом новом хэше работа начинается с нуля, поэтому случайная величина $\boldsymbol{T}$, измеряющая время, необходимое для майнинга блока, не запоминается, что означает, что для любого $t_1, t_2 > 0$

$\mathbb{P}[\boldsymbol{T} > t_1, t_2 \mid \boldsymbol{T} > t_2] = \mathbb{P}[\boldsymbol{T} > t_1]$

Поэтому, мы имеем

$\mathbb{P}[\boldsymbol{T} > t_1, t_2 \mid \boldsymbol{T} > t_2] = \mathbb{P}[\boldsymbol{T} > t_1+t_2 \mid \boldsymbol{T} > t_2]. \mathbb{P}[\boldsymbol{T} > t_2] = \mathbb{P}[\boldsymbol{T} > t_1]. \mathbb{P}[\boldsymbol{T} > t_2]$.

Это уравнение и аргумент непрерывности определяют экспоненциальную функцию и подразумевают, что $\boldsymbol{T}$ является экспоненциально распределенной случайной величиной:

$f_T(t) = \alpha e^{-\alpha t}$

для некоторого параметра $\alpha > 0$*, скорость добычи, с* $t_0 = 1/\alpha = \mathbb{E}[\boldsymbol{T}]$.

Если ($\boldsymbol{T}_1$,. . . , $\boldsymbol{T}_n$) - последовательность независимых одинаково распределенных экспоненциальных случайные величины (например, $\boldsymbol{T}_k$ -это время майнинга k-го блока), затем сумма

$$\boldsymbol{S}_n = \boldsymbol{T}_1 + \ldots + \boldsymbol{T}_n$$

является случайной величиной, следующей за гамма-плотностью с параметрами (n, α) (полученными сверткой экспоненциальной плотности):

$$f s_n(t) = \frac{\alpha^n}{(n-1)!} t^{n-1} e^{-\alpha t}$$

и кумулятивное распределение

$$F s_n(t) = \int_0^t f s_n(u)\,du = 1 - e^{-\alpha t} \sum_{k=0}^{n-1} \frac{(\alpha t)^k}{k!}.$$

Определим случайный процесс $\boldsymbol{N}(t)$ как количество добытых блоков за время t. Установка $S_0 = 0$, мы имеем

$$\boldsymbol{N}(t) = \#\{k \geq 1; \boldsymbol{S}_k \leq t\} = \max\{n \geq 0; \boldsymbol{S}n < t\}\ .$$

Так как $\boldsymbol{N}(t) = n$ эквивалентно $\boldsymbol{S}_n \leq t$, а $\boldsymbol{S}_n+1 > t$ получим

$$\mathbb{P}\,[\boldsymbol{N}(t) = n\,] = F_{sn}(t) - F_{sn+1}(t) = \frac{(\alpha t)^n}{n!} e^{-\alpha t}$$

это означает, что $\boldsymbol{N}(t)$ имеет распределение Пуассона с ожиданием αt.

8.4. Гонка майнеров

Рассмотрим ситуацию, где группа атакующих майнеров пытается провести двойную атаку. Группа злоумышленников имеет долю $0 < q < 1/2$ от общего *hashrate*, а остальные, честные майнеры, имеют фракции $p = 1 - q$. Таким образом, вероятность того, что злоумышленники найдут следующий блок является q, а вероятность для честных майнеров — это p. Накамото вычисляет вероятность того, что злоумышленники догонят z блоки, добытые честной группой майнеров. В общем, чтобы заменить добытую честными майнерами цепочку и добиться успеха в двойном растрате, атакующим необходимо добыть $z + 1$ блоков, т. е. добыть более длинную цепочку. В анализе предполагается, что мы не близки к обновлению трудности, которая остается постоянной (с*ложность регулируется каждые 2016 блоков*).

Вычисляя вероятность q_z атакующего, догоняющего, когда они отстают от z блоков за честными майнерами. Анализ правилен и похож на задачу о разорении игрока. Мы рассматриваем ее.

Лемма 3.1.

Пусть q_n вероятность события E_n, "*догоняющего из n блоков позади*". У нас есть,

$$q_n = (q/p)^n\ .$$

Доказательство,

Мы имеем $q_0 = 1$, $q_1 = q/p$, и Марковское свойство

$$q_n+_m = \mathbb{P}\,[E_{n+m}] = \mathbb{P}\,[E_n|E_m].\mathbb{P}[E_m] = \mathbb{P}\,[E_n].\mathbb{P}[E_m] = q_{n.m}\ ,$$

таким образом, получаем следующий результат $q_n = q^n{}_1$

Обратите внимание, что после того, как еще один блок был добыт, мы имеем для $n \geq 1$,

$$qn = qq_n - 1 + pq_{n+1}\ ,$$

и единственным решением этого повторения с $q_0 = 1$ и $q_n \to 0$ является $q_n = (q / p)^n$.

Рассмотрим случайные величины $\boldsymbol{T}$ и $\boldsymbol{S}_n$, - это связанные с группой честных значения, а $\boldsymbol{T}$ и $\boldsymbol{S}'n$ это нападавшие майнеры. Также рассмотрим случайный процесс Пуассона $\boldsymbol{N}(t)$, соответственно $\boldsymbol{N}'(t)$. Случайные величины $\boldsymbol{T}$ и $\boldsymbol{T}'$ явно независимы и имеют экспоненциальные распределения с параметрами α и α'.

$$\mathbb{P}[\boldsymbol{T} < \boldsymbol{T}'] = \frac{\alpha'}{\alpha+\alpha'},$$

$$p = \frac{\alpha}{\alpha+\alpha'},$$

$$q = \frac{\alpha'}{\alpha+\alpha'},$$

Более того, inf ($\boldsymbol{T}$,$\boldsymbol{T}'$) является экспоненциально распределенной случайной величиной с параметрами $\alpha + \alpha'$, которая представляет собой скорость майнинга всей сети, честных майнеров и атакующих вместе взятых. Протокол Bitcoin калибруется таким образом, что $\alpha + \alpha' = \tau_0$ с $\tau_0 = 10$ мин. Так мы имеем

$$\mathbb{E}[\boldsymbol{T}] = \frac{1}{\alpha} = \frac{\tau_0}{p},$$

$$\mathbb{E}[\boldsymbol{T}'] = \frac{1}{\alpha'} = \frac{\tau_0}{p}.$$

Эти результаты также можно получить следующим образом. Хэш-функция, используемая в проверке блока Bitcoin $h(x) = \mathrm{SHA256}(\mathrm{SHA256}(x))$. *Hashrate* -это количество хэшей в секунду, выполняемых майнерами. При стабильном режиме хэширования среднее время, необходимое для проверки блока сетью $\tau_0 = 10$ мин. Если установлена сложность $d \in (0, 2^{256} — 1)$, мы проверям блок при $h\ (BH) < d$, где заголовок блока *BH*. Псевдослучайный вывод SHA256 показывает, что нам нужно вычисление среднего числа $m = 2^{256}/d$ хэшей для поиска решения. Пусть h, будет *hashrates* честных майнеров, соответственно h', злоумышленников. Общая мощность сети h+h' и получим

$$p = \frac{h}{h+h'},$$

$$q = \frac{h'}{h+h'}.$$

Пусть t_0, среднее время, необходимое для проверки блока честными майнерами, t_0', соответственно злоумышленников.

$$(h + h')\,\tau_0 = m,$$
$$ht_0 = m,$$
$$h' t'_0 = m,$$

и из этого получаем, что τ_0 равно половине среднего гармонического t_0 и t_0'

$$\tau_0 = \frac{t_0 t'_0}{t_0 + t'_0},$$

а также,

$$p = \frac{t'_0}{t_0 + t'_0} = \frac{\tau_0}{t_0}$$

$$q = \frac{t'_0}{t_0 + t'_0} = \frac{\tau_0}{t'_0}$$

Возвращаясь к параметрам распределения Пуассона, мы получаем

$$\alpha = \frac{1}{t_0} = \frac{p}{\tau_0},$$

$$\alpha' = \frac{1}{t_0} = \frac{p}{\tau_0},$$

и мы восстанавливаем отношения

$$p = \frac{\alpha}{\alpha + \alpha'},$$

$$q = \frac{\alpha'}{\alpha + \alpha'}.$$

8.5. Анализ Накомото

Когда честные майнеры добывают *z*-й блок, атакующие добывают *k* блоков с вероятностью, вычисленной в следующем предожение 8.6.1. Если $k > z$, то цепь атакующих принимается, соответственно атака завершена успешна. В противном случае вероятность того, что они догонят сеть, равна $(q/p)^z$, как вычислено выше, поэтому вероятность *P* успеха атаки равна

$$P = \mathbb{P}[\boldsymbol{N}'(\boldsymbol{S}_z) \geq z] + \sum_{k=0}^{z-1} \mathbb{P}[\boldsymbol{N}'(\boldsymbol{S}_z) = k].\, q_{z-k}\,.$$

Затем Накамото делает упрощающее предположение, что блоки были добывались в соответствии со средним ожидаемым временем на блок. Это асимптотически верно, когда $z \to +\infty$, но в противном случае неверно. Точнее, он аппроксимирует $\boldsymbol{N}'(\boldsymbol{S}_z)$ через $\boldsymbol{N}'(t_z)$ = где

$$t_z = \mathbb{E}[\boldsymbol{S}_z] = z\mathbb{E}[\boldsymbol{T}] = \frac{z\tau_0}{p}\quad.$$

Как мы видели выше, случайная величина $\boldsymbol{N}'(t_z)$ следует за распределением Пуассона с параметром

$$\lambda = \alpha'\ t_z = \frac{z\alpha'\tau_0}{p} = \frac{zq}{p}.$$

Завершающее исчисление

$$P_{sn}(z) = \mathbb{P}[\boldsymbol{N}'(t_z) \geq z] + \sum_{k=0}^{z-1} \mathbb{P}[\boldsymbol{N}'(t_z) = k].\, q_{z-k}\,.$$

$$= 1 - \sum_{k=0}^{z-1} \mathbb{P}[\boldsymbol{N}'(t_z) = k] + \sum_{k=0}^{z-1} \mathbb{P}[\boldsymbol{N}'(t_z) = k].\, q_{z-k}\,.$$

$$= 1 - \sum_{k=0}^{z-1} e^{\lambda} \frac{\lambda^k}{k!}(1-q_{z-k}).$$

Заметим, что этот анализ не является правильным, потому что

$$\boldsymbol{N}'(\boldsymbol{S}_z) \neq \boldsymbol{N}'(t_z)$$

8.6. Правильный анализ

Пусть $\boldsymbol{X}_n = \boldsymbol{N}'(\boldsymbol{S}_z)$ - количество блоков, добытых атакующими, когда честные майнеры только что добыли n-й блок. Мы вычисляем распределение для $\boldsymbol{X}_n$.

Предожение 8.6.1 Случайная величина $\boldsymbol{X}_n$ имеет отрицательное биномиальное распределение с параметрами (n, p), т. е. для $k \geq 0$,

$$\mathbb{P}[\boldsymbol{X}_n = k] = p^n q^k \binom{k+n-1}{k}.$$

Доказательство,

Пусть $k \geq 0$. Мы имеем, что $\boldsymbol{N}'$ и $\boldsymbol{S}_n$ независимы, следовательно

$$\mathbb{P}[\boldsymbol{X}_n = k] = \int^{+\infty}_0 \mathbb{P}[\boldsymbol{N}'(\boldsymbol{S}_n) = k \mid \boldsymbol{S}_n \in [t, t+dt]] \cdot \mathbb{P}[\boldsymbol{S}_n \in [[t, t+dt]]$$

$$= \int^{+\infty}_0 \mathbb{P}[\boldsymbol{N}'(t) = k] \cdot f_{S_n}(t)\, dt$$

$$= \int^{+\infty}_0 \frac{(\alpha' t)^k}{k!} e^{\alpha' t} \cdot \frac{\alpha}{(n-1)!} t^{n-1} e^{-\alpha t} dt$$

$$= \frac{p^n q^k}{(n-1)!\, k!} \cdot \int^{+\infty}_0 t^{k+n-1} e^{-t} dt$$

$$= \frac{p^n q^k}{(n-1)!\, k!} \cdot (k+n-1)!$$

Таким образом, мы подтверждаем, что распределение $\boldsymbol{X}_n$ не является законом Пуассона с параметром nq/p, как утверждает Накамото. Только асимптотически мы имеем сходимость к распределению Пуассона

Предложение 8.6.2 В пределе $n \to +\infty$, $q \to 0$, и $l_n = nq/p \to \lambda$ мы имеем:

$$\mathbb{P}[\boldsymbol{X}_n = k] \to \frac{\lambda^k}{k!} e^{-\lambda}.$$

Доказательство,

$$\mathbb{P}[\boldsymbol{X}_n = k] = \frac{n^n}{(n+l_n)^n} \frac{l_n^k}{(n+l_n)} \frac{(k+n-1)!}{(n-1)!\, k!}$$

$$= \frac{l_n^k}{k!} \frac{1}{\left(1+\frac{l_n}{n}\right)^n} \frac{n(n+1)\ldots(n+k-1)}{(n+l_n)^k}$$

получаем следующий результат $(1+\frac{l_n}{n})^n \to e^{\lambda}$

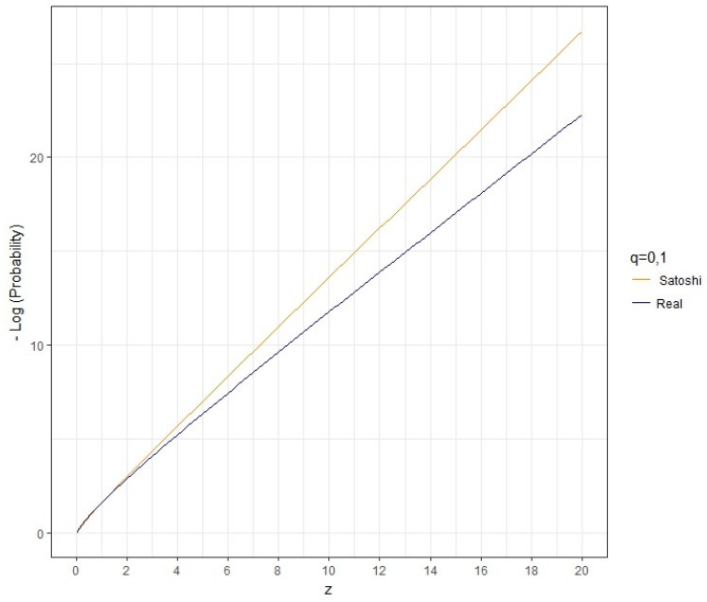
Предложение 8.6.3
(Вероятность успеха атакующих) вероятность успеха атакующих майнеров, после того, как z блоки были добыты честнымы майнерами.

$$P(z)=1-\sum_{k=0}^{z-1}(p^z q^k - q^z p^k)\binom{k+z-1}{k}.$$

Доказательство.

У нас есть,

$$P(z)=\sum_{k>z} p^z q^k \binom{k+z-1}{k}+\sum_{k=0}^{z}(\frac{q}{p})^{z-k} p^z q^k \binom{k+z-1}{k}$$
$$1-\sum_{k=0}^{z}(p^z q^k - q^z p^k)\binom{k+z-1}{k}$$
$$1-\sum_{k=0}^{z-1}(p^z q^k - q^z p^k)\binom{k+z-1}{k}$$



Численное применение.

Преобразуя в R код, заданный $0 < q < 1/2$ и $z \geq 0$, эта простая функция вычисляет нашу вероятность $P(z)$:

```
prob ← function(z,q){
p=1-q;
sum=1;
for (k in 0:(z-1)) {sum=sum-(p^z*q^k-q^z*p^k)*choose(k+z-1,k)} ;
return(sum)}
```

Мы можем сравнить с вероятностью P_{SN}

Для $q = 0.1$ мы имеем

z	$P(z)$	$P_{SN}(z)$

0	1.0000000	1.0000000
1	0.2000000	0.2045873
2	0.0560000	0.0509779
3	0.0171200	0.0131722
4	0.0054560	0.0034552
5	0.0017818	0.0009137
6	0.0005914	0.0002428
7	0.0001986	0.0000647
8	0.0000673	0.0000173
9	0.0000229	0.0000046
10	0.0000079	0.0000012

Для $q = 0.3$ мы имеем

z	$P(z)$	$P_{SN}(z)$
0	1.0000000	1.0000000
5	0.1976173	0.1773523
10	0.0651067	0.0416605
15	0.0233077	0.0101008
20	0.0086739	0.0024804
25	0.0033027	0.0006132
30	0.0012769	0.0001522
35	0.0004991	0.0000379
40	0.0001967	0.0000095
45	0.0000780	0.0000024
50	0.0000311	0.0000006

Решения для P менее 0,1% мы имеем

q	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45
z	6	9	13	20	32	58	133	539
z_{SN}	5	8	11	15	24	41	81	340

Объяснение в том, что результат Накамото верен только в том случае, если время майнинга честными майнерами точно соответствует ожидаемому времени. Если время больше среднего времени, то это только помогает злоумышленниками.

8.7. Формула замкнутой формы

Приведем формулу закрытой формы для $P(z)$ с использованием регуляризованной неполной бета-функции $I_x(a, b)$.

Теорема 8.7.1. Мы, с $s = 4pq$

$$P(z) = I_s(z, 1/2) .$$

Напомним, что неполная бета-функция определяется для $a, b > 0$ и $0 \le x \le 1$, по

$$B_x(a,b)=\int_0^x t^{a-1}(1-t)^{b-1}dt ,$$

а классическая бета-функция определяется через $B(a, b) = B_1(a, b)$.

Регуляризованная неполная бета-функция определяется

$$I_x=\frac{B_x(a.b)}{B_x(a,b)}=\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}B_x(a,b).$$

Доказательство. Интегральная функция распределения случайной величины ***X*** с отрицательным биномиальным распределением, с $0 < p < 1$ и $q = 1-p$ как обычно, задается

$$F_x(k) = \mathbb{P}[\boldsymbol{X} \le k] = \sum_{l=0}^{k} p^z q^l \binom{l+z-1}{l} = 1 - I_p(k+1, z).$$

Это вытекает из формулы

$$I_p = (k+1,z) = I_p(k,z) - \frac{p^k q^z}{kB(k,z)},$$

мы доказываем, интегрируя по частям определение $B_x(a, b)$.
Таким образом, мы получаем

$$P(z) = 1 - I_p(z,z) + I_q(z,z).$$

Внося изменение переменных $t \to 1-t$ в интегральное определение, мы также имеем отношение симметрии.

$$I_p = (a,b) + I_q(b,a) = 1$$

Поэтому, получаем $I_p(z,z) + I_q(z,z) = 1$, и $P(z)=2I_q(z,z)$. Результат при использовании $I_q(z,z) = \frac{1}{2} I_s(z, 1/2)$, где $s=4pq$.

8.8 Асимптотический и экспоненциальный распад.

Накамото делает наблюдение, без доказательства того, что вероятность уменьшается экспоненциально до 0 при $z \to +\infty$. Мы докажем этот факт для истинной вероятности $P(z)$ с помощью закрытой формула из предложения 8.7.1.

Предложение 8.8.1.

Когда $z \to +\infty$ мы имеем, с $s = 4pq < 1$,

$$P(z) \sim \frac{s^z}{\sqrt{\pi(1-s)z}}$$

Путем интегрирования по частям получаем следующую элементарную версию леммы Уотсона:

Лемма 8.8.2. Пусть $f \in C^1(\mathbb{R}_+)$ с $f(0)\ 6 \neq 0$ и абсолютно сходящимся интегралом

$$\int_0^{+\infty} f(u)e^{-zu}\,du < +\infty,$$

тогда, когда $z \to +\infty$, мы имеем

$$\int_0^{+\infty} f(u)e^{-zu}\,du \sim \frac{f(0)}{z}.$$

Тогда получаем следующую асимптотику:

Лемма 8.8.3. Для $s, b \in \mathbb{R}$, мы имеем, когда $z \to +\infty$,

$$B_s(z,b) \sim \frac{s^z}{z}(1-s)^{b-1}.$$

Доказательство.

Внесем изменения переменной $u = \log(s/t)$ в определение

$$B_s(z,b) = \int_0^s t^{z-1}(1-t)^{b-1}dt,$$

получаем

$$B_s(z,b) = s^z \int_0^{+\infty} (1-se^{-u})^{b-1}e^{-zu}\,dt,$$

и результат следует из леммы 8.8.2 с $f(u) = (1 - se^{-u})^{b-1}$.
Теперь мы заканчиваем доказательство предложения 8.8.1. По асимптотике Стирлинга,

$$B(z, 1/2) = \frac{\Gamma(z)\Gamma(1/2)}{\Gamma(z+1/2)} \sim \sqrt{\frac{\pi}{z}},$$

$$I_s(z, ½) = \frac{B_s(z,1/2)}{B(z,1/2)} \sim \frac{(1-s)^{-1/2}\frac{s^z}{z}}{\sqrt{\frac{\pi}{z}}} \sim \frac{s^z}{\sqrt{\pi(1-s)z}}.$$

8.9. Более точный анализ рисков.

На практике, во избежание двойной траты, получатель Bitcoin транзакции ожидает $z \geq 1$ подтверждений. Но, у него также есть информация о времени τ_1, которое потребовалось для подтверждения транзакции z раз. Очевидно, что вероятность успеха атакующих увеличивается с τ_1. Соответствующим параметром является относительное отклонение от ожидаемого времени.

$$K = \frac{\tau 1}{z t_0} = \frac{p\tau 1}{z\tau 0}.$$

Наша цель - вычислить вероятность $P(z, k)$ успеха атакующих. Обратите внимание, что $P(z, 1)$ - вероятность, вычисленная Накамото.

$$P_{SN}(z) = P(z, 1).$$

Расчет $P(z,k)$.

За время τ_1 злоумышленники добыли $k \geq 0$ блоков с вероятностью, следующей за распределением Пуассона с параметром

$$\lambda(z,k) = \alpha'\tau_1 = k \frac{zq}{p},$$

это значит

$$\mathbb{P}\left[\boldsymbol{N}'(\tau_1)=k\right]=\frac{\left(\frac{zq}{p}k\right)^k}{k!}e^{\frac{-zq}{p}k},$$

При к = 1 восстанавливается приближение Накамото.

Кумулятивное распределение Пуассона может быть вычислено с неполной регуляризованной гамма-функцией

$$Q(s,x)=\frac{\Gamma(s,x)}{\Gamma(x)},$$

$$\text{где}\quad \Gamma(s,x)=\int_x^{+\infty} t^{s-1}e^{-t}dt$$

является неполной гамма-функцией, а Γ(s) = Γ (s, 0) - регулярная Гамма-функция. У нас есть

$$Q(z,\lambda)=\sum_{k=0}^{z-1}\frac{\lambda^k}{k!}.$$

Мы вычисляем, как раньше

$$P(z,k)=\sum_{k=z}^{+\infty}\left(\frac{(\lambda(z,k)^k)}{k!}\right)e^{-\lambda(z,k)}+\sum_{k=0}^{z-1}\left(\frac{q}{p}\right)^{z-k}\frac{(\lambda(z,k))^k}{k!}e^{-\lambda(z,k)}$$

$$=\ 1-\sum_{k=0}^{z-1}\left(1-\left(\frac{q}{p}\right)^{z-k}\right)\frac{(\lambda(z,k))^k}{k!}e^{\lambda(z,k)}$$

$$=\ 1-Q(z,kzq,/p)+\left(\frac{q}{p}\right)^z e^{kz\frac{p-q}{p}}Q(z,kz).$$

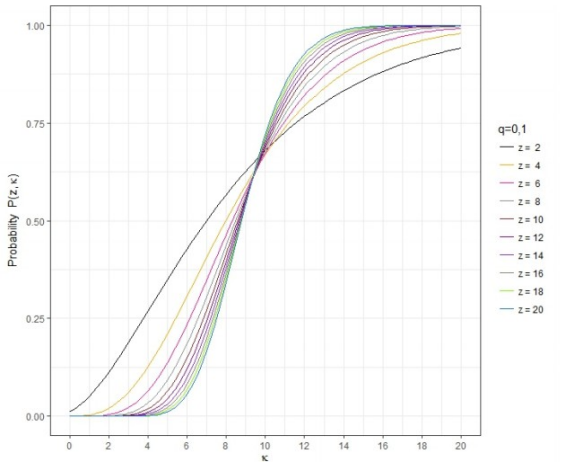


Рисунок 2. Вероятность успеха как функция *k*

Таким образом, мы получаем в явном замкнутом виде формулы для *P*(*z*,*k*),

Теорема 8.9.1. У нас есть

$$P(z,k)=1-Q(z,kzq,/p)+\left(\frac{q}{p}\right)^z e^{kz\frac{p-q}{p}}Q(z,kz),$$

$$P_{sn}(z)=P(z,1)=1-Q(z,kzq,/p)+(\frac{q}{p})^{z}\ e^{z\frac{p-q}{p}}\ Q(z,z).$$

Найдены асимптотики $Q(z, \lambda z)$ при $z \to +\infty$ при различных значениях $\lambda > 0$.

8.10. Асимптотика $P(z,k)$ и $P_{sN}(z)$.

Лемма 8.10.1. У нас есть

(1) Для $0 < \lambda < 1$, $Q(z, \lambda,z) \to 1$ и $Q(z, \lambda,z) \sim \frac{1}{1-\lambda}\frac{1}{\sqrt{2\pi z}}e^{-z(\lambda-1-\log\lambda)}$.

(2) Для $\lambda = 1$, $Q(z, z) \to 1/2$ и $1/2 - Q(z,z) \sim \frac{1}{3\sqrt{2\pi z}}$.

(3) Для $\lambda > 1$, $Q(z, \lambda z) \sim \frac{1}{1-\lambda}\frac{1}{\sqrt{2\pi z}}e^{-z(\lambda-1-\log\lambda)}$.

Доказательство. *Для $\lambda < 1$ имеем*

$$1 - Q(z,\lambda z)= \frac{\lambda(z,\lambda z)}{\Gamma(z)}$$

$$\sim \frac{z^z\lambda^z e^{-z\lambda}}{z!(1-\lambda)}$$

$$\sim \frac{1}{1-\lambda}\frac{1}{\sqrt{2\pi z}}e^{-z(\lambda-1-\log\lambda)}$$

$$\sim \frac{1}{2}$$

и

$$= \frac{1}{2}-Q(z,z)=\frac{1}{2}-\frac{z^{z-1}e^{-z}\sqrt{\frac{\pi z}{2}}(1-\frac{1}{3}\sqrt{\frac{2}{\pi z}+o(z^{-1/2})})}{(z-1)!}$$

$$= \frac{1}{2}-\frac{1}{2}\frac{\sqrt{2\pi z}(z/e)^z}{z!}(1-\frac{1}{3}\sqrt{\frac{2}{\pi z}+o(z^{-1/2})})$$

$$= \frac{1}{2}-\frac{1}{2}\frac{\sqrt{2\pi z}(z/e)^z}{\sqrt{2\pi z}(z/e)^z(q+\frac{1}{12z}+o(z-1))}(1-\frac{1}{3}\sqrt{\frac{2}{\pi z}+o(z^{-1/2})})$$

$$= \frac{1}{2}-\frac{1}{2}(1+\frac{1}{12z}+o(z^{z-1})).(1-\frac{1}{3}\sqrt{\frac{2}{\pi z}+o(z^{-1/2})})$$

$$= \frac{1}{3\sqrt{2\pi z}}+o(z^{-1/2})$$

(3) В силу формулы Стирлинга, при $\lambda > 1$ имеем

$$Q(z,\lambda z) = \frac{\Gamma(z,\lambda z)}{\Gamma(z)},$$

$$\sim \frac{(\lambda z)^z e^{-z\lambda}}{z!(\lambda-1)}$$

$$\sim \frac{1}{1-\lambda}\frac{1}{\sqrt{2\pi z}}e^{-z(\lambda-1-\log\lambda)}$$

При $x > 0$ определим $c(x) = x - 1 - \log_x$, что положительно, так как график $x \to 1 - x$ - касательная к $x = 1$ к вогнутому графу функции логарифма. Обозначим $0 < \lambda = q/p < 1$.

Мы имеем, что вероятность Накамото $P_{SN}(z)$ также уменьшается экспоненциально с z, как утверждает Накамото без доказательства.

Предложение 8.10.2. Мы имеем для $z \to +\infty$,

$$P_{SN}(z) \sim \frac{e^{-zc(\lambda)}}{2}$$

Доказательство. Результат следует из Формулы замкнутой формы из теоремы 7,

$$P(z, k) = 1 - Q(z, kzq/p) + (q/p)^z \; e^{kz\frac{p-q}{q}} Q(z, kz),$$

а затем из точек (1) и (2) Леммы 9.1,

$$1-Q(z,\frac{q}{p}z)=o(e^{-zc(q/p)}),$$

$$(\frac{q}{p})^z e^{z(1-\frac{q}{p})} Q(z,z)\sim\frac{1}{2}e^{-zc(q/p)}.$$

В более общем случае, мы имеем пять различных режимов асимптотики $P(z, k)$ для $0 < к < 1$, $к = 1$, $1 < к < p/q$, $k = p/q$ и $k > p/q$.

Предложение 8.10.3. Мы имеем для $Z \to +\infty$

(1) *Для* $0 < к < 1$,

$$P(z,k)\sim\frac{1}{1-k\lambda}\frac{1}{\sqrt{2\pi z}}e^{-zc(k\lambda)}.$$

(2) *Для* $k=1$,

$$P(z,1) = P_{sn}\;(z)\sim\frac{1}{2}e^{-zc(\lambda)}$$

(3) *Для* $1 < к < p/q$,

$$P(z,k)\sim\frac{k(1-\lambda)}{(k-1)(1-k\lambda)}\frac{1}{\sqrt{2\pi z}}e^{-zc(k\lambda)}.$$

(4) *Для* $к = p/q$, $P(z, p/q) \to 1/2$ *и*

$$P(z,p/q)-1/2\sim\frac{1}{2\pi z}(\frac{1}{3}+\frac{q}{p-q})$$

(5) *Для* $p/q < k$, $P(z, k) \to 1$ *и*

$$1-P(z,k)\sim\frac{k(1-\lambda)}{(k-1)(k\lambda-1)}\frac{1}{\sqrt{2\pi z}}e^{-zc(k\lambda)}$$

Доказательство. *(1) Если* к *< 1, то также kq / p < 1, и*

$$1-Q(z,kzq/p)\sim\frac{1}{1-kq/p}\frac{1}{\sqrt{2\pi z}}e^{-z(kq/p-1-\log(kq/p))},$$

и

$$(q/p)^z e^{kz\frac{p-q}{q}}=e^{-z(kq/p-1-\log(kq/p))}$$

$$e^{-z(1-k)(1-q/p)}.e^{-z(q/p-1-\log(q/p))},$$

и

$$\frac{(q/p)^z e^{kz\frac{p-q}{q}}}{1-Q(z,kzq/p)}\sim(1-kq/p)\sqrt{2\pi z}\cdot e^{-z(1-k)(1-q/p)}\cdot e^{-z(q/p-1-\log(q/p)-(kq/p-1-\log(kq/p)))}$$

$$\sim\quad(1-kq/p)\cdot\sqrt{2\pi}z\cdot e^{-z(1-k)(1-q/p)}\cdot e^{-z(1-\kappa)q/p}\cdot e^{-z\log k}$$

$$\sim\quad(1-kq/p)\cdot\sqrt{2\pi}z\cdot e^{-z(1-k-\log k)}=o(1).$$

Поскольку $Q(z, kz) \to 1$ мы имеем,

$$P(z,k)=1-Q(z,kzq/p)+(q/p)^z e^{kz\frac{p-q}{q}}\ Q(z,kz)$$

$$\sim\ 1-\ Q(z,\ kzq/p)$$

$$\sim\quad\frac{1}{(1-kq/p)\sqrt{2\pi z}}\cdot e^{-z(kq/p-1-\log(kq/p))}$$

(2) это было доказано в предложении 8.10.2,
(3) при 1 < к < p/q, то по Лемме 8.10.1,

$$(q/p)^z e^{kz\frac{p-q}{q}}\ Q(z,kz)\frac{1}{(k-1)\sqrt{2\pi z}}\cdot e^{-z(kq/p-1-\log(kq/p))},$$

и

$$1-Q(z,kzq/p)(q/p)\sim\frac{1}{1(1-kq/p\sqrt{2\pi z})}\cdot e^{-z(kq/p-1-\log(kq/p))}.$$

Так мы имеем

$$P(z,k)\sim(\frac{1}{1-kq/p}+\frac{1}{k-1})\cdot\frac{1}{\sqrt{2\pi z}}\cdot e^{-z(kq/p-1-\log(kq/p))}$$

$$\sim\quad\frac{k(1-q/p)}{(k-1)(1-kq/p)}\frac{1}{\sqrt{2\pi z}}\cdot e^{-z(\kappa q/p-1-\log(\kappa q/p))}.$$

(4) Предыдущая асимптотика в начале доказательства (3) также справедлива для 1 < к = p/q и дает

$$(q/p)^z e^{kz\frac{p-q}{q}}\ Q(z,kz)\sim\frac{q}{p-q}\frac{1}{\sqrt{2\pi z}},$$

и Леммой 8.10.1.

$$P(z,p/q)=1-Q(z,z)+(q/p)^z e^{kz\frac{p-q}{q}}\,Q(z,kz)$$

$$\frac{1}{2}+\frac{1}{\sqrt{2\pi z}}\left(\frac{1}{3}+\frac{q}{p-q}\right)+o(1/\sqrt{z}).$$

(5) для к > p / q мы снова используем ту же асимптотику (3), чтобы получить

$$Q(z,kzq/p)\sim\frac{1}{kq/p-1}\frac{1}{\sqrt{2\pi z}}e^{-z(kq/p-1-\log(kq/p))},$$

и еще

$$(q/p)^z e^{kz\frac{p-q}{q}}\,Q(z,kz)\sim\frac{1}{(k-1)\sqrt{2\pi z}}e^{-z(kq/p-1-\log(kq/p))},$$

так

$$1-P(z,k)\sim\left(\frac{1}{kq/p-1}-\frac{1}{k-1}\right)\sqrt{2\pi z}\,e^{-z(kq/p-1-\log(kq/p))}$$

$$\sim\ \frac{(1-q/p)}{(\kappa q/p-1)(k-1)}\sqrt{2\pi z}\,e^{-z(kq/p-1-\log(kq/p))}$$

8.11. Сравнение асимптотики *P*(*z*) и P_{SN} (*z*).

У нас есть асимптотическое сравнение,

Предложение 8.11.1. Мы имеем для Z → +∞,

$P_{SN}(z) \prec P(z)$.

Доказательство. Заметим, что

$$\frac{q}{p}-1-\log\left(\frac{q}{p}\right)-\log\left(\frac{1}{4pq}\right)=\left[\frac{1}{2p}-1-\log\left(\frac{1}{2p}\right)\right]>0$$

Итак, при s = *4pq* < 1 имеем

$$0<\log\frac{1}{s}<\frac{q}{p}-1-\log\frac{q}{p}=c(q/p)=c(\lambda),$$

и для *z* большой

$$P_{SN}(z)<e^{-zc(\lambda)}\prec\frac{s^z}{\sqrt{\pi(1-s)z}}\sim P(z).$$

Как мы увидим позже, мы можем более подробно рассказать о неравенстве между P_{SN} (*z*) и *P*(*z*).

8.12. Восстановление *P*(*z*) из *P*(*z, k*).

Выше мы видели, что $P_{SN}(z)$ можно восстановить из $P(z, k)$, приняв значение k=1. Оказывается, мы также можем восстановить $P(z)$ как взвешенное среднее по k $P(z, k)$.

Теорема 8.12.1. У нас есть

$$P(z)=\int_0^{+\infty} P(z,k)\,dp_z(k)$$

с функцией плотности

$$dp_z(k)=\frac{z^z}{(z-1)!}k^{z-1}e^{-zk}\,dk.$$

Мы проверяем, что

$$\int_0^{+\infty} dp_z(k)=1.$$

Мы можем написать

$$P(z)=1-\sum_{k=0}^{z-1} f_k(k),$$

где

$$f_k=(1-(\frac{q}{z})^{z-k})\frac{(zq/p)^k}{k!}k^k e^{\frac{zq}{p}k}.$$

Тогда Теорема вытекает из прямого вычисления,

Лемма 8.12.2. Для $k \geq 0$, мы имеем

$$\int_0^{+\infty} f_k(k)\,dp_z(k)=(p^z q^k-q^z p^k)\binom{k+z-1}{k}.$$

Приведем еще одно концептуальное доказательство.

Доказательство. Рассмотрим случайную величину

$$\boldsymbol{k}=\frac{p}{z\tau_0}\boldsymbol{S}_z.$$

Выше мы видели, что $\boldsymbol{S}_z \sim \Gamma(z, \alpha)$ так $\boldsymbol{k} \sim \Gamma\left(z, \alpha\frac{z\tau_0}{p}\right) = \Gamma(z, z)$. Таким образом, плотность dp_z равна распределению $\boldsymbol{\kappa}$. Достаточно доказать, что

$$P(z) = \mathbb{E}[P(z,\boldsymbol{k})].$$

У нас есть

$$P(z) = \mathbb{P}[\boldsymbol{N}'(\boldsymbol{S}_z) \geq z] + \sum_{k=0}^{z-1} \mathbb{P}[\boldsymbol{N}'(\boldsymbol{S}_z) = k] \cdot q_{z-k}$$

$$=1 - \sum_{k=0}^{z-1} (1-q_{z-k})\,\mathbb{P}[\boldsymbol{N}'(\boldsymbol{S}_z) = k].$$

И благодаря кондиционированию S_z мы получаем

$$P(z) = 1 - \sum_{k=0}^{z-1} (1-q_{z-k})\, \mathbb{E}[\mathbb{P}][\boldsymbol{N}'(\boldsymbol{S}_z) = k \mid \boldsymbol{S}_z]]$$

$$=1 - \mathbb{E}\ [\sum_{k=0}^{z-1} \frac{(\alpha' \boldsymbol{S}_z)}{k!} e^{\alpha' \boldsymbol{S}_z}] + (\frac{q}{p})^z\ \ \mathbb{E}\ \ [e^{\alpha' \frac{p-q}{q} \boldsymbol{S}_z} \sum_{k=0}^{z-1} \frac{(\frac{\alpha' p}{q})}{k!} e^{\frac{\alpha' p}{q} \boldsymbol{S}_z}]$$

$$= \mathbb{E}\ \ [1 - Q(z, \frac{zq}{p}\boldsymbol{k}) + (\frac{q}{p})^z e^{z(1-\frac{q}{p})\boldsymbol{k}} Q(z, z\boldsymbol{k})]$$

$$= \mathbb{E}\ \ [P(z, \boldsymbol{k})],$$

поскольку $\mathbb{P}[\boldsymbol{N}'(\boldsymbol{S}_z) = k\boldsymbol{S}_z] = \frac{(\alpha' \boldsymbol{S}_z)^k}{k!} e^{\alpha' \boldsymbol{S}_z}, q_{z-k} = (q/p)^{z-k}$, и

$$Q(z, x) = \sum_{k=0}^{z-1} \frac{x^k}{k!} e^{-x}.$$

Мы также отмечаем, что $\mathbb{E}[\boldsymbol{k}] = 1$.

8.13. Диапазон *k*.

Вероятность наблюдения отклонения больше k равна $\mathbb{P}[\boldsymbol{k} > k]$ с $\boldsymbol{k} = \frac{p}{z\tau_0} \boldsymbol{S}_z$.

Мы имеем, что $\boldsymbol{k}$ следует Г-распределению, $\boldsymbol{k} \sim \Gamma(z, z)$, поэтому

$$\mathbb{P}[\boldsymbol{k} > k] = \frac{1}{\Gamma(z)} \int_k^{+\infty} z^z t^{t-1} e^{-zt}\, dt$$

$$= \frac{1}{\Gamma(z)} \int_k^{+\infty} t^{z-1} e^{-t}\, dt$$

$$= \frac{\Gamma(z, kz)}{\Gamma(z)}$$

$$= Q(z, kz).$$

Тогда по лемме 8.10.1 $\mathbb{P}[\boldsymbol{k} > k] \sim \frac{1}{k-1} \frac{1}{\sqrt{2\pi z}} e^{-zc(k)}$ при $\boldsymbol{k} > 1$. Заметим, что эта вероятность не зависит от p. При $z = 6$ имеем $\mathbb{P}[\boldsymbol{k} > 4] \approx 3 \cdot 10^{-6}$ и для $z = 10$, $\mathbb{P}[\boldsymbol{k} > 4] \approx 4 \cdot 10^{-9}$. Таким образом, на практике вероятность иметь $\boldsymbol{k} > 4$ очень маловероятна. Ниже мы представили график $\boldsymbol{k} \to P(z, k)$ при разных значениях z ($q = 0{,}1$) и $0 < k < 4$.

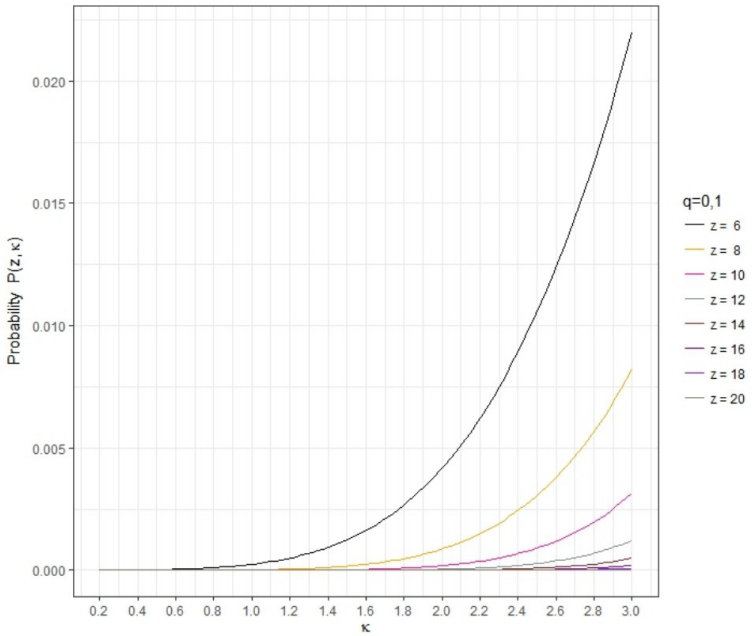


Рисунок 3. Вероятность $P(z, k)$ как функция k

Мы видим, что $k \to P(z, k)$ выпукло в диапазоне значений рассматриваемого k. Мы изучаем выпуклость более подробно в следующем разделе.

8.14. Сравнение $P_{SN}(z)$ и $P(z)$.

Теперь изучим выпуклость $k \to P(z, k)$. Напомним, что $\lambda = q/p < 1$. Из теоремы 8.2.1. мы имеем

$$P(z, k) = 1\text{-}Q(z,z\lambda k)+ \lambda^z e^{z(1-\lambda)k} Q(z,zk) .$$

Поскольку

$$\Gamma(z)\partial_2 Q(z, x) = -x^{z-1}e^{-x},$$

мы получаем, после некоторых отмен,

$$\Gamma(z)\partial_2 P(z,k) = \lambda^z z(1-\lambda)e^{z(1-\lambda)k}\Gamma(z,zk).$$

Мы наблюдаем, что $\partial_2 P(z,k) > 0$, поэтому $P(z,k)$ является возрастающей функцией k, как и ожидалось. Для второй производной мы имеем

$$\Gamma(z)\partial_2 P(z,k) = \lambda^z z^2 (1-\lambda)e^{z(1-\lambda)k}[(1-\lambda)\Gamma(z,zk) - (zk)^{z-1}e^{-kz}]$$

$$= \lambda^z z(1-\lambda)e^{-\lambda kz}(zk)^z[(1-\lambda)Q(z,zk)z!e^{kz}(zk)^{-z}-k^{-1}] .$$

Поэтому мы изучаем признак

$$g_{\lambda,z}(k) = (1-\lambda)Q(z, zk)z!e^{kz}(zk)^{-z} - k^{-1}$$

$$(1-\lambda)\sum_{k=0}^{z-1}\frac{z!}{z^{z-k}k!}\frac{1}{k^{z-k}} - k^{-1}$$

$$\frac{1-\lambda}{k}\left(\left(1-\frac{1}{z}\right)\frac{1}{k}+\left(1-\frac{1}{z}\right)\left(1-\frac{2}{z}\right)\frac{1}{k^2}+\ldots\right)-\frac{\lambda}{k}$$

Для $z = 1$ имеем

$$g_{\lambda,1}(k) = -\lambda/k < 0\ ,$$

поэтому $k \to P(1, k)$ является вогнутой функцией и неравенством Йенсена

$$P(1)=\int_0^{+\infty} P(z,k)\,d\rho_1(\tau)\le P(1,\bar{k})=P(1,1)=P_{SN}(1).$$

Следствие 8.14.1. *Имеем (для всех $0 < q < 1/2$)*

$$P(1) \le P_{SN}(1)\ .$$

В общем случае для $z \ge 2$ мы имеем обратное неравенство. Для определения знака $g_{\lambda, z}$ изучаем его нули.

Уравнение для решения

$$(1-\frac{1}{z})\frac{1}{k}+(1-\frac{1}{z})(1-\frac{2}{z})\frac{1}{k^2}+\ldots+(1-\frac{1}{z})\ldots(1-\frac{z-1}{z})\frac{1}{k^{z-1}}=\frac{\lambda}{1-\lambda}.$$

Это полиномиальное уравнение в $1 / k$, коэффициенты увеличиваются на z, а левая сторона уменьшается на $k \in (0,+\infty)$ от $+\infty$ до 0, поэтому существует единственное решение $k(z)$, и

$$k(2) < k(3) < \ldots$$

Мы вычисляем

$$k(2)=\frac{1-\lambda}{2\lambda}=\frac{1}{2q}-1>0.$$

В этом случае функция $k \to P(z, k)$ выпуклая только в интервале $(0, k(z))$. При больших z , большая часть меры dp_z содержится в этом интервале и мы имеем по неравенству Йенсена

$$P(z)\approx\int_0^{k(z)} P(z,k)\,dp_z(k)\ge P(z,\bar{k})\approx P(z,1)=P_{sn}(z),$$

где

$$\bar{k}_z=\int_0^{k(z)} k\,dp_z(k)\approx\int_0^{+\infty} k\,dp_z(k)=1.$$

Мы можем получить некоторые оценки на $k(z)$ при $z \to +\infty$. Первое замечание состоит в том, что для больших z имеем $k(z) > 1$. Асимптотические пределы для $Q(z, kz)$ для $k < 1$ и $k = 1$ (Лемма 8.10.1) и асимптотическая формула Стирлинга дают

$$Q(z, kz)z!e^{kz}(zk)^z \to +\infty\ ,$$
$$\text{и } g_{\lambda,z}(k)\ 6 \ne 0.$$

Для $k > 1$ можно использовать асимптотику, $z \to +\infty$,

$$\Gamma(z,kz)\sim\frac{(kz)^z e^{-kz}}{(k-1)z}$$

и

$$(1-\lambda)\Gamma(z,\kappa z)-(\kappa z)^{z-1}e^{-kz}\sim(kz)^{z-1}e^{-kz}\left((1-\lambda)\frac{k}{k-1}-1\right),$$

таким образом, с

$$g_{\lambda,z}(k) = (1 - \lambda)\,\Gamma(z, kz) z e^{kz} (kz)^{-z} - k^{-1},$$

у нас есть

$$g_{\lambda,\infty}(k) = \lim_{z \to +\infty} g_{\lambda,z}(k) = \frac{1}{k}\left((1-\lambda)\frac{k}{k-1} - 1\right) = \frac{1-\lambda}{k-1} - \frac{1}{k}.$$

Теперь, если

$$k(\infty) = \lim_{z \to +\infty} k(z),$$

у нас есть $g_{\lambda,\infty}(k_\infty) = 0$, так что мы получаем:

Предложение 8.14.2.

$$k(\infty) = \lim_{z \to +\infty} k(z) = \lambda^{-1} = \frac{p}{q}.$$

Используя асимптотику второго порядка, для $k > 1$, $z \to +\infty$,

$$\Gamma(z, kz) \sim \frac{(kz)^z e^{-kz}}{z(k-1)}\left(1 - \frac{k}{(k-1)^2 z}\right),$$

так

$$g_{\lambda,z}(k) \sim \frac{1-\lambda}{k-1}\left(1 - \frac{k}{(k-1)^2 z}\right) - k^{-1}.$$

пишем

$$k(z) = \frac{p}{q} - \frac{a}{z} + o(z^{-1}),$$

и используем

$$\frac{1-\lambda}{k(z)-1}\left(1 - \frac{k(z)}{(k(z)-1)^2 z}\right) - k(z)^{-1}$$

мы получаем

Предложение 8.14.3. Для $z \to +\infty$

$$k(z) = \frac{p}{q} - \frac{p^2}{q(p-q)}\frac{1}{z} + 0(z^{-1}).$$

Также мы имеем

$$\frac{p}{q} - 1 > \frac{p^2}{q(p-q)}\frac{1}{z}$$

для

$$z > \left(\frac{p}{p-q}\right)^2,$$

Итак, для z порядка $(1-\lambda)^{-2}$ имеем $k(z) > 1$.

8.15. Оценки для *P*(*z*)

Помните, что мы установили *s* = *4pq*. У нас есть следующее неравенство, которое является частным случаем более общих неравенств Гаутши.

Лемма 8.15.1. Пусть $z \in \mathbb{R}_+$. У нас есть

$$\sqrt{\frac{z}{z+\frac{1}{2}}} \leqslant \frac{\Gamma(z+\frac{1}{2})}{\sqrt{z\Gamma(z)}} \leqslant 1.$$

Доказательство. По неравенству Коши-Шварца мы имеем:

$$\begin{aligned}
\Gamma(z+\tfrac{1}{2}) &= \int_0^{+\infty} t^{z-\frac{1}{2}} e^{-t}\, dt \\
&\int_0^{+\infty} (t^{\frac{z}{2}} e^{-\frac{t}{2}}).(t^{\frac{z}{2}-\frac{1}{2}} e^{\frac{t}{2}})\, dt \\
\leq \quad & \Big(\int_0^{+\infty} t^z e^{-t}\, dt\Big)^{\frac{1}{2}}.\Big(\int_0^{+\infty} t^{z-1} e^{-t}\, dt\Big)^{\frac{1}{2}} \\
\leq \quad & \Gamma(z+1)^{\frac{1}{2}}.\Gamma(z)^{\frac{1}{2}} \\
\leq \quad & (z\Gamma(z))^{\frac{1}{2}}.\Gamma(z)^{\frac{1}{2}} \\
\leq \quad & \sqrt{z\Gamma(z)}
\end{aligned}$$

С другой стороны, последнее неравенство с заменой *z* на *z* + ½ дает:

$$z\Gamma(z) = \Gamma(z+\frac{1}{2}+\frac{1}{2}) \leq \sqrt{z+\frac{1}{2}\Gamma(z+\frac{1}{2})}$$

Лемма 8.15.2. Для *z* > 1, мы имеем

$$\sqrt{\frac{z}{z+\frac{1}{2}}}.\frac{s^z}{\sqrt{\pi z}} \leq P(z) \leqslant \frac{1}{\sqrt{1-s}}.\frac{s^z}{\sqrt{\pi z}}.$$

Доказательство. Функция $x \to (1 - x)^{-½}$ не уменьшается. Таким образом, по определению I_s и верхней границы неравенства леммы 8.15.1, имеем

$$\begin{aligned}
P(z) = I_s(z,\tfrac{1}{2}) &= \frac{\Gamma(z+\frac{1}{2})}{\Gamma(\frac{1}{2})\Gamma(z)} \int_0^s t^{z-1}(1-t)^{-\frac{1}{2}}\, dt \\
&\leq \frac{1}{\sqrt{\pi}} \frac{\Gamma(z+\frac{1}{2})}{\Gamma(z)} \int_0^s t^{z-1}(1-s)^{-\frac{1}{2}}\, dt \\
&\leq \frac{\Gamma(z+\frac{1}{2})}{\sqrt{z}\,\Gamma(z)}.\frac{s^z}{\sqrt{\pi(1-s)z}}
\end{aligned}$$

$$\leq \frac{1}{\sqrt{1-s}} \cdot \frac{s^z}{\sqrt{\pi z}}.$$

Точно так же, используя нижнюю грань неравенства леммы 8.15.1, имеем

$$P(z) = I_s(z, \frac{1}{2}) \geqslant \frac{1}{\sqrt{\pi}} \frac{\Gamma(z+\frac{1}{2})}{\Gamma(z)} \int_0^s t^{z-1} dt$$

$$\geq \frac{\Gamma(z+\frac{1}{2})}{\sqrt{z}\,\Gamma(z)} \cdot \frac{s^z}{\sqrt{\pi z}}$$

$$\geq \sqrt{\frac{z}{z+\frac{1}{2}}} \cdot \frac{s^z}{\sqrt{\pi z}}.$$

Заметим, что это снова приводит к экспоненциальному уменьшению вероятности Накамото.

8.16. Верхняя граница для $P_{SN}(z)$

Доказательство 8.16.1. Мы имеем,

$$P_{sn}(z) < \frac{1}{1-\frac{q}{p}} \frac{1}{\sqrt{2\pi z}} e^{-(\frac{q}{p}-1-\log\frac{q}{p})z} + \frac{1}{2} e^{-(\frac{q}{p}-1-\log(\frac{q}{z})z)}$$

Эта верхняя граница довольно резка, учитывая асимптотику предложения 8.10.3 (2).

Лемма 8.16.2. Пусть $z \in \mathbb{N}^*$ и $\lambda \in \mathbb{R}^*_+$.

(1) Если $\lambda \in]0, 1[$, то $1 - Q(z, \lambda z) < \frac{1}{1-\lambda} \frac{1}{\sqrt{2\pi z}} e^{-(\lambda-1-\log\lambda)z}$

(2) $Q(z, z) < ½$

Доказательство.

$$\gamma(a, x) = e^{-x} x^a \sum_{n=0}^{\infty} \frac{\Gamma(a)}{\Gamma(a+n+1)} x^n,$$

что справедливо для $a, x \in \mathbb{R}$. Пусть $\lambda \in]0, 1[$. Используя $\Gamma(z + 1) = z\Gamma(z)$, получаем:

$$\gamma(z, \lambda z) = e^{-\lambda z} (\lambda z)^z \sum_{n=0}^{+\infty} \frac{\Gamma(z)}{\Gamma(z+n+1)} (\lambda z)^n$$

$$= e^{-\lambda z} (\lambda z)^z \left(\frac{1}{z} + \frac{1}{z(z+1)}(\lambda z) + \frac{1}{z(z+1)(z+2)}(\lambda z)^2 + \ldots\right)$$

$$\leq e^{-\lambda z} (\lambda z)^z \left(\frac{1}{z} + \frac{1}{z^2}(\lambda z) + \frac{1}{z^3}(\lambda z)^2 + \ldots\right)$$

$$\leq e^{-\lambda z} (\lambda z)^z \frac{1}{z} \frac{1}{1-\lambda}$$

$$\leq \quad \frac{\lambda^z z^{z-1} e^{-\lambda z}}{1-\lambda}$$

С другой стороны, мы имеем

$$\frac{1}{\Gamma(z)} < \frac{e^z}{\sqrt{2\pi z}\, z^{z-1}},$$

и для любого $0 < \lambda < 1$,

$$1-Q(z,\lambda z) = \frac{\gamma(z,\lambda z)}{\Gamma(z)}$$

$$< \quad \frac{1}{1-\lambda}\frac{1}{\sqrt{2\pi z}} e^{-(\lambda-1-\log\lambda)z}$$

Это происходит непосредственно из

Ссылаясь на то, что $P_{SN}(z) = P(z,1) = \quad 1-Q(z,\frac{q}{p}z)+(q/p)^z e^{z(p-q)/p} Q(z,z),$, мы получаем предложение 15.1.

8.17. Повтроное сравнение $P_{SN}(z)$ и $P(z)$.

Целью данного раздела является вычисление явного ранга z_0 (не резкий), для которых $P_{SN}(z) < P(z)$ для $z \geq = z_0$.

Лемма 8.17.1. Пусть $\alpha > 0$. Для всех $x > \log \alpha$, $e^x - \alpha x > \frac{\alpha}{2}(x-\log\alpha)^2+\alpha(1-\log\alpha)$.

Доказательство.

Пусть $g(x) = e^x - \alpha x - \frac{\alpha}{2}(x-\log\alpha)^2 - \alpha(1-\log\alpha)$. Мы получаем
$g'(x) = e^x - \alpha - \alpha(x-\log\alpha), g''(x) = e^x - \alpha$ и $g^{(3)}(x) = e^x$.

$g(\log\alpha) = g'(\log\alpha) = g''(\log\alpha) = 0$ и $g^{(3)} > 0$. Поэтому, $g(x) > 0$ для $x > \log\alpha$.

Лемма 8.17.2. Для $\alpha > 0$ и $x > \quad (1+1/\sqrt{2})\log\alpha$ имеем $e^x > \alpha x$.

Доказательство.

При $x \leq 0$ неравенство тривиально. Таким образом, мы можем считать, что $x > 0$. Для $0 < \alpha < 1$, мы имеем $e^x > x > \alpha x$. Для $1 < \alpha < e$, по Лемме 8.17.1, имеем $e^x - \alpha x > 0$ для $x > \log\alpha$. Для $\alpha > e$ - наибольший корень многочлена $\frac{\alpha}{2}(x-\log\alpha)^2+\alpha(1-\log\alpha)$ это $\log\alpha+\sqrt{2(\log\alpha-1)}$ который меньше, чем $(1+1/\sqrt{2})\log\alpha$ поскольку $\sqrt{2(u-1)} \leqslant u/\sqrt{2}$ для $u \geq 1$. Таким образом, неравенство снова вытекает из леммы 8.17.1.

Лемма 8.17.3. Для μ, ψ, x > 0, если

$$x > \frac{1}{2\sqrt{2}} - \frac{1+\sqrt{2}}{2\sqrt{2}}\frac{\log(2\psi\mu^2)}{\psi}$$

то у нас есть

$$e^{-\psi x} < \frac{\mu}{\sqrt{x+\frac{1}{2}}}.$$

Доказательство. Мы имеем

$$e^{-\psi\cdot x} < \frac{\mu}{\sqrt{x+\frac{1}{2}}} \Leftarrow\Rightarrow (x+1/2)\, e^{-2\psi\cdot x} < \mu^2$$
$$\Leftarrow\Rightarrow \quad (x+1/2)\, e^{-2\psi\cdot(x+1/2)} < \mu^2 e^{-\psi}$$
$$\Leftarrow\Rightarrow \quad e^{-2\psi\cdot(x+1/2)} > \frac{x+1/2}{\mu^2 e^{-\psi}}$$
$$\Leftarrow\Rightarrow \quad e^{-2\psi\cdot(x+1/2)} > \frac{1}{2\psi\mu^2 e^{-\psi}}, 2\psi\cdot(x+1/2)$$

По Лемме 8.17.2 последнее неравенство выполняется, как только

$$2\psi\cdot(x+1/2) > (1+1/\sqrt{2})\log(\frac{1}{2\psi\mu^2 e^{-\psi}}).$$

Более того, мы имеем

$$2\psi\cdot(x+1/2) > (1+1/\sqrt{2})\log(\frac{1}{2\psi\mu^2 e^{-\psi}}) \Leftarrow\Rightarrow 2\psi\cdot x+\psi > (1+1/\sqrt{2})\log(\frac{e^{\psi}}{2\psi\mu^2})$$

$$\Leftarrow\Rightarrow \quad 2\psi\cdot x+\psi > (1+1/\sqrt{2})\,\psi - (1+1\sqrt{2})\log(2\psi\mu^2)$$
$$\Leftarrow\Rightarrow \quad 2\psi\cdot x > \frac{1}{\sqrt{2}}\cdot\psi - (1+1/\sqrt{2})\log(2\psi\mu^2)$$
$$\Leftarrow\Rightarrow \quad x > \frac{1}{2\sqrt{2}} - \frac{1+1/\sqrt{2}}{2}\frac{\log(2\psi\mu^2)}{\psi}$$

Теорема 8.17.4. Пусть $z \in \mathbb{N}$. Достаточным условием наличия $P_{SN}(z) < P(z)$ является $z \geq z_0$, где $z_0 = \lceil z^*_0 \rceil$ - наименьшее целое, большее или равное

$$z^*_0 = \quad max(\frac{2}{\pi(1-\frac{q}{p})^2}), \frac{1}{2\sqrt{2}} - \frac{(1+\frac{1}{\sqrt{2}})}{2}\frac{\log(\frac{2\psi(p)}{\pi})}{\psi(p)}$$

$$где \quad \psi(p) = \frac{q}{p} - 1 - \log(\frac{q}{p}) - \log(\frac{1}{4pq}) > 0.$$

Доказательство. Во-первых, обратите внимание, что

$$\psi(p) = \frac{q}{p} - 1 - \log(\frac{p}{q}) - \log(\frac{1}{4p^2}\frac{p}{q})$$

$$2\left[\frac{1}{2p} - 1 - \log(\frac{1}{2p})\right]$$

Итак, $\psi(p) > 0$ и z_0 корректно определены. Пусть $z > z_0$. По Лемме 8.15.2 и следствию 8.16.1 достаточно доказать, что

$$\frac{1}{1-\frac{q}{p}}\frac{1}{\sqrt{2\pi z}}e^{-z}(\frac{q}{p} - 1 - \log\frac{q}{p}) + \frac{1}{2}e^{z(\frac{q}{p}-1-\log(\frac{q}{p}))} < S\sqrt{\frac{z}{z+\frac{1}{2}}}\frac{s^z}{\sqrt{\pi z}}$$

Мы имеем $z \geq z_0 \geq \frac{2}{\pi\left(1-\frac{q}{p}\right)^2}$ поэтому, $\frac{1}{1-\frac{q}{p}} \frac{1}{\sqrt{2\pi z}} \leq \frac{1}{2}$. Итак, неравенство выполняется, как только $e^{-z\psi(p)} < \frac{\left(\frac{1}{\sqrt{\pi}}\right)}{\sqrt{z+\frac{1}{2}}}$ и результат следует из Леммы 8.17.3.

Резкие значения вычисляются численно и приведены в таблице ниже:

z_o	2	3	4	5	6	7	8	9	10	11
$q \geq$	0.000	0.232	0.305	0.342	0.365	0.381	0.393	0.401	0.409	0.415

8.18. Таблицы для P(*z*, *k*).

Таблица для *P*(3, *k*) (*z* = 3) при разных значениях *k* и *q* в %.

*k**q*	0.02	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18	0.2	0.22	0.24	0.26
0.1	0	0.01	0.03	0.09	0.18	0.33	0.55	0.88	1.34	1.96	2.78	3.87	5.27
0.2	0	0.01	0.05	0.11	0.23	0.42	0.71	1.12	1.68	2.44	3.44	4.74	6.39
0.3	0	0.02	0.06	0.15	0.3	0.55	0.91	1.42	2.11	3.04	4.24	5.77	7.7
0.4	0	0.02	0.08	0.19	0.39	0.69	1.14	1.77	2.62	3.74	5.17	6.98	9.22
0.5	0	0.03	0.1	0.24	0.49	0.87	1.43	2.2	3.22	4.56	6.25	8.36	10.93
0.6	0	0.04	0.13	0.31	0.61	1.08	1.76	2.69	3.92	5.49	7.47	9.9	12.83
0.7	0.01	0.05	0.16	0.38	0.75	1.33	2.14	3.25	4.7	6.54	8.82	11.59	14.89
0.8	0.01	0.06	0.19	0.46	0.92	1.61	2.58	3.88	5.57	7.7	10.3	13.42	17.11
0.9	0.01	0.07	0.24	0.56	1.11	1.92	3.06	4.58	6.53	8.96	11.9	15.39	19.45
1	0.01	0.08	0.28	0.67	1.32	2.27	3.6	5.36	7.58	10.32	13.61	17.47	21.9
1.1	0.01	0.1	0.34	0.8	1.55	2.66	4.19	6.2	8.71	11.78	15.42	19.64	24.44
1.2	0.02	0.12	0.4	0.94	1.81	3.09	4.84	7.1	9.92	13.32	17.32	21.91	27.05
1.3	0.02	0.14	0.47	1.09	2.1	3.55	5.53	8.07	11.2	14.95	19.3	24.24	29.72
1.4	0.02	0.16	0.54	1.26	2.4	4.06	6.27	9.1	12.55	16.64	21.34	26.62	32.41
1.5	0.02	0.19	0.62	1.44	2.74	4.59	7.06	10.18	13.96	18.39	23.44	29.04	35.12
1.6	0.03	0.22	0.71	1.64	3.1	5.17	7.9	11.32	15.43	20.2	25.58	31.49	37.83
1.7	0.03	0.25	0.81	1.85	3.48	5.78	8.78	12.51	16.95	22.06	27.76	33.96	40.53
1.8	0.04	0.28	0.91	2.08	3.89	6.42	9.7	13.75	18.52	23.95	29.96	36.42	43.2
1.9	0.04	0.32	1.03	2.33	4.32	7.1	10.67	15.03	20.13	25.88	32.18	38.88	45.84
2	0.05	0.36	1.15	2.58	4.78	7.8	11.67	16.35	21.77	27.83	34.4	41.32	48.43
2.1	0.05	0.4	1.28	2.86	5.26	8.54	12.71	17.7	23.44	29.8	36.62	43.74	50.96
2.2	0.06	0.44	1.41	3.15	5.77	9.31	13.78	19.09	25.14	31.78	38.84	46.12	53.43
2.3	0.07	0.49	1.56	3.46	6.3	10.11	14.88	20.51	26.86	33.77	41.04	48.46	55.84
2.4	0.07	0.54	1.71	3.78	6.85	10.94	16.01	21.95	28.59	35.75	43.21	50.76	58.17
2.5	0.08	0.6	1.87	4.11	7.42	11.79	17.17	23.41	30.34	37.73	45.36	53	60.43
2.6	0.09	0.65	2.04	4.46	8.01	12.67	18.35	24.89	32.09	39.7	47.48	55.19	62.6
2.7	0.1	0.71	2.22	4.83	8.62	13.57	19.56	26.39	33.84	41.65	49.56	57.32	64.7
2.8	0.11	0.78	2.41	5.21	9.26	14.49	20.78	27.9	35.59	43.59	51.6	59.38	66.71
2.9	0.12	0.85	2.6	5.6	9.91	15.44	22.02	29.42	37.34	45.5	53.6	61.39	68.64
3	0.13	0.92	2.81	6.01	10.58	16.4	23.28	30.94	39.08	47.38	55.55	63.32	70.49
3.1	0.14	0.99	3.02	6.44	11.27	17.38	24.55	32.47	40.81	49.24	57.45	65.19	72.25
3.2	0.15	1.07	3.24	6.87	11.97	18.38	25.83	34	42.52	51.06	59.31	67	73.93
3.3	0.16	1.15	3.47	7.32	12.69	19.39	27.12	35.52	44.22	52.85	61.11	68.73	75.53

3.4	0.17	1.23	3.7	7.78	13.43	20.42	28.42	37.05	45.9	54.61	62.86	70.39	77.05
3.5	0.19	1.32	3.95	8.26	14.18	21.46	29.73	38.56	47.56	56.32	64.55	71.99	78.5

Таблица для $P(6, k)$ ($z = 6$) при разных значениях k и q в %.

$k\backslash q$	0.02	0.04	0.06	0.08	0.1	0.12	0.14	0.16	0.18	0.2	0.22	0.24	0.26
0.1	0	0	0	0	0	0	0	0.01	0.02	0.04	0.08	0.15	0.28
0.2	0	0	0	0	0	0	0.01	0.01	0.03	0.06	0.12	0.23	0.41
0.3	0	0	0	0	0	0	0.01	0.02	0.05	0.09	0.18	0.34	0.6
0.4	0	0	0	0	0	0.01	0.01	0.03	0.07	0.15	0.28	0.51	0.88
0.5	0	0	0	0	0	0.01	0.02	0.05	0.11	0.23	0.42	0.75	1.28
0.6	0	0	0	0	0	0.01	0.04	0.08	0.17	0.34	0.63	1.1	1.84
0.7	0	0	0	0	0.01	0.02	0.06	0.13	0.26	0.51	0.91	1.57	2.57
0.8	0	0	0	0	0.01	0.03	0.08	0.19	0.39	0.73	1.3	2.19	3.53
0.9	0	0	0	0	0.02	0.05	0.12	0.28	0.55	1.03	1.81	2.99	4.73
1	0	0	0	0.01	0.02	0.07	0.18	0.39	0.78	1.43	2.45	3.99	6.19
1.1	0	0	0	0.01	0.04	0.1	0.25	0.54	1.06	1.92	3.25	5.2	7.93
1.2	0	0	0	0.01	0.05	0.14	0.35	0.74	1.42	2.53	4.21	6.63	9.94
1.3	0	0	0	0.02	0.07	0.2	0.47	0.98	1.86	3.26	5.35	8.29	12.23
1.4	0	0	0	0.03	0.09	0.26	0.62	1.28	2.39	4.14	6.68	10.19	14.79
1.5	0	0	0.01	0.03	0.12	0.34	0.8	1.64	3.02	5.15	8.19	12.3	17.58
1.6	0	0	0.01	0.05	0.16	0.45	1.02	2.06	3.76	6.31	9.89	14.63	20.59
1.7	0	0	0.01	0.06	0.21	0.57	1.29	2.56	4.6	7.62	11.77	17.16	23.78
1.8	0	0	0.02	0.08	0.27	0.71	1.6	3.14	5.56	9.07	13.82	19.86	27.13
1.9	0	0	0.02	0.1	0.34	0.89	1.96	3.79	6.63	10.67	16.04	22.72	30.59
2	0	0	0.03	0.12	0.42	1.09	2.37	4.53	7.82	12.42	18.4	25.71	34.14
2.1	0	0	0.03	0.15	0.51	1.32	2.83	5.35	9.12	14.29	20.9	28.81	37.73
2.2	0	0	0.04	0.19	0.62	1.58	3.36	6.26	10.54	16.29	23.51	31.98	41.34
2.3	0	0	0.05	0.23	0.75	1.88	3.95	7.26	12.06	18.41	26.23	35.21	44.94
2.4	0	0.01	0.06	0.28	0.89	2.21	4.59	8.35	13.69	20.64	29.02	38.47	48.49
2.5	0	0.01	0.07	0.33	1.05	2.59	5.3	9.52	15.42	22.95	31.87	41.73	51.97
2.6	0	0.01	0.09	0.4	1.24	3	6.08	10.78	17.24	25.35	34.77	44.98	55.35
2.7	0	0.01	0.1	0.47	1.44	3.45	6.92	12.12	19.15	27.81	37.69	48.19	58.63
2.8	0	0.01	0.12	0.55	1.67	3.95	7.82	13.54	21.14	30.33	40.62	51.34	61.78
2.9	0	0.02	0.14	0.64	1.92	4.49	8.79	15.04	23.19	32.89	43.54	54.42	64.8
3	0	0.02	0.17	0.74	2.2	5.08	9.82	16.6	25.31	35.48	46.44	57.41	67.66
3.1	0	0.02	0.19	0.85	2.5	5.71	10.91	18.24	27.47	38.08	49.29	60.3	70.38
3.2	0	0.03	0.22	0.97	2.83	6.39	12.06	19.93	29.68	40.68	52.1	63.09	72.94
3.3	0	0.03	0.26	1.11	3.18	7.11	13.27	21.68	31.93	43.28	54.84	65.75	75.33
3.4	0	0.03	0.3	1.25	3.57	7.88	14.54	23.48	34.2	45.86	57.52	68.3	77.57
3.5	0	0.04	0.34	1.41	3.98	8.69	15.86	25.33	36.48	48.41	60.11	70.72	79.66

Заключение

Нами движет глобальная идея и инновационные технологии. Мы хотим привлечь внимание широкой аудитории, чтобы масштабировать проект Tkeycoin на международном уровне.

Сейчас мы работаем над запуском рабочей сети на основе нашего ядра. Наши специалисты подготавливают базовый фундамент, который в дальнейшем послужит отправной точкой для тысяч технологических решений. Ядро, которое мы разрабатываем, - универсально, и в будущем может применяться в любой сфере деятельности.

На базе протокола можно будет запускать децентрализованные приложения, создавать инновационные решения для банковской и финансовых сфер, подключать корпоративный процессинг для приема платежей на сайте, использовать технологию для безопасного хранения больших данных.

Наши взгляды кардинально отличаются от большинства ICO проектов, которые присутствуют сейчас на рынке. Мы готовы решать самые сложные задачи, предоставляя миллионам людей те продукты, в которых они нуждаются.

После проведения ограниченной эмиссии посредством ICO, платформа перейдет на второй этап развития, где сеть станет децентрализованной. В течение следующих 12 месяцев будет проведена дополнительная работа с целью повышения стабильности, совместимости и совершенствования управления. По мере разработки протокола Tkeycoin DAO будет расти сложность используемых технологий, увеличиваться их взаимозависимость, будут использоваться более неординарные криптографические примитивы.

Если у Вас есть предложения или желание участвовать и помогать проекту, направьте запрос на support@tkeycoin.com

Список вспомогательных ресурсов:

Данный список может помочь пользователям, кто хочет дополнительно разобраться в представленной проектной документации.

I. Эллиптическая кривая Curve25519
https://en.wikipedia.org/wiki/Curve25519
II. Математическое ожидание
http://www.statisticssolutions.com/directory-of-statistical-analyses-mathematical-expectation/
III. Система цифровой подписи с открытом ключом Ed25519
https://ed25519.cr.yp.to/
IV. Анализ эллиптической кривой Curve25519
http://safecurves.cr.yp.to/
V. Теория множеств
https://en.wikipedia.org/wiki/Set_theory
VI. Hashcash. Адам Бак
http://www.hashcash.org/papers/hashcash.pdf
VII. Сатоши Накомото. Bitcoin: Одноранговая платежная система
https://bitcoin.org/bitcoin.pdf
VIII. Вэй Дай. «B-Money»
http://www.weidai.com/bmoney.txt
IX. Ожидаемое значение
https://en.wikipedia.org/wiki/Expected_value
X. Уильям Феллер. Введение в теорию вероятностей и ее приложения
https://www.researchgate.net/profile/William_Balthes/post/Stronger_limit_Theorems_then_the_strong_law_of_large_numbers/attachment/59d6256e6cda7b8083a21880/AS%3A445697653776384%401483274010393/download/William+Feller-An+Introduction+To+Probability+Theory+And+Its+Applications.+Vol+II.pdf
XI. Chord одноранговая сеть
https://en.wikipedia.org/wiki/Chord_(peer-to-peer)
XII. Pastry одноранговая сеть
https://en.wikipedia.org/wiki/Pastry_(DHT)
XIII. Tapestry одноранговая сеть
https://en.wikipedia.org/wiki/Tapestry_(DHT)
XIV. Сергей Попов. Вероятностный анализ алгоритма Nxt
http://ledger.pitt.edu/ojs/index.php/ledger/article/view/46
XV. Алгоритм Гровера
https://en.wikipedia.org/wiki/Grover%27s_algorithm